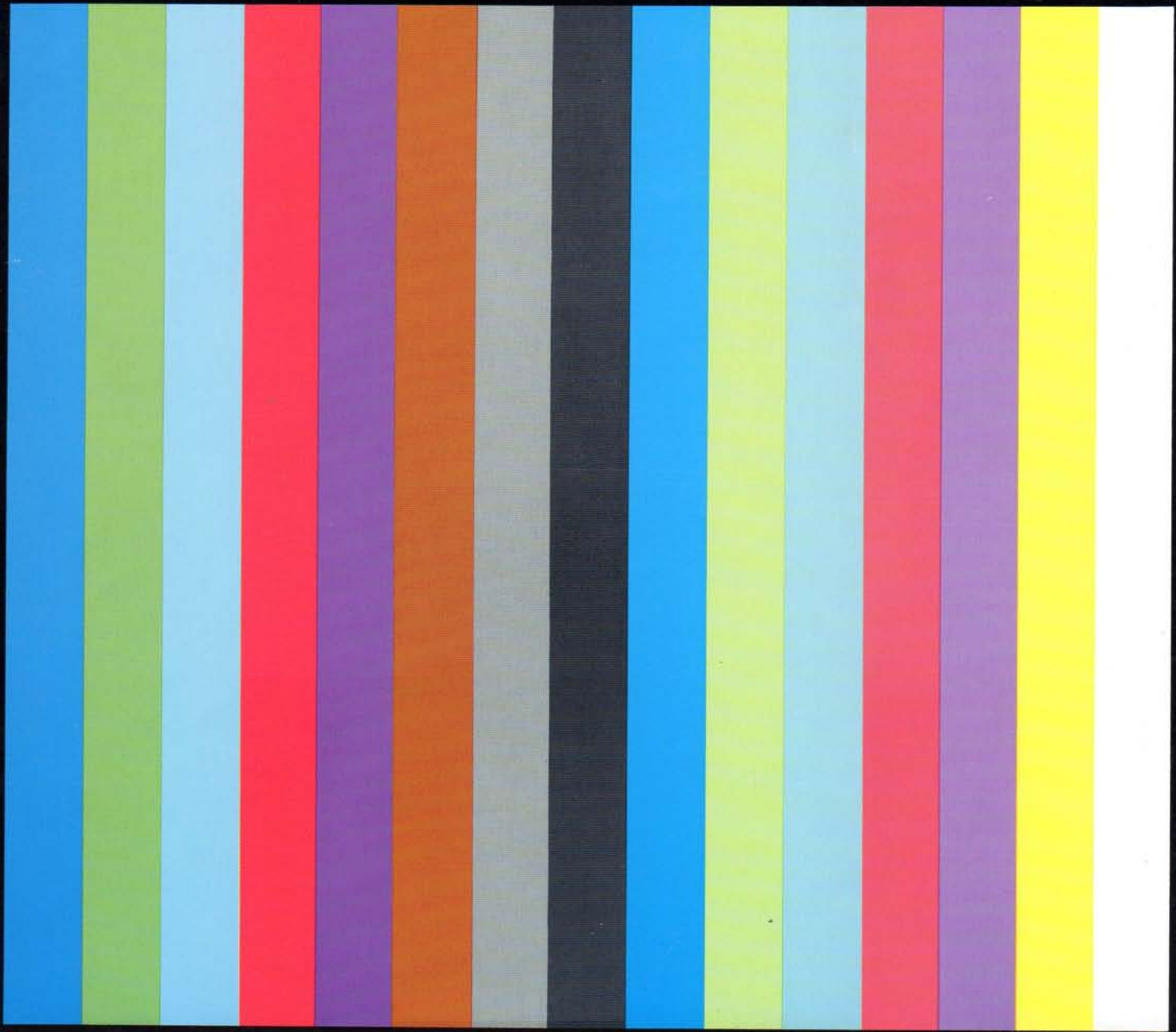
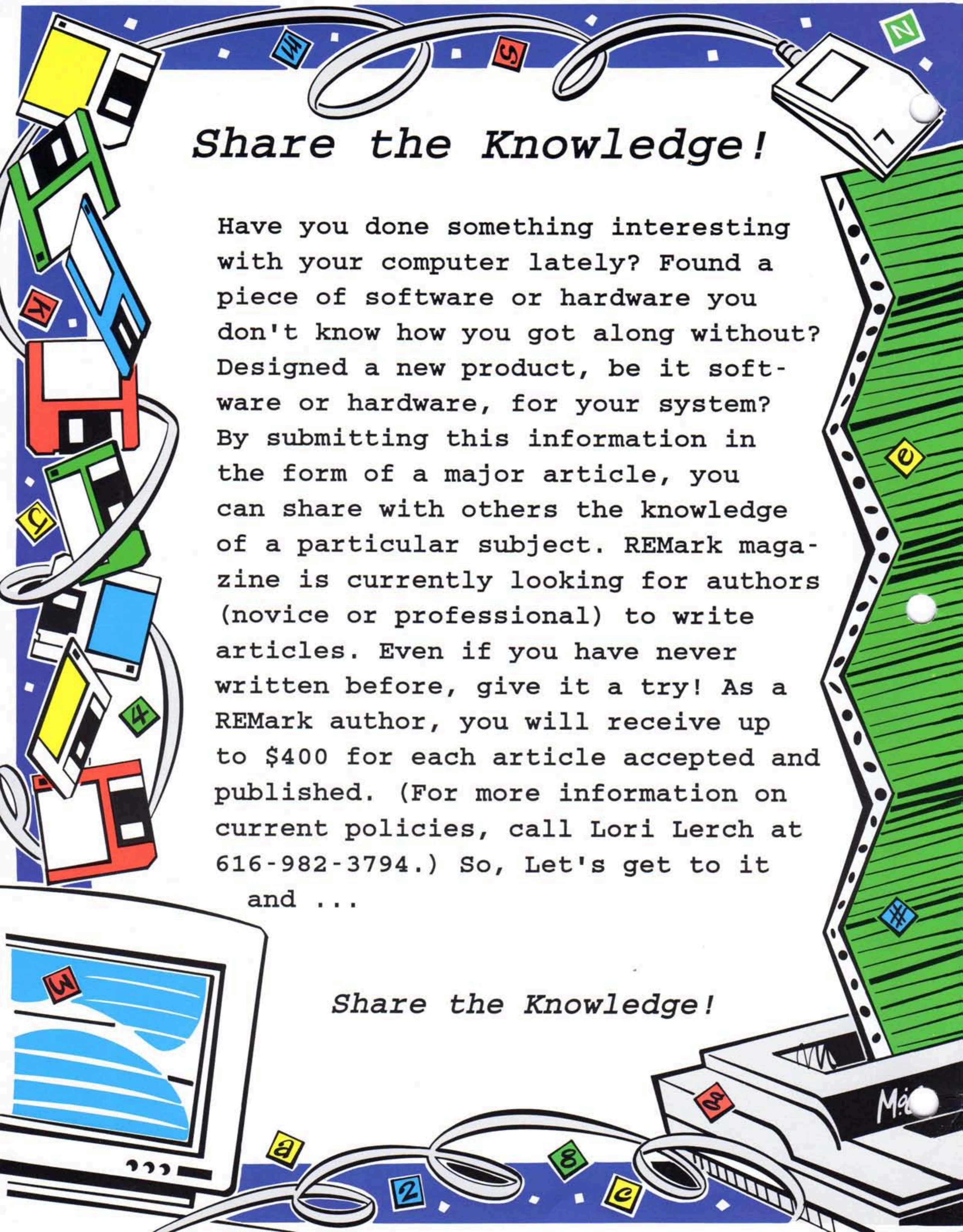
April 1992

REMark®

The Official Zenith Data Systems Computer Users' Magazine



->*.DOS [Pages 7, 16, 21]



Share the Knowledge!

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 product, be it software or hardware, for your system? By submitting this information in the form of a major article, you can share with others the knowledge of a particular subject. REMark magazine is currently looking for authors (novice or professional) to write articles. Even if you have never written before, give it a try! As a REMark author, you will receive up to $400 for each article accepted and published. (For more information on current policies, call Lori Lerch at 616-982-3794.) So, Let's get to it and ...

Share the Knowledge!

REMark®

April 1992

The Official Zenith Data Systems Users Magazine

Advertising

Resources

Software

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
H8 - H/Z-89/90				
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 PKG	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	CHKBK 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 PKG	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 SECT SUPPORT PKG	885-1121	HDOS	UTILITY	30.00
HDOS PROG. HELPER	885-8017	HDOS	UTILITY	16.00
HOME FINANCE	885-1070	HDOS	BUSINESS	18.00
HUG DISK DUP UTILITY	885-1217-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG	885-4500	VARIOUS	PROD TO 1982	9.75
HUGMAN & MOVIE ANIM	885-1124	HDOS	ENTERTAINMENT	20.00
INFO SYS AND TEL. & MAIL SYS	885-1108-[37]	HDOS	DBMS	30.00
LOGBOOK	885-1107-[37]	HDOS	AMATEUR RADIO	30.00
MAGBASE	885-1249-[37]	CPM	MAGAZINE DB	25.00
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 DEC '80	20.00
RUNOFF	885-1025	HDOS	TEXT PROCR	35.00
SCICALC	885-8027	HDOS	UTILITY	20.00
SMALL BUISNESS PACKAGE	885-1071-[37]	HDOS	BUSINESS	75.00
SMALL-C COMPILER	885-1134	HDOS	LANGUAGE	30.00
SOFT SECTOR SUPPORT PKG	885-1127-[37]	HDOS	UTILITY	20.00
STUDENT'S STATISTICS PKG	885-8021	HDOS	EDUCATION	20.00
SUBMIT (Z80 ONLY)	885-8006	HDOS	UTILITY	20.00
TERM & HTOC	885-1207-[37]	CPM	COMMUN & UTIL	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)				
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	CHKBK 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 CAT UPDT #1	885-4501	VARIOUS	PROD 1983 TO 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
SEA BATTLE	885-1211-[37]	CPM	GAME	20.00
UTILITIES BY PS	885-1226-[37]	CPM	UTILITY	20.00
UTILITIES	885-1237-[37]	CPM	UTILITY	20.00
X-REFERENCE UTIL FOR MBASIC	885-1231-[37]	CPM	UTILITY	20.00
ZTERM	885-3003-[37]	CPM	COMMUNICATIONS	20.00

Software Order Form

Detach this form, enclose your check, money order or credit card information (no cash please).

Part No.	Description	Price
885-		
885-		
885-		
885-		

Sub-Total ______

10% S&H ______
(Min. $1.00/Max. $5.00)

Total ______

-HS = Hard-Sector
-37 = Soft-Sector
-80 = 3.5"
-90 = 2" (Add $3.00)

Name ______

Address ______

______

City, State, Zip ______

Credit Card # ______

Expiration Date ______

Price List

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
		H/Z-100 (Not PC) Only		
CARDCAT	885-3021-37	MSDOS	BUSINESS	20.00
CHEAPCALC	885-3006-37	MSDOS	UTILITY	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
DUNGN & DRAGONS (ZBASIC)	885-3009-37	MSDOS	GAME	20.00
ETCHDUMP	885-3005-37	MSDOS	UTILITY	20.00
EZPLOT II	885-3049-37	MSDOS	PRINTER PLOT UTIL	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	UTILITY	20.00
HELPSCREEN	885-3039-37	MSDOS	UTILITY	20.00
HUG BKGRD 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
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	ATRONOMY UTILITY	20.00
SMALL-C COMPILER	885-3026-37	MSDOS	LANGUAGE	30.00
SPELL5	885-3035-37	MSDOS	SPELLING CHECKER	20.00
SPREADSHEET CONTEST PKG	885-3018-37	MSDOS	VARIOUS SPRDST	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
BACKGRD 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
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 CAT UPD #1	885-4501	MSDOS	PROD 1983 - 1985	9.75
HUGMCP	885-3033	MSDOS	COMMUNICATION	40.00
ICT 8080 - 8088 TRANSLATOR	885-3024	MSDOS	UTILITY	20.00
MAGBASE	885-3050	VARIOUS	MAG DATABASE	25.00
MATT	885-8045	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025	MSDOS	UTILITIES	20.00
PS' PC &Z100 UTILITIES	885-3052	MSDOS	UTILITIES	20.00
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
REMARK VOL 11 ISSUES 120-131	885-4011	N/A	1990	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		
CARDCAT	885-6006	MSDOS	CAT SYSTEM	20.00
CHEAPCALC	885-6004	MSDOS	SPREADSHEET	20.00
CLAVIER	885-6016	MSDOS	ENTERTAINMENT	20.00
CP/EMULATOR II & ZEMULATOR	885-6002	MSDOS	CPM & Z100 EMUL	20.00
DUNGEONS & DRAGONS	885-6007	MSDOS	GAME	20.00
EZPLOT II	885-6013	MSDOS	PRINTER PLOT UTIL	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	UTILITIES	20.00
PS' 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 UTIL	20.00
TCSPELL	885-8044	MSDOS	SPELLING CHECKER	20.00
ULTRA RTTY	885-6012	MSDOS	AMATEUR RADIO	20.00
YAUD (YET ANOTHER UTIL DSK)	885-6015	MSDOS	UTILITIES	20.00

Managing Editor
Jim Buszkiewicz
(616) 982-3837

Software Engineer
Pat Swayne
(616) 982-3463

Production Coordinator
Lori Lerch
(616) 982-3794

Secretary
Lisa Cobb
(616) 982-3463

COM1 Bulletin Board
(616) 982-3956
(Modem Only)

ZUG Software Orders
(616) 982-3463

Contributing Editor
William M. Adney

Printer
Imperial Printing
St. Joseph, MI

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

To Locate your Nearest:
Dealer 1-800-523-9393
Service Center 1-800-777-4630

	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. Check ZUG Product List for availability of bound volumes of past issues. Requests for magazines mailed to foreign countries should specify mailing method and include appropriate, additional cost.

Send Payment to: Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49023-0217
(616) 982-3463

Although it is a policy to check material placed in REMark for accuracy, ZUG 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 Zenith Data Systems Computer Centers.

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

ZENITH
data systems

Groupe Bull

REMark Magazine Subscription & ZDS-COM1 Bulletin Board Information

Your subscription entitles you to receive REMark, our monthly magazine containing articles specific to Zenith Data Systems computer and generally to other PC Compatible computers. All articles in REMark are submitted by readers like you. We welcome YOUR articles, and will pay you for any we accept!

A REMark subscription also allows you full access to the ZDS-COM1 bulletin board system (COM1, for short). There are many, many megabytes of free and shareware software available for downloading to registered COM1 users. Full access also lets you order products from the "Bargain Centre" section of COM1. The money you can save in the Keyboard Shopping Club will pay for decades of REMark subscriptions.

Last, but definitely not least, your subscription puts you in touch with thousands of other Zenith Data Systems computer users, from whom invaluable information can be exchanged.

REMark subscriptions, currently $22.95, can be obtained in one of three ways. First, by ordering one on the COM1 bulletin board (see the Keyboard Shopping Club section); second, by phone with VISA, MasterCard, or American Express; and third, through the US Mail using a credit card, money order or check made payable to: Zenith Data Systems. Our address is:

Zenith Data Systems Users' Group
P.O. Box 217
Benton Harbor, MI 49023-0217
(616) 982-3463

Once you receive your ID number, registration on the COM1 BBS is NOT automatic. It requires that you log on, enter your first name and last name EXACTLY as they appear on your REMark mailing label, and then enter your ID number as your password. The FIRST time you access the board, you must elect to start a NEW ACCOUNT and answer the various questions. Once you've done this, our automated scanner will compare the system's database against the subscription database. If you made no mistakes, you will be verified and given full access within 24 hours.

Once you've been authorized as a full member, several important things happen. First, you're given full downloading privileges of up to one megabyte per day. Secondly, you'll have full access to the message boards. And finally, you'll be able to take full advantage of the Bargain Centre product savings.

- - - - ✂ -

Detach this form, enclose your check, money order or credit card information (no cash please).

REMark Subscription / Renewal Form

New Member: □ Yes □ No　　Credit Card # ____________________

ID Number: ____________________　　Exp. Date ____________________

Address Change? □

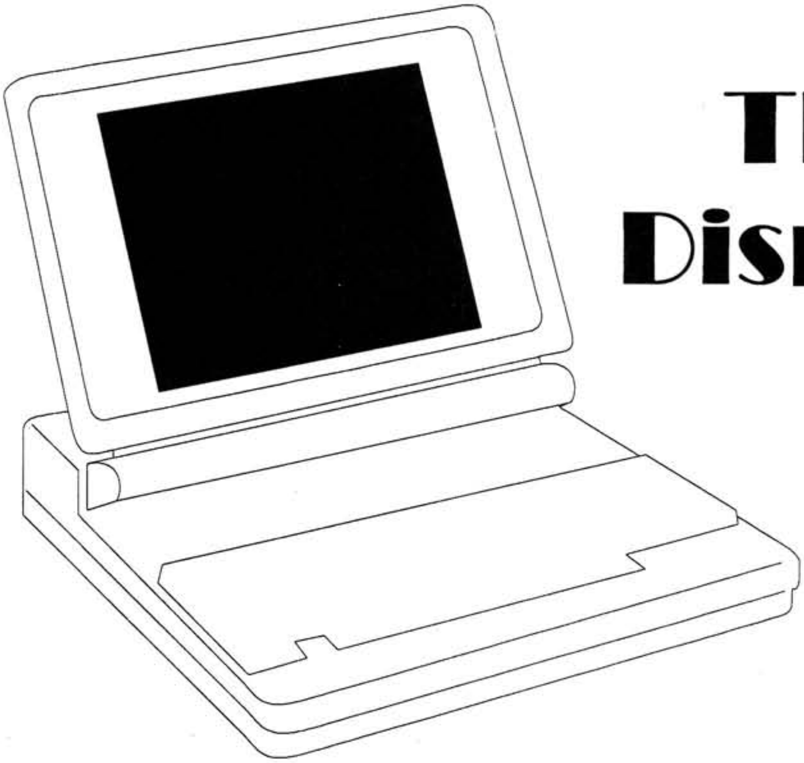
Name: ____________________

Address: ____________________

City, State, Zip: ____________________

Daytime Phone #: (　　) ____________________

	Renew	New
U.S. Bulk Mail	□ 19.95	□ 22.95
U.S. First Class	□ 32.95	□ 37.95
APO/FPO Surface Overseas	□ 32.95	□ 37.95
Air Printed Overseas	□ 52.95	□ 57.95



The Laptop Display Blues

Joseph M. Thomas
274 A Main Street
Stoneham, MA 02180

It is three in the morning and you have been up all night working on that report that was supposed to be on the boss's desk the day before. You're in the middle of making the last set of changes to the document when, for no apparent reason, your liquid crystal display (LCD) turns completely blue. You rub your eyes hoping that it is simply early signs of fatigue playing with your mind. You open your eyes slowly only to find that the LCD remains completely blue. You confirm that your laptop computer still has power and that the contrast and brightness adjustment slides are set properly. You then feel your way through the program by pressing the appropriate sequence of keys so you can save what changes you made to the document and hopefully find a quick fix to your problem. Does this scenario sound familiar? If you have experienced this problem you might want to read on, if not you might want to put this article aside and save it for future reference.

Before proceeding I should mention that most repairs on laptop computers should be made by an authorized repair technician, especially if the laptop is still under warranty. However, if the laptop is out of warranty and you don't have access to another computer on which you can finish your work you might want to try to fix the problem yourself. If you are as inquisitive as I am, you barely hesitate when you have a chance to open the case and attempt to fix the problem yourself, although at times this could be dangerous.

With a Phillip's head screwdriver in hand it is possible to have your Zenith Data Systems SupersPort laptop computer up and running again in about 30 minutes. In the following paragraphs I will describe some of the symptoms that can be associated with a loose display cable and then provide a simple step by step procedure for taking your laptop apart and checking the cable.

One of a laptop computer's weak points is the connection between the motherboard and the LCD. Over time the cable that connects the LCD to the motherboard tends to loosen. I have owned my Zenith Data Systems 184-20 for about 4 years and the cable has come loose a total of three times. The first time occurred while the laptop was under warranty so I took it to the service center and had the problem fixed. The problem started out to be intermittent. The bottom half of my display looked like a banner blowing in the wind with the eddies working their way across the screen. The technician saw the abnormal display and new what the problem was right away. The original laptop computers did not have a clamp installed to keep the cable from working its way free. A clamp was installed on my unit and the problem went away. ZDS now attaches a clamp at the factory to minimize the probability that the cable will detach itself from the motherboard.

Even with the clamp installed the cable tends to works its way free during normal use. I have fixed my own laptop two times and worked on two others in our lab. The bluing of the screen has some early symptoms that can provide a warning that the connection is working loose. As you are typing, does your screen seem to flicker or jitter? As the connector works its way free the regularity of the flickering will become more pronounced. This flicker continues until at some point you see nothing on the screen.

To check the cable I recommend that you have a cloth or towel that is large enough to lay the laptop on, a phillip's head screwdriver, a role of double sided scotch tape, and a few Q-tip swabs. To open the Z-184 series computer you want to do the following:

1. To minimize damage to your hard disk, ship or park the heads.
2. Turn off the power to the computer, remove any cables connected to the back plane, and lock the screen in its closed position.
3. Place the cloth on a kitchen table or other flat surface where there is enough room to work.
4. Place the computer upside down on the cloth. This will minimize the possibility of scratching the laptop's case.
5. At this time it might be a good idea to touch a metal object so that you discharge any static electricity that you have in your body.
6. There are three sets of screws that need

Continued on Page 40

QUIKDATA - 16 YEARS OF H/Z SUPPORT!

YOUR H/Z STORAGE & ENHANCEMENT EXPERTS!

ACCELERATE YOUR PC/XT/AT!

For your H/Z241, 248 and 386/16 we have the direct replacement **WESTERN IMAGING Z-33 ZUNI MOTHERBOARD** that's just loaded with features. Super fast 33Mhz 80386 main board with ten expansion slots **and** built in are two serial ports, one parallel port, floppy disk controller and an IDE hard disk controller! Add video (and your old MFM controller if you desire) and you still have plenty (9 slots) of expansion left! Support to 32 meg on-board RAM using SIMMs, and 32 additional megs with memory expansion.

WIZ-33 - $795 with 0K RAM **80387-33** - $195 Coprocessor
SIM1X9-8 $45 1 meg X 1 70ns SIMM DRAM
SIM256-8 - $13 256K X 1 70ns SIMM DRAM

From Sota Technologies, Inc., the fastest and most proven way to speed up your **H/Z150/160/150/158/159** series of computers. In many cases it will run faster than a standard IBM AT type computer! The EXP12 **286i** is a 12Mhz 80286 accelerator board with 16K on-board CACHE.
EXP-12 - $249

MEMORY UPGRADES

Note: All memory upgrades come without memory chips. 150ns 256K DRAMs are $1.45 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.

EAZYRAM - $89 Upgrades EaZy PC from 512 to 640K

ZMF100 - $55 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 - $55 Similar to ZMF100 above, but for new motherboards with p/n 181-4918 or greater.

3MB RAM BOARD for Z241/248 computers is an excellent memory card. Will backfill your 512 to 640K, and provide both extended and expanded RAM; all can coexist. Uses 100ns M256-10 RAM chips ($1.59 each), 36 per megabyte desired. Minimum of 18 DRAM chips required ($1.79 each).
EVATRD - $109

Z248/12, Z286LP RAM UPGRADE Z605-1 consists of 2MB SIMM 80ns RAM kits to upgrade your H/Z systems.
Z605-1 - $110

Z386/20, Z386/25, Z386/33, Z386 EISA 2MB SIMM 80ns UPGRADE to add increments of 2MB to these systems. Two required.
ZA3600ME - $79

ZA3800MK - $245 4 megabyte SIMM upgrade for above. Must have 4-1 meg SIMMs installed first.

We also have memory upgrades for just about every H/Z desktop and laptop unit except the 181, 183, and 184 series. Tell us what you need and we can quote you a price.

HARD DRIVE & TAPE UPGRADE KITS

PCW40 - $319 Complete MFM winchester setup for a H/Z150, 148, 158, 159, 160, PC etc. Includes **42 meg** formatted half-height Segate ST-251 28ms drive, controller, cable set, doc.
ST-251-1 - $269 Bare drive only
PCW30 - $295
ST-138 - $249 Bare drive only
PCW80 - $459 80 meg with Segate ST-4096 full size drive.
ST-4096 - $415 Bare drive only

DTCON - $59 PC/XT hard drive controller board
WDATCONF - $95 1:1 interleave HD/floppy controller for AT's
IDECON - $69 AT bus IDE/floppy controller for placing an IDE hard drive in any AT compatible.

TAPE BACKUP UNITS that work off the floppy controller in any PC/XT/AT computer from CMS, the DJ-10 (120MB unit) and DJ-20 (250MB) units. Low cost and impressive. Uses DC2000 series cartridges.
DJ-10 - $239 **DJ-20** - $295

FLOPPY DRIVE SAMPLE

MF501	- $71	5" 360K DS/DD drive
MF504	- $75	96 TPI 1.2 meg AT/Z100 drive
MF353	- $71	720K 3.5" drive in 5" frame
MF355	- $75	1.4 meg 3.5" AT drive in 5" frame
TM100-2R	- $65	40tk DS refurb (H8/89/Z100 PC type)

ANY TYPE FLOPPY IN YOUR PC/XT/AT

With the **CompatiCard**, you can install up to four additional drives in your PC/XT/AT computer. Add a 1.2 meg, a 1.44 meg, or any other drive, including 8" to any PC with an expansion slot. 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. CCARD4 also supports 2.8MB 3.5" floppy drives. Additional cables and external enclosures may be required.
CCARD2 - $79 **CCARD** - $99 **CCARD4** - $119

OR, add a floppy to any laptop or PC with a parallel port easily and inexpensively. With **Backpack**, you simply connect the external unit to your parallel printer port (do not lose printer function), install software, and away you go! No expansion boxes needed for laptops, and no slots required. Want a 2.8mb/1.4mb/720K floppy on your MinisPort? Want to add a 1.2 meg to your laptop? Want to add an additional drive to your desktop? Plug it in and go. 2.8MB 3.5" version will read and write 2.8 meg, 1.4 meg, and 720K format.

BPACK2.8 - $279	**BPACK1.4** - $229
BPACK1.2 - $229	**BPACK360** - $229

ADD AN EXTERNAL 80MB TAPE BACKUP DRIVE to any laptop or PC with the Microsolutions Backpack QIC-80 tape backup system. Plugs into a parallel port to give you an affordable way to easily backup your hard drive and/or transfer data. Uses DC2000 tapes. Fast! BPACKT8 - $445

ADD AN EXTERNAL HARD DRIVE - Just like above, but these units are completely portable hard drives. Think of the uses, including the security! 40 and 100MB units available by Microsolutions.
BPHD40 - $429 **BPHD100** - $575

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. Of course we carry much more. Find out!

OTHER STUFF

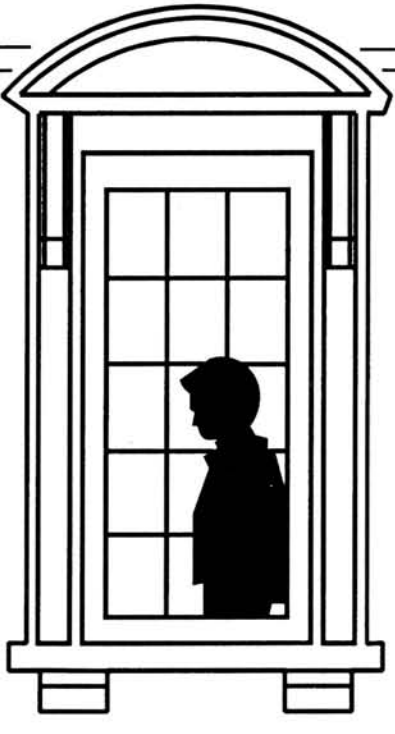
Quikdata also stocks ROM upgrades and batteries for most H/Z PC/XT/AT computers, spike protection filters, backup power supplies, tape backup units, modems, printers, cables and ribbons, disk drives and diskettes of all types, external hard drive and floppy drive enclosures, cables and connectors, video monitors and cards, memory chips and cards, ICs, joysticks, accessory cards, mice, software and much more! **Need a PC/AT computer?** Tell us what you want and we will quote you a price on one of our custom assembled QD computers made up to meet your demands.

Call or write in to place your order, inquire about any products, or request our 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 add $5 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

Looking Through WINDOWS at DR DOS Operating



Thomas R. Fox
5426 S. Gordon Avenue
Newaygo, MI 49337

As you will quickly discover, I could use some help in my grammar and punctuation. For instance, I have been unsure for years when a semi-colon was preferred over a comma and when it was all right to use a dash. Or how about the ellipse? When . . . ah . . . should that . . . you know . . . ah, ah . . . those dots be used? Also, I have fallen in love recently with Bartlett's Familiar Quotations. My favorite quotation is, "There are more things in heaven and computers, Horatio, than are dreamt of in your philosophy." I might not have gotten that just right, but it's close enough.

I discovered recently that a professional English tutor, with a rather bizarre name but great recall for famous quotations, could be bought for a flat fee of under $80. He didn't even charge by the hour! His name is Mr. C. W. Toolkit. After I paid the fee I found out it wasn't a person after all, but a software program called *The Complete Writer's Toolkit*! Was I shocked! Luckily I already had a computer to use the software on! The only problem was that the software required at least 5 Megabytes of space on a hard disk and my H-5100X computer, running MS-DOS 4.0 with Windows 3.0, had only 6 Megabytes left on its 42 MB hard disk – the numbers were too close for comfort.

There were many options available; the cheapest being the deletion of a number of files and programs. If you ever looked at my basement or garage, you'll quickly discover I hate to throw anything out, and this includes files stored magnetically! (While it may be a cop out, I do have a logical basis for this 'squirrel syndrome'. In the past, I remember dashing around the place looking for something. My wife would see me in my nearly senseless state and say, matter-of-factly, without asking a question, "You threw it out over a month ago, dude.")

The next cheapest alternative was to use one of those file compression utilities. When I heard about DR DOS 6.0, I really became excited when I realized: 1) It could be purchased for around $65, and 2) it included SuperStor which was licensed to it by Addstor, Inc. In addition, it supposedly had even better memory management capabilities than MS-DOS 5.0 (which apparently is its biggest selling point). Its only obvious drawback was that, unlike MS-DOS 5.0, it did not come with QBASIC. In fact, it didn't come with any BASIC whatsoever. For some, this may be sufficient to reject it. Since I have GW-BASIC from earlier versions of MS-DOS, this lack of BASIC was just a trivial annoyance.

When I received my copy of DR DOS 6.0 I was in a hurry to install it. I was in too much a hurry. I did not read the documentation thoroughly. This caused problems later on. First, a little bit more about my computer. I have a Heath H-5100X (16 MHz, 386SX) with a 1.2 MB 5-1/4" disk drive in drive A: and a 1.44 MB 3-1/2" disk drive in drive B: along with a 42 MB hard disk in drive C:. Also, it has 3 MB of total memory configured as follows according to the H-5100 monitor's setup screen: MAIN RAM, 640k BASE, 2048K EXTENDED, 288K EMS. (Note: By the time you read this I will probably have filled the MAIN RAM with 8 MB.) I didn't know it before I purchased it (sight unseen via phone), but DR DOS 6.0 requires installation from drive A:. Since I prefer the substantial feel of 3-1/2" disks, I told the salesman to send me the box with 3-1/2" disks. However, my A: drive is 5-1/4" which means I had to first copy all four 3-1/2" disks to 5-1/4" disks before I could install DR DOS. (The reasoning behind the fact that relatively inexpensive computer games usually come with both types of disks, while relatively expensive programs, such as DR DOS, usually come with one type of disk, alludes me.) After DR DOS was installed I hurried to try out SuperStor. (As mentioned earlier, I shouldn't have hurried.)

As instructed, I first used the CHKDSK command to check for any errors, and correct them, if necessary, using the /F option, before installing SuperStor. It is also important to backup all important data before installing SuperStor because a disk or power failure during the conversion process can be a real pain. The actual installation of SuperStor is painless. About the only thing I could tell that was changed, after it was installed, was the addition of another logical drive labelled D:. SuperStor produces an extended partition, and D: is the first logical drive of this extended partition. All files on logical drive C: had been compressed while files on D: remained uncompressed. The following files were put on my logical drive D: by the SuperStor install program: COMMAND.COM, DCONFIG.SYS, DEVSWAP.COM, DISKMAP.DAT, SSTORDRV.SYS and in subdirectory DRDOS (D:\DRDOS) is EMM386.SYS and HIDOS.SYS.

The reason for this is obvious once you understand that SuperStor is transparent to the user, which means it automatically compresses the data before writing to the disk and then automatically decompresses it when reading from it. For instance, if the COMMAND.COM file was a compressed file, how would it become decompressed? Keep in mind the compression and decompression is done entirely with software stored on a hard (or sometimes floppy) disk. If you don't fully understand the why here, it should be obvious from the evidence these files shouldn't be compressed; if a problem wasn't associated with compressing the files, wouldn't SuperStor compress them?

I was absolutely amazed by what SuperStor did with my nearly fully-loaded hard disk. Before running SuperStor I had only 6 megabytes free on my 42 megabyte disk, and after running SuperStor the CHKDSK command said I had nearly 40 megabytes free! It was mind-boggling! The 42 megabyte disk, which was nearly filled, now had nearly 40 megabytes of space left! My hard disk was born again! Of course, this 40 megabytes was only an estimate assuming the same compression ratio as before. Nonetheless, the SSTOR program's Statistics screen stated I had over 20 megabytes of actual free hard disk space left. The compression ratio (compressed space/actual space) was close to 2. One interesting note. After installing *The Complete Writer's Toolkit*, CHKDSK said there was only 30 megabytes left on the hard disk. Ten megabytes were missing; even though the Toolkit's manual claimed the program only took up 5 megabytes of space! Apparently, *The Complete Writer's Toolkit* files are about as compact as they can be.

I had installed DR DOS 6.0 and SuperStor after I had already installed Windows 3.0. MS-DOS 4.0 was my original operating system; DR DOS 6.0 replaced it. Everything seemed to work OK at first, although, of course, with a total of only 1MB of RAM, Windows was only operating in the REAL mode. However, when I installed an additional 2MB of RAM I started to have problems with Windows. The problems seemed to grow until the system was useless as it stood. I caused additional problems myself since I added the line "win.exe" to the autoexec.bat file in order to load Windows automatically at boot-up. You can imagine what happened when Windows went goofy. The only way to get out of Windows, which locked up, was by re-booting. (Pressing Ctrl-C did not solve my problem.) However, re-booting caused one to go directly into the locked-up Windows again and the only way to get out of that was by re-booting but. . . you know. . . (that's another good spot to include an ellipse!) I have to admit that I was tempted to start from the beginning: de-compress the hard disk and re-install MS-DOS. However, if I did that I would lose all the data on my hard disk. Even if I stored the data temporarily on a case full of floppy disks I wouldn't have enough room on my uncompressed hard disk to again store all the data! I was like my computer system: A bona fide mess. After I calmed down and the neurons started clicking again, I got out my floppy disk copy of DR DOS and inserted it into drive A:. Then I pressed the ALT, CTRL and INS keys together and entered the H-5100X monitor program. Next, I typed BF0 and pressed enter. Well, I got the computer useable, but was wise enough not to immediately start Windows again.

Down-To-earth Computer Tip Number 1: Do not have snake-like, entangled super-programs, such as Windows, start up automatically; they might just turn on you and bite when you are must vulnerable.

Don't get me wrong here, I do love Windows and continue to use it. However, it is a complex program and I am just learning some of its intricacies. (By the way, the only reason I originally set it up to start automatically was to make it easier for my kids to run. There often seems to be a line, and sometimes a fight, waiting to use it. However, the computer is normally running during the day, of every month that has an R in it, from before sun up to past 9 anyway; and even a 5 year old can type 'WIN' and press return!)

After everything seemed to be working, I used DR DOS' ViewMax to delete all Windows files. (I was thinking of not re-installing Windows but using ViewMax and possibly TaskMax in its place. However, Windows is prettier!) I then decided to try to avoid problems and be logical in re-installing Windows. First, I again installed DR DOS 6.0. From DR DOS' Setup menu, I installed all default options (this is simply accomplished by pressing the ENTER key) except for when it asked, under the section DiskMax, "Do you run Windows in Standard or Enhanced Mode?". The default is NO. You should positively change this to YES if you really are interested in working with Windows, rather than just playing with it. Another related point here; if you respond NO (the default setting) to the question "Support Windows In Standard Mode?" you will not be able to run Windows in Standard mode. If you answer "YES" to this question, you'll be able to run in Standard mode, but upper memory will then be disabled, which means you won't have as much conventional memory available. (Answering YES makes SETUP add the /WINSTD option to the EMM386.SYS device driver in the CONFIG.SYS (or DCONFIG.SYS) file. I found that, for my particular system, the /WINSTD option decreased my conventional memory by more than 100 KB!)

Down-To-Earth Computer Tip Number 2: The more memory the better, and this applies to the 640K curse (Conventional Memory) as well as the Shining Knight (extended memory).

I am starting to think that if you use Windows, you never have enough memory, never. Nonetheless, if you watch what you are doing with Windows (don't have more than 2 or 3 applications running at a time) 8 MB may be OK, maybe. Humorous, but sometimes obnoxious things seem to happen when systems start to run low on memory. The simplest way to achieve a maximum amount of available conventional memory is to put TSRs and the operating system kernel into upper (between 640K and 1000K) or extended memory. DR DOS automatically relocates the operating system kernel in either upper or high memory (the first 64K of extended memory). It also provides optional device drivers to take care of TSRs.

Down-To-Earth Computer Tip Number 3: If you don't use it, don't load it!

Example. I have a Logitech Serial 3-button Mouse which came loaded with drivers and utilities. I used the simple installation program that came with the mouse. After noticing my free RAM was down to around 570K, even after installing DR DOS and running their MemoryMax software (without MemoryMax free RAM was down to 470K), I decided to look at the AUTOEXEC.BAT file. I removed 'click' and 'logimenu' from the file which gained me around 40K of memory. Most importantly, I haven't noticed any change in the mouse's performance! The features these TSRs provided may be nice, but I didn't need them.

My next step was to re-install Windows. This is time consuming, but simple; the setup program takes care of the details for you. If you don't watch out, the Windows setup program will change the CONFIG.SYS file to include the Windows HIMEM.SYS and SMARTDRV.SYS device drivers. After installing Windows, use DR DOS' Editor to check that these drivers aren't included in the CONFIG.SYS file. If they are, remove them.

Down-To-Earth Computer Tip Number 4: Look at your system's CONFIG.SYS and AUTOEXEC.BAT files and try to understand what they mean. HOWEVER, DO NOT RUSH TO CHANGE ANYTHING.

If you do plan on changing either the CONFIG.SYS or AUTOEXEC.BAT files, first store the original files on a floppy disk and physically remove them from the computer before you try out the new one. If everything gets screwed up you can re-install them and start at square one again. Remember, these files must be pure ASCII, so use caution if you are using a word processor. It may be smart to use DR DOS' Editor to alter these files. This text editor isn't sophisticated, but it is easy to use – not the pain in the brain like that "fabulous" line editor that came with your PC's operating system!

DR DOS' Instructional 'MEM' Command

DR DOS' MEM command is greatly enhanced over MS-DOS 4.0's. If you use the /A option you should also use the /P (pause) option, since you will get about 4 screen-fulls of information. The information is both detailed and specific (which can be confusing at first), but is repeated in a simplified graphical form. This visual display of the system memory's usage is not extremely detailed, but is easy to understand. While I haven't used it extensively yet, it is fairly obvious that if one studies the information provided by 'MEM', one can easily optimize your system's memory management. In addition, this is a simple, cheap and interesting way of learning more about the details of your computer system.

SuperStor and The Mystery of the Hidden File

As mentioned earlier, my primary incentive for purchasing DR DOS 6.0 was for the file compression program SuperStor. However, I also was interested in its memory management capabilities. After the first time I installed DR DOS I took a look at the CONFIG.SYS file. There was no EMM386.SYS listed (EMM386.SYS is a device driver to make better use of memory in '386- or '486-based computers.) I examined the documentation and it stated in black and white that EMM386.SYS should be on the CONFIG.SYS file. I then decided to put it there myself, thinking that somehow the stupid Install program missed it. After booting up I briefly saw the message to the effect "EMM386.SYS has already been installed." I was mystified. Where was it? A hidden file? While playing my violin and smoking my pipe, my mind studied the clues and came to the conclusion that there is a hidden file. It was elementary. This hidden file must have been created when I installed SuperStor. Checking into it some more I discovered its name is DCONFIG.SYS and it is on the uncompressed D: drive. The first thing DCONFIG.SYS does is install the driver EMM386.SYS, along with its options. The next thing it does is load the SSTORDRV.SYS and the DEVSWAP.COM, and then it chains to the good old CONFIG.SYS file located on C:. If you examine the documentation carefully, you will discover DCONFIG.SYS is mentioned in a note on page 478 of the manual, but it isn't listed in the index or mentioned in the glossary. It isn't really that important, although it can be confusing, since only the CONFIG.SYS file is talked about in the manual and DCONFIG.SYS is left to a single small note.

A Mistake I Made That You Can Learn From

A Windows 3.0 permanent swap file can be handy. However, permanent swap files cannot reside on a compressed volume. While SuperStor does automatically create a logical drive where it places uncompressed data (usually D:) it doesn't reserve enough room on it for a permanent swap file. Apparently, Windows requires an absolute minimum of 2048 KB (2MB) in order to create a permanent swap file. Keep these facts in mind when first installing SuperStor, since you have the option of only compressing a part of the hard disk. If you choose the default installations and recommendations, like I did, you will have a problem later if you really desire to create a permanent swap file. Note: If you stuff your computer full of memory, there probably will never be a need for any type of swap file.

I strongly recommend that, at the initial installation of SuperStor, you leave half your drive in uncompressed form. If you need more space you can compress it later. If your disk is so loaded that you can't leave half of it uncompressed, the minimum uncompressed space I recommend is 10% of your drive size. For instance, I probably would be in a lot better shape now, and definitely have a more versatile system, if I left 5 MB uncompressed. Don't over pack your disk though. The conversion process requires at least 1.5 MB of free space on the disk.

Another thing to keep in mind before installing SuperStor is that many operating systems, including earlier versions of DR DOS, do not support compressed drives. In certain circumstances this could be a problem. Also recall that if you decide to uncompress a drive all data on the drive will be lost. This is another reason to back-up the data on the drive before installing SuperStor. This brings up:

Down-To-Earth Computer Tip #5: Don't be in a hurry! Study the documentation thoroughly before you change anything on your system. *DON'T DO AS I DO, DO AS I SAY!*

AND YOU THOUGHT THE TERMS ENHANCED MEMORY AND EXTENDED MEMORY WERE CONFUSING. DID YOU KNOW DISK COMPACTION AND DISK COMPRESSION REFER TO TWO DIFFERENT FUNCTIONS ENTIRELY?

Although they sound similar in function, a disk-compaction utility (e.g., DISKOPT) differs in principle and function from a disk-compression utility (e.g., SuperStor). A disk compression program increases your disk capacity by the use of data compression algorithms, as well as a super efficient sector allocation scheme. (With SuperStor space is reserved in small sector units of usually 512 bytes, while standard DOS reserves space in 2048 to 4096 byte units.)

Disk-compaction utilities do not increase disk space. Instead they make reading and writing to your hard disk much more efficient by reorganizing your disk so the information in each file is stored as close together as possible. This is visually demonstrated when you run DISKOPT; with dots representing data and a shading representing empty space. After completion, all the empty space is at the bottom of the screen. Disk-compaction utilities are also referred to as defragmentation utilities or disk organizers. While these other terms are less confusing (the confusion stems from the fact there is such an animal as a disk compression utility), the Windows user guide refers to these utilities as disk-compaction. Let's face it. Windows is big. And when Windows talks, people listen. The Windows User's Guide strongly recommends the use of a disk-compaction utility immediately before installing Windows, but they don't supply the utility with Windows.

Several tips here if you do use DISKOPT. First, make sure you run directly from DOS (definitely not from the Windows DOS prompt!), and make sure no other program uses the disk while running DISKOPT. To be safe, re-boot the computer and go directly to the DOS prompt. Second, run CHKDSK and correct any errors it finds. Third, if you don't have your computer connected to a UPS (Uninterruptable Power Supply), backup the disk before running DISKOPT. (Of course, if there are small children in the house don't leave your computer unattended while running DISKOPT. You may be tempted to do this since it sometimes takes quite a bit of time to run.)

The SHIP Utility: The Remnant of the Zenith Data Systems Version of MS-DOS 4.0 I Still Use

DR DOS 6.0 does not come with a hard disk head parking utility. From what I have read and been told, as well as from personal experience, it is extremely important to use a head parking utility anytime the hard disk is moved. Also, it is apparently wise to park the head before you shut down your computer. I still use the SHIP utility which came with MS-DOS 4.0. Despite the fact that it came with the MS-DOS operating system, version 4.00.00 of the SHIP utility has a Zenith Data Systems 1989 copyright. Apparently, this utility moves the hard disk's read/write heads to the shipping cylinder, which makes the heads and platters less prone to damage. To run this command simply type SHIP and type Y to the question "Do you wish to continue: (Y/N)". The only thing here is that if you answer Y, no further operation is possible on your computer and you must shut it off.

A Brief Look at *The Complete Writer's Toolkit*

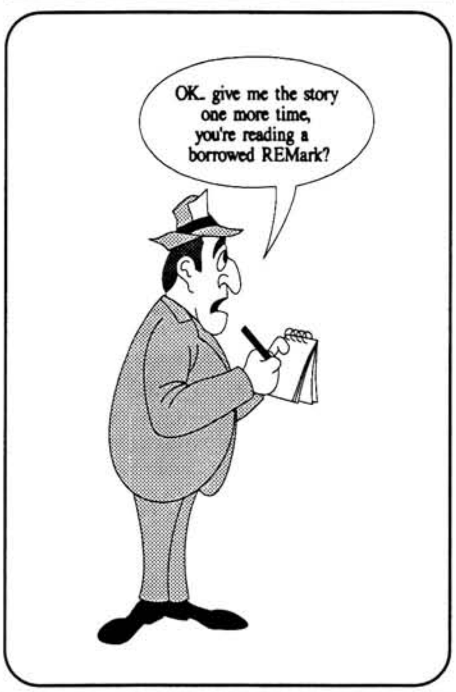
As explained earlier, the desire to use *The Complete Writer's Toolkit* was the reason for my purchase and use of DR DOS 6.0. This Toolkit came with an installation program that made it easy to install, despite the fact that it came with seven 3-1/2" diskettes. I presently use Textra 6.0 as my principle word processor. It is relatively inexpensive, intuitive to use and has the option of producing ASCII and Word Perfect files in addition to other formats. Writer's Toolkit seems to be fairly compatible with Textra 6.0, although there are a few idiosyncrasies. More on this shortly. On the Toolkit's package is a list of 41 different word processors which they claim varying degrees of compatibility for. They do not mention Textra. The Toolkit is really a compilation of 6 separate programs they refer to as modules: the justifiably, highly rated American Heritage Electronic Dictionary, Roget's II Electronic Thesaurus, the interesting Concise Columbia Dictionary of Quotations, the on-line English reference book Written Word III, an abbreviation program, and the Houghton Mifflin Grammar and Style Checker.

There are several ways to handle loading the Toolkit. What I did was make a .BAT file consisting of WTK.EXE (the Toolkit) and TEXTRA.EXE (my word processor.) I named this file TEX.BAT so all I needed to do was type TEX and the Toolkit would load immediately before Textra. My actual TEX.BAT file looks like this:

```
CD C:\TEXTRA6
C:\WTK\CWT.EXE
C:\TEXTRA6\TEXTRA.EXE
```

(Here TEXTRA6 is the name of a subdirectory containing TEXTRA 6.0.)

When you load DR DOS' MemoryMax software (this is simply done by typing Y to the question that appears: "Load MemoryMax software Y/N?"), the Writer's Toolkit's menu appears in less than a second after pressing the key combination ALT-M. If you don't use MemoryMax software, it takes nearly 5 seconds for the menu to show up. Despite this time lag, you still might choose not to run MemoryMax software. More on this later. From the Toolkit's menu you choose one of the 6 different modules. However, if you choose "CorrecText Grammar and Style Checker," you are told to invoke the Grammar checker directly from DOS by typing GRAM. As it should, the ESC key brings you back to your word processor's editing screen. However, if you are using Textra 6.0 and try to go right back to the Toolkit's menu again before you hit another key, you must press ALT-M twice to get back to the menu. This is no problem (apparently an idiosyncrasy with Textra 6.0), although before I figured this out I thought my keyboard was starting to malfunction! In addition to using the menu, you can go directly to the module by using a Hot Key combination. As in most well-thoughtout programs, the key combinations are intuitive. For instance, to access the Dictionary use ALT-D, for the Thesaurus you use ALT-T, et cetera.


OK. give me the story one more time, you're reading a borrowed REMark?

One of my primary incentives to purchase the Toolkit was to use the Thesaurus, despite the fact that Textra 6.0 comes with an extensive Thesaurus in addition to a great spelling checker. (You can never have enough words to choose from!) The Toolkit's Thesaurus has a listing of usages which makes choosing the right word for the right occasion easier than ever. According to the box, there are 500,000 synonyms for 42,000 words. Sounded like plenty to me when I purchased it! While the Toolkit's Thesaurus is excellent in design and usage, I feel it still lacks size (needs more words) as well as pizzazz (as far as I can determine it lacks slang and is sparse on colloquial or conversational words and phrases). The Thesaurus allows one to automatically insert the replacement word in your word processing file. Simply prepare for it by pressing TAB and then use the up/down arrow to choose the word which is highlighted. Once the word is chosen press the INSERT key. If you did not load MemoryMax software, there is no problem. With MemoryMax installed pressing INSERT will cause the replacement all right. However, if you are using Textra 6.0, the last letter of the replacement word will continue to be repeated until a key is pressed. (If you didn't press a key, I have a feeling this repetition of letters wouldn't end until all the memory was used up!) This weird problem may only show up in Textra 6.0; it does not occur in Textra 5.2 or DR DOS' Editor.

While I don't use *The Complete Writer's Toolkit* as much as I thought I might (perhaps since the Thesaurus isn't exactly what I was looking for), I do find it quite useful and reassuring to have. I especially like the authoritative Dictionary and am pleased with the Dictionary of Quotations, although I wish it included more quotations from Shakespeare. (How about a Dictionary of Shakespeare's Quotations?) I am pleasantly surprised by the use I give the English reference book (Written Word III.) This module is chuck full of information and help. Just a few things it has that you wouldn't expect: A great guide to the metric system, a currency conversion table (Did you know that in Ethiopia there are 100 cents in an abirr, or in Papua, New Guinea there are 100 toea in a kina?), National holidays, symbols and signs, lists of important Dictionaries and Encyclopedias and even a list, with authors, of books on computers and word processors! There is much more.

While I have used grammar checkers before, I am not a big fan of them. They just seem too restrictive. (Especially since the critical phrase "Passive Voice" shows up more than I like to admit when a grammar checker is checking my writing.) The grammar checker that comes with *The Complete Writer's Toolkit* may not be the best (it may be, I just don't know), but it definitely is comparable to anything on the market. I would recommend trying out the Toolkit's grammar checker for awhile before you purchase another single-purpose grammar checker program. If you hate it, at least you have several other modules that you'll probably like to use.

If you do much writing, but don't have an English professor for a spouse, I have to recommend *The Complete Writer's Toolkit*. While I can't guarantee it will improve your writing, it should make you a bit more confident.

Products Mentioned

The Complete Writer's Toolkit
Version 1.1 $79.00
Multiple Zones International, Inc.
18005 NE 68th Street, Suite A110
Redmond, Washington 98052
1-800-258-8088

DR DOS 6.0 $64.95
Computability
P.O. Box 17882
Milwaukee, Wisconsin 53217
1-800-558-0003

TEXTRA 6.0 Call for Latest Price
Ann Arbor Software
345 S. Division
Ann Arbor, MI 48104
1-313-769-9088

WINDOWS 3.0 and MS-DOS 5.0
Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, WA 98073-9717
Available through many dealers

Logitech's 3-Button Serial Mouse
Logitech Inc.
6505 Kaiser Drive
Fremont, CA 94555
1-415-795-8500
Available through many dealers

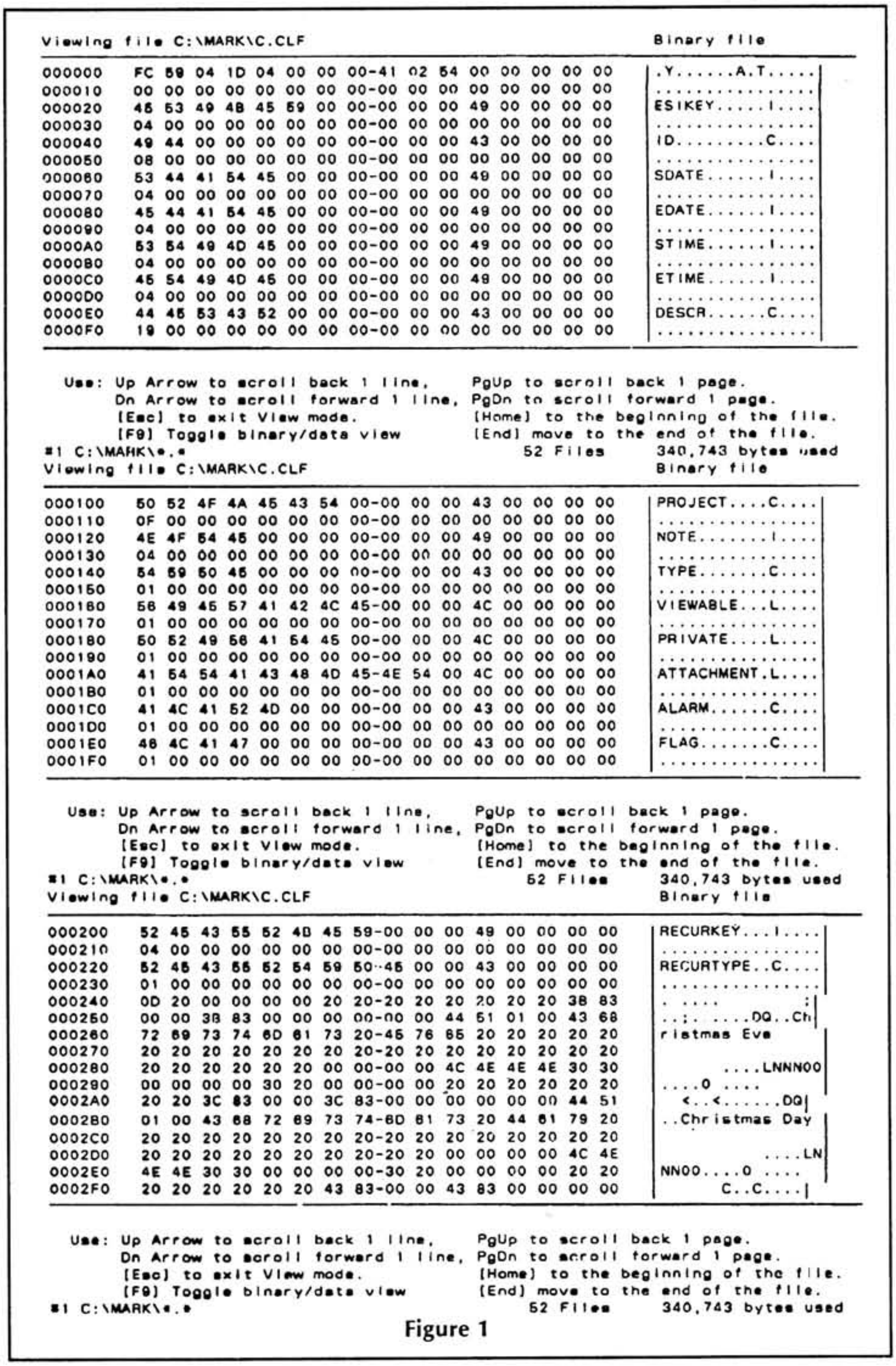
Printing ENABLE'S Gadgets

Mark Packard
2439 Legend Drive
Colorado Springs, CO 80920

One of the new features included in Enable's version 4.0 release is a category of desktop utilities affectionately known as Gadgets. These Gadgets can be found under the Tools selection found on the main menu bar. Enable provides the following utilities under the Gadgets category: a calculator, a cardfile database, a calendar and appointment book, a clock, and a telephone book (actually a predesigned cardfile). With these utilities, Enable has taken a big step toward filling some holes in an otherwise well-designed, integrated package. Unfortunately, Enable did not provide any way to print the calendar, cardfile, or telephone book. This article explains two utilities I developed to provide a way to print the data maintained by these Gadgets.

My original idea was to provide a rudimentary program to print a monthly calendar for my own use. However, as development progressed and "what ifs" were added, the utility became a polished product suitable for other users. The resulting program allows the user to select a range of dates to be printed and provides neatly formatted output for the user. The utility allows the range of dates to cross over into the next year. For example, the user can enter 10/1/91 and 10/31/91 to print the appointments for the month of October 1991 and the dates 12/1/91 and 1/31/92 will print all appointments for December 1, 1991 through January 31, 1992.

I originally planned to write the utility using Borland's Turbo Pascal. I later switched to the Mix Software, Inc. Power C compiler to make use of the many built-in functions provided by the C language. The program was developed on my Z-100 at home and debugged and refined using a Z-



```
Viewing file C:\MARK\C.CLF                                          Binary file

000000    FC 59 04 1D 04 00 00 00-41 02 54 00 00 00 00 00    |.Y......A.T.....|
000010    00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
000020    45 53 49 4B 45 59 00 00-00 00 00 49 00 00 00 00    |ESIKEY.....I....|
000030    04 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
000040    49 44 00 00 00 00 00 00-00 00 00 43 00 00 00 00    |ID.........C....|
000050    08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
000060    53 44 41 54 45 00 00 00-00 00 00 49 00 00 00 00    |SDATE......I....|
000070    04 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
000080    45 44 41 54 45 00 00 00-00 00 00 49 00 00 00 00    |EDATE......I....|
000090    04 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
0000A0    53 54 49 4D 45 00 00 00-00 00 00 49 00 00 00 00    |STIME......I....|
0000B0    04 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
0000C0    45 54 49 4D 45 00 00 00-00 00 00 49 00 00 00 00    |ETIME......I....|
0000D0    04 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
0000E0    44 45 53 43 52 00 00 00-00 00 00 43 00 00 00 00    |DESCR......C....|
0000F0    19 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|

  Use: Up Arrow to scroll back 1 line,      PgUp to scroll back 1 page.
       Dn Arrow to scroll forward 1 line,  PgDn to scroll forward 1 page.
       [Esc] to exit View mode.             [Home] to the beginning of the file.
       [F9] Toggle binary/data view         [End] move to the end of the file.
#1 C:\MARK\*.*                                  52 Files      340,743 bytes used
Viewing file C:\MARK\C.CLF                                          Binary file

000100    50 52 4F 4A 45 43 54 00-00 00 00 43 00 00 00 00    |PROJECT....C....|
000110    0F 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
000120    4E 4F 54 45 00 00 00 00-00 00 00 49 00 00 00 00    |NOTE.......I....|
000130    04 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
000140    54 59 50 45 00 00 00 00-00 00 00 43 00 00 00 00    |TYPE.......C....|
000150    01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
000160    56 49 45 57 41 42 4C 45-00 00 00 4C 00 00 00 00    |VIEWABLE...L....|
000170    01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
000180    50 52 49 56 41 54 45 00-00 00 00 4C 00 00 00 00    |PRIVATE....L....|
000190    01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
0001A0    41 54 54 41 43 48 4D 45-4E 54 00 4C 00 00 00 00    |ATTACHMENT.L....|
0001B0    01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
0001C0    41 4C 41 52 4D 00 00 00-00 00 00 43 00 00 00 00    |ALARM......C....|
0001D0    01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
0001E0    46 4C 41 47 00 00 00 00-00 00 00 43 00 00 00 00    |FLAG.......C....|
0001F0    01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|

  Use: Up Arrow to scroll back 1 line,      PgUp to scroll back 1 page.
       Dn Arrow to scroll forward 1 line,  PgDn to scroll forward 1 page.
       [Esc] to exit View mode.             [Home] to the beginning of the file.
       [F9] Toggle binary/data view         [End] move to the end of the file.
#1 C:\MARK\*.*                                  52 Files      340,743 bytes used
Viewing file C:\MARK\C.CLF                                          Binary file

000200    52 45 43 55 52 4B 45 59-00 00 00 49 00 00 00 00    |RECURKEY...I....|
000210    04 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
000220    52 45 43 55 52 54 59 50-45 00 00 43 00 00 00 00    |RECURTYPE..C....|
000230    01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    |................|
000240    0D 20 00 00 00 00 20 20-20 20 20 20 20 20 3B 83    |. ....        ;|
000250    00 00 3B 83 00 00 00 00-00 00 44 51 01 00 43 68    |..;.......DQ..Ch|
000260    72 69 73 74 6D 61 73 20-45 76 65 20 20 20 20 20    |ristmas Eve     |
000270    20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20    |                |
000280    20 20 20 20 20 20 00 00-00 00 4C 4E 4E 4E 30 30    |      ....LNNN00|
000290    00 00 00 00 30 20 00 00-00 20 20 20 20 20 20 20    |....0 ....      |
0002A0    20 20 3C 83 00 00 3C 83-00 00 00 00 00 00 44 51    |  <..<......DQ|
0002B0    01 00 43 68 72 69 73 74-6D 61 73 20 44 61 79 20    |..Christmas Day |
0002C0    20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20    |                |
0002D0    20 20 20 20 20 20 20 20-20 20 00 00 00 00 4C 4E    |          ....LN|
0002E0    4E 4E 30 30 00 00 00 00-30 20 00 00 00 00 20 20    |NN00....0 ....  |
0002F0    20 20 20 20 20 20 43 83-00 00 43 83 00 00 00 00    |      C..C....|

  Use: Up Arrow to scroll back 1 line,      PgUp to scroll back 1 page.
       Dn Arrow to scroll forward 1 line,  PgDn to scroll forward 1 page.
       [Esc] to exit View mode.             [Home] to the beginning of the file.
       [F9] Toggle binary/data view         [End] move to the end of the file.
#1 C:\MARK\*.*                                  52 Files      340,743 bytes used
```

Figure 1

248 and the Power Ctrace multi-window debugger. The Power Ctrace debugger gives the programmer the ability to view up to four windows of information simultaneously. The programmer can choose between source code, data variables, assembler code, logical watchpoints, output, memory, and symbols windows. For my purposes the source code and data variables windows made it a simple task to isolate and correct run time errors. Throughout the program I attempted to stay as close to standard C as possible to ensure the program's compatability with other compilers. The only include files used are STDIO.H used for basic I/O and STRING.H required to declare the string functions used in the program.

While explaining the algorithms used, I will point out some of the features of C that may not be obvious to some readers. Throughout the article all function and variable names will be printed in bold type. For example, **prtline** represents the variable name associated with the array. The functions **MonthDay**, **MonthName**, and **DayofYear** were adapted from Brian Kernighan and Dennis Ritchie's *The C Programming Language*. These functions are efficient, yet easy to implement at the same time.

Data Analysis

Soon after learning I could not print the appointments stored in my calendar file, I set out to solve the problem. I first tried to print the data file with little success. This inability to print led me to believe the file was in binary format. Fortunately, Enable provides a binary file viewing capability as part of their file manager option. Actually, any utility (including DOS's DEBUG) capable of displaying binary data will do. Using the View option found under the File manager item of the File pull-down menu, I was able to determine the record structure from the file header information (See Figure 1). From the header data I coded the following C structure:

```
typedef struct {
   long tsdate;
   long tedate;
   long tstime;
   long tetime;
   char desc[23];
   char trail[45];
} BINDATA;
```

The typedef identifier creates a user defined type called BINDATA. I can now create variables of this new "type" using the following statement:

```
BINDATA *rec;
```

which tells the compiler to create a pointer to **rec** that consists of data in the BINDATA format. An equivalent statement would be:

```
struct BINDATA rec;
```

but the latter declaration can get confusing when several structures are being used. The date variables are stored as long integers since the date information is stored using the year 1900 as the baseline ((1991 - 1900) * 365.25 = 33,268). Similarly, time is stored in seconds making these values large as well. The data found in the trail string is of no use to us so it is discarded.

The Program

A pseudocode outline for the utility is as follows:

```
Prompt for start and end dates
Read past header data
While not eof(data)
   Read a record
   Decode the date and time info
End while
Sort qualified entries by time
Sort entries by date
Print the sorted array
```

The command line for using the utility is:

```
prtcal filename.clf.
```

If the user omits the filename, he is prompted to enter a filename. If the file exists, the program continues by asking for the starting and ending dates to be printed. The **DayofYear** function takes the year, month, and day values and produces the corresponding Julian date. The **day_tab** two-dimensional array used by this function takes leap year into account automatically. To aid in understanding the process, I use the year 1991 as a baseline. For years greater than 1991 I simply add 365 days for each year greater than 1991 to arrive at the correct date.

Once past the initialization phase I begin to process the binary data file. The header data, useful during the analysis phase, is useless now so I scan past this information and then read the first record. From this raw data I want to convert the date and time information into an easy to read format. The **DATE**, **TIME**, and **ENTRY** structures are used to group the converted data into logical segments for printing. As most of the work occurs in the **Format** function, I will take the time to discuss the function in the next few paragraphs.

The **Format** function provides two valuable pieces of information. The first piece is the formatted record stored into **prtline[recno]**. This is made possible since **prtline[recno]**'s beginning address is passed as an input parameter to **Format** and, since **ent** is defined as a pointer to a structure of type **ENTRY**, any changes to **ent**'s structure items is really a change to the items found in **prtline[recno]**. The second useful piece of data provided by **Format** is a Boolean value telling me if the record is valid (i.e., falls within the specified dates).

Format uses the following equations to convert the raw date information:

```
tsyr = (int) (bin->tsdate / 365.25)
tsday = (int) (((bin->tsdate / 365.25) -
                  91) * 365.25) + 1.
```

Similar equations are used to decode **teyr** and **teday**. C allows the user to typecast values on the fly. The **tsyr** variable is of type integer but the result of division by a floating point is floating point so this result must be converted to integer using the (int) typecast. The **tsday** integer variable contains the date stored in Julian day format.

As I mentioned before, **Format** returns a Boolean value to the **main** routine. The function returns true if the starting date of an appointment falls within the dates entered by the user, and false otherwise. Before I can make this determination I must check to see if **stday** and **teday** are not equal. If this is true I replace **tsday** with **teday** because, when Enable stores repeated appointments, the starting date becomes corrupted. At this point I can now assess the record's validity with the following statement

```
if (tsday < stday || tsday > endday)
   return(0);
```

Next I check to see if the Julian date is greater than 365; if it is I normalize this value so that it falls between 1 and 365. Next, **MonthDay** takes the converted year and Julian day values and produces the month and day within that month. **MonthName** takes the integer month produced by **MonthDay** and returns the appropriate name of the month. These values are stored into their appropriate structure fields ready to be printed.

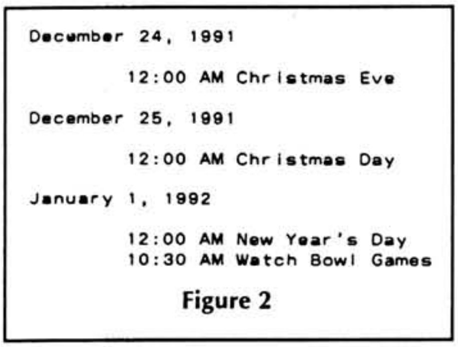
Once the date information has been formatted and stored into the appropriate fields in the **ent** structure I begin the necessary conversions for the time information. The time data is converted with these equations:

```
tshr = (int) (bin->tstime / 3600)
tsmin = (int) (((bin->tstime % 3600) /
                  3600.0) * 60.0.
```

The "%" operator is the modulus operator which returns the remainder of the division in integer format. For instance 34,200 % 3600 = 1800. However, I need this value expressed as a floating point to recover the minutes correctly. To convert this result to minutes I divide the remainder by 3600.0 to get the floating point value of 0.5 and then multiply by 60.0 to get 30.0 minutes. Again the final result is typecast as int since **tsmin** is an integer. The **StdTime** function is used to determine if the appointment occurs in the morning or afternoon (i.e., "AM" or "PM").

I use the fields **jdate** and **tstime** to store data to be used as sort keys once the array is completed. The **ent->jdate** field stores the Julian date for that record and the **tstime** variable stores the raw time in seconds. The statement **ent->desc[23] = '\0'** ensures the description string terminates properly. At this point, the **Format** function returns true and the array control variable **recno** is incremented.

Once the end of the data file is reached, the **prtline** array contains all of the appointments occurring within the specified dates. I use two separate iterations of the bubble sort algorithm to arrange the **prtline** array into the proper order. The first sort uses time as the key while the second sort key is the Julian date. This ensures the appointments occurring on the same day are grouped together in the correct chronological order.



```
December 24, 1991

          12:00 AM Christmas Eve

December 25, 1991

          12:00 AM Christmas Day

January 1, 1992

          12:00 AM New Year's Day
          10:30 AM Watch Bowl Games
```

Figure 2

Figure 2 is a sample of the output produced by the **PrintArray** function. A key feature of this function is the ability to send data directly to the printer using the **stdprn** device name. The **stdprn** parameter is a Power C feature that sends data to the parallel printer port of the computer. It is an extension of the standard files **stdin**, **stdout**, and **stderr**. If your compiler does not have this feature it will be necessary to create an output file to store the results in and then use the DOS print command to print this file. To do this, add the following line to the main variable declaration section:

```
FILE *outfile;
```

Next add the following code before the user prompt section:

```
outfile = fopen("CALENDAR.PRT", "w");
if (outfile == NULL) {
   printf("Can't open output file.\n");
   exit(1);
}
```

Finally, change the first parameter in the call to **PrintArray** from **stdprn** to **outfile**.

Printing the Telephone Book

Once the calendar printing utility was completed, I turned my attention to the Telephone Book cardfile. The data analysis followed the same steps as before, beginning with viewing the file in its binary format. I then created the following structure based upon my analysis.

```
typedef struct {
   char name[31];
   char fill1[21];
   char addr[31];
   char fill2[11];
   char city[31];
   char fill3[11];
   char st[21];
   char fill4[11];
   char zip[13];
   char fill5[9];
   char home[21];
   char fill6[8];
   char work[21];
} RECORD;
```

Unlike the calendar data file, the phone book data did not require any conversions, so the RECORD structure was the only one needed. In fact, the phone book utility was rather trivial compared to the calendar utility. Another difference in the two programs involved the interrecord data. Where this data had been included as part of the record in the calendar program, I chose to create a single character string, **filler** to hold the data between records. I chose the latter approach because the interrecord gap was significantly larger in the telephone data versus the calendar data (272 bytes to 45). A 100-item array would waste 27,200 characters. There is the added overhead of an extra call to **fread** for each record, but this becomes insignificant when compared to the amount of storage saved.

One new function, **putEOS** is introduced in the **prtphone** utility. **putEOS** places the null character (\0) at the end of each useful data field in the **RECORD** structure. This ensures that only the good data will be printed by the program. The C function **strtok** was useful for handling cities made up of more than one word. **strtok** breaks a string down into individual tokens or words. For example, if **entry->city = "Colorado Springs \0"** the result of the while loop in **putEOS** would be "Colorado Springs\0". This modified value is copied back into **entry->city** to properly format the city, state, and zip code on one print line. Without this modified value the spacing between these three fields was awkward, to say the least. Other than the **putEOS** function, **prtphone** execution is fairly straight forward. The command line input is: **prtphone filename.cdf**. Pseudocode for this routine is as follows:

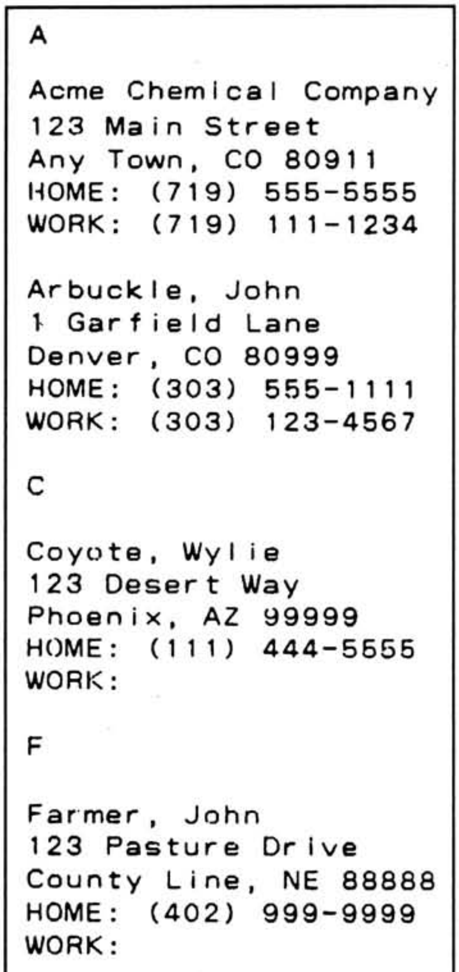
```
Read filename if not provided
Open data file as input
Read past header data
Initialize card counter to 0
While not eof(data)
   Read a record
   Read interrecord data
   Put end-of-string character
                   on useful fields
   Add 1 to card counter
End while
Sort the array with last name as key
Print the names and addresses
Close the data file and terminate
```

One assumption made is that names are entered last name first in the index field of the cardfile. This allows me to use Power C's **qsort** sorting function to sort the data items. The **qsort** function takes four parameters. The first parameter is the base address of the array to be sorted. Next comes the number of records to be sorted. The third value is the size of each record in the array. The final parameter is the function to be used by **qsort** to handle the comparisons. If the compiler you are using does not provide a built-in sorting function, the bubble sort code in prtcal can be substituted here. Simply replace the sort key with the **name** variable and you're all set.

The **PrintAddr** function arranges the addresses like those found in an address book. The user is given the option to either skip pages for each letter having an entry or to print without skipping pages. One possible modification to this function would include adding the printer control codes to make use of your printer's condensed fonts. For my personal use I take the output (See Figure 3) and reduce the image using a photocopier.

Final Notes

I selected a **MAX** of 100 for both the **prtline** and **card** arrays as that value suited my needs. I made this a constant for the express purpose of being modified to individual requirements. The bubble sort algorithm used in the calendar utility is not the most efficient, but for less than 100 items performance is not seriously degraded. During development I successfully tested both programs using a variety of command line inputs and starting and ending date combinations for the **prtcal** program. I have not experienced any problems since I began using them. This does not mean there are not any bugs in the code. If you encounter any problems with either utility, please let me know. I wrote these programs to satisfy my particular needs and welcome you to distribute the utilities to anyone who could use them.



```
A

Acme Chemical Company
123 Main Street
Any Town, CO 80911
HOME: (719) 555-5555
WORK: (719) 111-1234

Arbuckle, John
1 Garfield Lane
Denver, CO 80999
HOME: (303) 555-1111
WORK: (303) 123-4567

C

Coyote, Wylie
123 Desert Way
Phoenix, AZ 99999
HOME: (111) 444-5555
WORK:

F

Farmer, John
123 Pasture Drive
County Line, NE 88888
HOME: (402) 999-9999
WORK:
```

Figure 3

Listing 1 — PRTCAL.C

```
#include <stdio.h>
#include <string.h>
#define MAX 100
#define offset1  590

typedef struct {
      int day;
      char mon[10];
      int month;
      int yr;
   } DATE;

typedef struct {
      int hr;
      int min;
      char am_pm[3];
   } TIME;

typedef struct {
      long tsdate;
      long tedate;
      long tstime;
      long tetime;
      char desc[23];
      char trail[45];
   } BINDATA;

typedef struct {
      long srtime;
      int jdate;
      DATE sdate;
      TIME stime;
      TIME etime;
      char desc[24];
   } ENTRY;

int stday, endday, yr1, yr2;

static int day_tab[2][13] = {
   {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
   {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
};

int MonthDay(year, yearday, pmonth, pday)
int year, yearday, *pmonth, *pday;
{
   int i, leap;

   leap = year%4 == 0 && year%100 != 0 || year%400 == 0;
   for (i=1; yearday > day_tab[leap][i]; i++)
      yearday -= day_tab[leap][i]; _6á2
   *pmonth = i;
   *pday = yearday;
}

char *MonthName(n)
int n;
{
   static char *name[] = {
      "illegal", "January", "February", "March",
      "April", "May", "June", "July", "August",
      "September", "October", "November", "December"
   };
   return((n < 1 || n >12) ? name[0] : name[n]);
}

int DayofYear(year, month, day)
int year, month, day;
{
   int i, leap;
   leap = year % 4 == 0 && year % 100 != 0 || year % 400 == 0;
   for (i=1; i < month; i++)
      day += day_tab[leap][i];
   return(day);
}

int Normalize(julian)
int *julian;
{
   while (*julian > 365)
      *julian -=365;
}

int StdTime(hour)
int *hour;
{
   if (*hour == 0) {
      *hour +=12;
      return(0);
   }
   else if (*hour == 12)
      return(1);
   else if (*hour > 12) {
      *hour -=12;
      return(1);
   }
   return(0);
}

int Format(bin, ent)
BINDATA *bin;
ENTRY *ent;
{
   char *MonthName(); _6á2
   int MonthDay(), StdTime();
   int tsday, tsyr, teday, teyr;
   int tshr, tsmin, tehr, temin;
   tsyr = (int) (bin->tsdate / 365.25);
   tsday = (int) (((bin->tsdate / 365.25) - 91) * 365.25) + 1;
   teyr = (int) (bin->tsdate / 365.25);
   teday = (int) (((bin->tedate / 365.25) - 91) * 365.25) + 1;
   if (tsday != teday)
      tsday = teday;
   if (tsday < stday || tsday > endday)
      return(0);
   if (tsday > 365)
      Normalize(&tsday);
   MonthDay(tsyr, tsday, &ent->sdate.month, &ent->sdate.day);
   strcpy(ent->sdate.mon, MonthName(ent->sdate.month));
   ent->jdate = tsday;
   ent->sdate.yr = tsyr + 1900;
   tshr = (int) (bin->tstime / 3600);
   tsmin = (int) (((bin->tstime % 3600) / 3600.0) * 60.0);
   tehr = (int) (bin->tetime / 3600);
   temin = (int) (((bin->tetime % 3600) / 3600.0) * 60.0);
   ent->srtime = bin->tstime;
   if (StdTime(&tshr))
      strcpy(ent->stime.am_pm, "PM");
   else
      strcpy(ent->stime.am_pm, "AM");
   ent->stime.hr = tshr;
   ent->stime.min = tsmin;
   StdTime(&tehr, ent->etime.am_pm);
   ent->etime.hr = tehr;
   ent->etime.min = temin;
   strcpy(ent->desc, bin->desc);
   ent->desc[23] = '\0';
   return(1);
}

void PrintArray(fp, rec, norecs)
FILE *fp;
ENTRY rec[MAX];
int norecs;
{
   int i, linect;
   fprintf(fp, "\n\n\n\n\n");
   linect = 5;
   for (i=0;i<norecs;i++) {
      if (rec[i].jdate != rec[i-1].jdate) {
         fprintf(fp, "\n");
         fprintf(fp, "\t%s ", rec[i].sdate.mon);
         fprintf(fp, "%d, ", rec[i].sdate.day);
         fprintf(fp, "%d\n\n", rec[i].sdate.yr);
         linect += 3;
      }
      fprintf(fp, "\t\t%2d", rec[i].stime.hr);
      if (rec[i].stime.min == 0) _6á2
         fprintf(fp, ":00 ");
      else
         fprintf(fp, ":%d ", rec[i].stime.min);
      fprintf(fp, "%s ", rec[i].stime.am_pm);
      fprintf(fp, "%-23s\n", rec[i].desc);
      linect += 1;
      if (linect > 50) {
         fprintf(fp, "\f");
         fprintf(fp, "\n\n\n\n\n");
         linect = 5;
      }
   }
}

main(int argc, char *argv[])
{
   char c, inname[13];
   int DayofYear();
   int i, j, recno, dd, mm;
   BINDATA *item;
   ENTRY prtline[MAX], temp;
   FILE *data;
   void ReadRecord();
   int Decode();
   void PrintArray();
   if (argc < 2) {
      printf("Enter Calendar file to print: ");
      gets(inname);
      data = fopen(inname, "rb");
   }
   else {
      strcpy(inname, argv[1]);
      data = fopen(inname, "rb");
   }
   if (data == NULL) {
      printf("Can't open input: %s\n", inname);
      exit(1);
   }
   printf("Enter start day for print (mm/dd/yy): ");
   scanf("%d/%d/%d", &mm, &dd, &yr1);
   yr1 += 1900;
   stday = DayofYear(yr1, mm, dd);
   stday += (yr1 - 1991) * 365;
   printf("Enter last day for print (mm/dd/yy): ");
   scanf("%d/%d/%d", &mm, &dd, &yr2);
   yr2 += 1900;
   endday = DayofYear(yr2, mm, dd);
   endday += (yr2 - 1991) * 365;
   for (i=0;i<offset1;i++)
      c = getc(data);
   recno = 0;
   while (!feof(data)) {
      item = malloc(sizeof(BINDATA));
      fread(item, sizeof(BINDATA), 1, data); _6á2
      if (Format(item, prtline[recno]))
         recno++;
   }
   for (i=1; i < recno; ++i)
      for (j=recno-1; j>=i; –j) {
         if (prtline[j-1].srtime > prtline[j].srtime) {
            temp = prtline[j-1];
            prtline[j-1] = prtline[j];
            prtline[j] = temp;
         }
      }
   for (i=1; i < recno; ++i)
      for (j=recno-1; j>=i; –j) {
         if (prtline[j-1].jdate > prtline[j].jdate) {
            temp = prtline[j-1];
            prtline[j-1] = prtline[j];
            prtline[j] = temp;
         }
      }
   PrintArray(stdprn, prtline, recno);
}
```

Listing 2 — PRTPHONE.C

```
#include <stdio.h>
#include <string.h>
#define MAX 100

typedef struct {
        char name[31];
        char fill1[21];
        char addr[31];
        char fill2[11];
        char city[31];
        char fill3[11];
        char st[21];
        char fill4[11];
        char zip[13];
        char fill5[9];
        char home[21];
        char fill6[8];
        char work[21];
     } RECORD;

putEOS(entry)
RECORD *entry;
{

   char *wordptr;
   char temp[31];
   entry->name[30] = '\0';
   entry->addr[30] = '\0';
   entry->city[30] = '\0';
   wordptr = strtok(entry->city, " ");
   strcpy(temp, wordptr);
   while ((wordptr = strtok(NULL, " ")) != NULL) {
      strcat(temp, " ");
      strcat(temp, wordptr);
   }
   strcpy(entry->city, temp);
   entry->st[2] = '\0';
   entry->zip[12] = '\0';
   entry->home[20] = '\0';
   entry->work[20] = '\0';
}

PrintAddr(fp, add, nocards, paging)
FILE *fp;
RECORD add[MAX];
int nocards;
int paging;
{
   int i;
   char currndx, nextndx;
   i = 0;
   currndx = nextndx = toupper(add[i].name[0]);
   while (i < nocards) {
      fprintf(fp, "%c\n\n", currndx);
      while (currndx == nextndx) {
         fprintf(fp, "%s\n", add[i].name);
         fprintf(fp, "%s\n", add[i].addr);
         fprintf(fp, "%s, ", add[i].city);
         fprintf(fp, "%s ", add[i].st);
         fprintf(fp, "%s\n", add[i].zip);
         fprintf(fp, "HOME: %s\n", add[i].home);
         fprintf(fp, "WORK: %s\n\n", add[i].work);
         nextndx = toupper(add[i+1].name[0]);
         i++;
      }
   if (paging)
      fprintf(fp,"\f");
   currndx = nextndx = toupper(add[i].name[0]);
   }
}

main(int argc, char *argv[])
{
   FILE *data;
   RECORD card[MAX];
   char infile[12], filler[272], answer;
   int i = 0;
   int c, cardno, skip;

   if (argc < 2) {
      printf("Enter cardfile to print: ");
      gets(infile);
      data = fopen(infile, "rb");
   }
   else
      data = fopen(argv[1], "rb");

   if (data == NULL) {
      printf("Can't open input: %s", argv[1]);
      exit(0);
   }

   printf("Skip pages between groups? (Y/N)");
   scanf("%c", &answer);
   if (toupper(answer) == 'Y')
      skip = 1;
   else
      skip = 0;
```

Continued on Page 38

MS-DOS 5.0

Features & Configuration Tips

Ron Siebers
9334 Wentlock Road
Woodbury, MN 55125-3420

PC-DOS/MS-DOS History

Year	Version	Feature
1981	1.0	IBM PC Introduction
1983	2.0	PC Jr. Introduction
1983	2.1	IBM XT Introduction
1984	3.0	IBM AT Introduction, 1.2 MB drive support
1985	3.1	LAN Support
1986	3.2	720 K drive support, Xcopy & Driver.sys added
1987	3.3	PS/2's Introduced, 3.5" 1.44 MB drive support
1988	4.0	Shell, >32 MB HD Parts, EMS 4.0 support
1991	5.0	Improved Shell, loads high, uses less base memory, more

Microsoft and IBM have migrated through a long series of DOS releases since the original version created for the IBM PC. The table is a chronological history of DOS releases. One of the most significant previous DOS improvements was version 4.0 which first provided the ability to partition hard drives larger then the 32 MB limit originally imposed. MS-DOS 4.0 also had bugs that were corrected in version 4.01. Zenith Data Systems never issued a 4.01 because the bugs were corrected by the time it issued 4.0. The penalty of using DOS 4.0 was a significant increase in memory use that reduced the portion of 640K base memory available to load programs. The added DOS Shell program was klutzy, a memory hog and slow. ZDS and several other clone suppliers of custom DOS versions chose to add the larger partition feature to DOS 3.3. This preserved the leaner version for those using LAN software, which consumed enough memory in addition to DOS 4 that it prevented some programs from loading. ZDS called this version 3.3+ and recommended it over DOS 4.0 for most users.

By now, all registered ZDOS users have received an offer from ZDS to upgrade for $59.00. If you haven't, you can certainly order it using stock number OS-113-MS. The question is, should you? The next question is, if you do upgrade, can you properly configure it for your hardware and software? This article is intended to help you decide about upgrading and to pass on my sometimes painful experiences getting DOS 5.0 tuned for my 386 hardware and memory use, and then getting Windows 3.0 to work in enhanced mode again after I did that.

MS-DOS 5.0 Features

The new MS-DOS 5.0 has improved many areas and added many new features. The bad news is, you have to figure out how to use all these features on your computer setup. MS-DOS 5 simply copies files to your hard drive (or floppies if you change a default) and creates a default CONFIG.SYS. It does nothing to optimize for your particular computer hardware and memory. If you are loath to crack a thick DOS manual and read about the new features, you should either get help or forget it. If you are the adventuresome type, and open to trial and error, keep reading.

Users of 8086 PC's have little reason to upgrade. Read on to see why. Users of 80286 computers with only 640K are in the same category and will gain little benefit unless they can and do add more memory. 80286 machines having 1 MB or more of memory are capable of utilizing some MS-DOS 5.0 upgrade features. If you are using a 386- or 486-based computer, I strongly recommend upgrading. This is particularly true if you are or intend to use Windows 3.X or you are currently having problems getting enough base memory to load some programs. The major improvements of MS-DOS 5.0 over previous versions of DOS include:

- Ability To Load Device Drivers in the upper memory area—The DEVICEHIGH command can load certain device drivers into unused memory in the area between 640K and 1 MB. The amount of memory used by each device driver you can load into upper memory increases the amount of base (640K) memory available at the DOS prompt (after loading DOS) for programs.
- Ability To Load Part of DOS into the High Memory Area (HMA) — The DEVICE=HIMEM.SYS line reserves a 64K portion of the HMA starting at 1MB. The DOS=HIGH line loads a 64K portion of DOS into HMA. This increases available base memory by the same amount.
- On-line Help — You can call up help information for all MS-DOS and Shell commands.
- Improved Data Security — The UNDELETE command will restore a deleted file if used immediately after the deletion. The UNFORMAT command will undo a disk format, restoring the disk with its original files.
- Improved MS-DOS Shell — The Shell program has been substantially improved to provide file management using the mouse and allow you to create custom menus to load programs.
- MS-DOS Editor — a new full-screen text editor EDIT is provided in addition to EDLIN for editing ASCII text files including autoexec.bat, other batch files, and config.sys. Use of EDIT requires Qbasic be present in the same directory (usu-

ally called DOS).

- QBASIC — A smaller version of Microsoft QuickBasic that replaces Basic and Basica and is significantly improved. Qbasic includes a conversion program for the older basic files.
- Create Larger than 32MB hard drive partitions — You can now use FDISK to create hard drive partitions up to 2 gigabytes. Use of SHARE is no longer required as with DOS 4.0.
- File Search through multiple levels of directories — Subdirectories can be searched for a file.
- DIR Command Enhanced — New switches (i.e. /w) mean you can now list all filenames including hidden files, sort by type of file, date and time, or file size as well as alphabetical.
- DOSKEY is a program you can use to recall, edit and execute previous DOS prompt commands. It eliminates the need to retype commands executed repeatedly at the DOS prompt.
- Support for more than 2 hard drives — DOS now supports more than two physical drives. Any drive can have multiple partitions, each with a drive letter. An extended partition (in addition to the DOS partition) can be divided into as many as 23 drives.
- 2.88 MB 3.5" Floppy Drive Supported — DOS can now read, write and format the new 2.88 MB drive (4 MB floppy) standard.
- New Easier Setup Program — Guides you through installing MS-DOS 5.0 over previous versions of DOS.

Since 8088 based PC's and PC XT's cannot address more than 640K of memory, the MS-DOS 5 base memory increase features are not useable. MS-DOS 5.0 therefore offers little benefit to these computers. The ability to format larger than 32 MB hard drive partitions may be the only reason to upgrade. MS-DOS 5.0 uses about 1K less than MS-DOS 4.0 and about 10K more than MS-DOS 3.3 on these computers. This actually decreases the memory available to programs by 10K over MS-DOS 3.3 on machines without upper or high memory available. Another drawback is that COMMAND.COM has enlarged from version 4's 32K to 47K in size. MS-DOS 3.3 was only 25K. This should only be a problem if you need to format 360K bootable disks.

Before you buy MS-DOS 5.0, be aware that there are two upgrade versions available in stores. Microsoft MS-DOS 5.0 (called MS-DOS) will upgrade all IBM compatible computers, and will work on real IBM computers. The second version is from IBM. Each of these versions come on either 5¼" or 3.5" disks. You can only install the IBM version on computers running previous versions of IBM's DOS (PC-DOS). It checks the current DOS and aborts install if is not real IBM DOS. Also, the IBM version will only install from the A: drive, so you should buy the disk version that matches your A: floppy drive type. The MS-DOS version can be installed on any IBM or compatible computer including Zenith or a real IBM running IBM DOS. It can be installed from any floppy drive. Some computer manufacturers such as Dell and Zenith are offering custom versions of MS-DOS 5.0 and it is usually best to order their version to upgrade your computer. Zenith has simply retained the value added files it provided in MS-DOS 3.3+ and 4.0. This includes the nifty COMPACT program that reorders your hard drive to unfragment most files so they are stored in contiguous sectors on the hard drive so they load faster. You normally have to buy third party utility software such as Norton Utilities to obtain this capability. Zenith gives it to you with MS-DOS 5.0. Microsoft and IBM do not.

The MS-DOS 5.0 upgrade is strongly recommended for those having 80386/80486 computers who are using or intend to use Microsoft Windows because it offers Windows more memory. The primary improvement of MS-DOS 5.0 is the ability to free up 100K or more of base memory in these computers by loading files above 640K in memory. MS-DOS 5 accomplishes this in two ways. First, in computers with 2 MB or more of memory, 64K just above the 1 MB (1024K) address point can be reserved and a 64K chunk of DOS loaded there. That frees 64K of base memory. The extended memory (XMS) manager must be loaded using the command DEVICE=HIGHMEM.SYS. This activates reserved Expanded memory, and the DOS=HIGH loads part of DOS in the reserved high memory area. The second method is the DEVICEHIGH command for CONFIG.SYS drivers and LOADHIGH for AUTOEXEC.BAT files that load these into available memory between 640K and 1 MB. There is typically another 96K available there, although you usually can't use quite all of it. To use these "high" memory commands, there are two additional CONFIG.SYS statements required. They are DEVICE=EMM386.EXE and DOS=UMB. This second method is useable on some 80286 as well as all 80386/80486 computers. These two methods combine to make between 590K and 630K of memory available at the DOS prompt.

Installing MS-DOS 5.0 On Your Hard Drive

Installation on your hard drive is quite simple. You place Disk 1 in your A: drive and type "Setup". Then answer the menu questions to start installation. After that, you simply swap disks when prompted until installation is completed. Microsoft's version of MS-DOS 5.0 is quite intelligent at this process, renaming your old DOS directory to DOS_OLD and preserving your old DOS files. Zenith apparently chose to eliminate that feature. It also saves boot track information that will allow it to restore your old DOS using the Uninstall disks. The first thing you should do after this installation process is complete is to format a bootable MS-DOS 5.0 disk using the "C:\MSDOS5\Format A:\S\V" command from the C: prompt. This will be needed if you later are unable to boot from your hard drive due to configuration errors. If your hard drive is now short of space, you can regain over a megabyte of hard drive space. If you do not intend to use the Shell program and or QBasic, you could delete these files to save hard drive space. One last note, SETUP will copy a file named "WINA20.386" into the root directory on 386 machines. This file must be present in the root to run Windows 3.0.

Dealing With Non-standard Hard Drive Partitions

Running setup may give an error message. MS-DOS 5.0 cannot install on drives partitioned using third party software to get around the 32 MB limit. This is likely for your computer if you are currently using MS-DOS 3.2 or earlier and have a hard drive partition larger than 32 MB. To test for a non-DOS partitioned hard drive you will need to make a bootable floppy disk from MS-DOS 5.0. Run SETUP and use the "Install to Floppy" option in the Install menu. Have plenty of disks available and make sure they are high density if the floppy drive is. The installation program assumes this and automatically formats the disks high density, a danger if they are double density. Next, switch to your MS-DOS 5.0 floppy drive and type "Format X:/S/V" to format a floppy from your MS-DOS 5.0 disk. Then copy FDISK.COM and FORMAT.COM to your bootable floppy. You will need to back up the hard drive files because the following will erase all files. Next run FDISK from your current DOS version and remove the current partitions. Then, boot off your working MS-DOS 5.0 floppy, run FDISK and repartition the hard drive as large as you want it. After you exit FDISK, you can type "Format C:/S/V" for the Boot partition and "Format X:/V" for each additional nonbootable partition. The /V prompts you for a disk label, but is not necessary. Finally, you will need to create an MS-DOS 5.0 directory by typing "MD \MSDOS5". Then copy all the MS-DOS 5.0 files from the floppies. The setup program will not automatically install MS-DOS on a hard drive already formatted for MS-DOS 5.0. *(Editor's Note: Most MS-DOS 5.0 files are compressed. You will have to use the EXPAND command to uncompress them when you copy them to a new disk or directory. You will also have to know the proper filename* *extensions. For example, to copy the EGA.IN_ file, you must type EXPAND A:EGA.IN_ C:EGA.INI.)*

Understanding Memory

The illustration shows the basic memory structure of 386/486 computers. An explanation of terms is necessary to understand the different types of memory controlled by MS-DOS 5.0.

- Conventional Memory – The first 640K of RAM (Random Access Memory). Most DOS applications are limited to using the remainder of 640K after bootup.
- Expanded Memory – Paged memory above 640K that the computer hardware does not recognize. This memory requires a software driver called EMM (Expanded Memory Manager). Older 8088-, 8086- and 80286-based computers require special memory boards that provide the expanded memory. Newer 80286-, 80386- and 80486-based computers provide for Extended Memory. A driver such as EMM.SYS or EMM386.SYS for 386/486 computers that complies with the Expanded Memory Specification (EMS) established by Lotus-Intel-Microsoft is required to convert Extended memory into LIM-EMS Expanded memory.
- Extended Memory – Memory above the 1 MB (1,024K) point that cannot be used by applications directly. This memory is used for RAM drives (RAMDRIVE.SYS), hard drive cache (SMARTDRV.SYS) and applications that can control Extended Memory such as DesqView and Windows (HIMEM.SYS).
- Upper Memory – Usually called upper memory blocks, this is the area of memory between 640K and 1024K. This area was originally reserved as shadow RAM, an area to copy ROM code for faster computer operation. The computer ROM BIOS and the video BIOS are typically copied to areas of this RAM during boot-up of 386/486 computers. The unused portions of upper memory can be accessed by the DOS=UMB command.
- High Memory – The first 64K of Extended memory. Using an extended memory specification (XMS) driver, device drivers and TSR's can be loaded and run from this memory area. This leaves more conventional memory for programs to use. MS-DOS 5.0 uses this area via the DOS=HIGH command.

Configuring MS-DOS 5.0

Each computer setup can be different and MS-DOS 5.0 makes no attempt to optimize the installation. That is left up the user. The manual does offer help in the form of a chapter covering "Customizing Your System" and another on "Optimizing Your System". Both provide good sample CONFIG.SYS statements for each feature. Since I use a 386 computer both at work and at home, my experience is mostly limited to this type. The configuration of MS-DOS 5.0 on 386/486 computers with 2 MB or more of RAM requires a certain sequence of commands in the CONFIG.SYS and AUTOEXEC.BAT files to maximize the amount of base (640K) memory available at the DOS prompt for programs. Following are the command lists that I use for each file and brief explanation of why. You will note that I make extensive use of paths for these commands. Previous versions of DOS would not allow this and required these drivers to be in the root directory. MS-DOS version 3.3 and higher have allowed path statements for loading drivers that allow you to place them in directories for more orderly file storage. Using path names instead of relying on the AUTOEXEC.BAT PATH statement loads a little faster. A clean root directory will also speed up your computer operation slightly. It is therefore good practice to follow this command sequence. It is particularly important to use a path for the Device=EMM386.EXE command to assure proper loading of Windows into 386 Enhanced Mode to provide multitasking.

Once you have edited these two files to reflect MS-DOS 5.0 capabilities on your computer, reboot the computer to reload DOS with these changes. If your computer crashes during bootup or doesn't load files into high memory, a problem probably exists with the command sequence, improper command switches or lack of switches in your CONFIG.SYS file. Watch the screen for an error message that gives hints of which command caused the crash. To correct these files, boot up using a floppy, check your commands against the manual and edit the commands until you can boot properly. You can use the new DOS Edit program for this purpose. To verify that DOS loads into high memory, type MEM /C | More. The | is the shifted backslash on most keyboards. A list of all your device drivers, how much memory they use and where they are loaded (conventional or high memory) fills your screen. Press return to continue the listing until you see the DOS prompt. You should see a significant improvement in base memory available at the DOS prompt, typically more than 600K unless you use a lot of extra device drivers for special hardware that exceed the upper memory space available.

If you don't see enough improvement and, in particular, don't show any commands loaded in upper memory or see a "DOS loaded in High Memory" statement, check that HIMEM.SYS is listed first and that EMM386.EXE parameters are listed properly and you see a loading message briefly on screen during bootup. EMM386.EXE was the greatest source of problem for me since I wanted to use the RAM parameter to provide both EMS paged Expanded memory for my DOS programs and load drivers in the Upper Memory Blocks. Reserving EMS memory is apparently not compatible with Windows running in Enhanced (386) mode. Another problem with using upper memory blocks is possible overwriting shadow RAM where your computer hardware loads your ROM BIOS and Video ROM into high memory for faster operation (an automatic feature on Zenith 386/486 computers). MS-DOS 5 is apparently quite ignorant of this feature and will overwrite the memory causing your computer to crash. To find out if and where shadow RAM is being loaded, you will need a utility like Manifest from Quarterdeck. Manifest is available by itself, with QEMM386 or with DesqView/QEMM386 bundle. It will list what is resident in base and upper memory and what address range it occupies. A way to fix this is to use the / X (exclude) switch and specify the shadow RAM memory range, usually A000-AFFF. Chapter 11 of the MS-DOS 5 User's Guide is titled "Optimizing Your System". This is necessary reading to understand the options available and what they mean. Another source of help is the README.TXT file included with MS-DOS 5.0. You should print it out and wade through it for the couple of tidbits that may make a difference for your computer setup.

Using MS-DOS 5.0

Once you have your CONFIG.SYS and AUTOEXEC.BAT files working to your satisfaction, you should always make backup copies and a hardcopy printout. This will provide quick replacement when your hard drive fails and easy hardcopy reference if you need help. Then load your programs or Windows with little fear of that annoying "Insufficient Memory" message. The DOS Shell should be tried, particularly if you don't currently use any type of menu/shell program such as Norton Commander. For those using computers with a mouse, the ability to select a drive, directory and file and copy to another drive, directory or filename entirely with the mouse will be a real treat. Other improvements to check out are the added commands not present on your old version of DOS. If you prefer working from the DOS prompt, and have never used XCOPY before, give it a try. The /s switch allows you to copy a whole directory, all its sub-directories and the files in each to another drive or disk. It will even create the main directory if it doesn't exist on the target drive.

What About Windows?

If you are using Windows 3.0, you may wonder if you should use its version of the drivers HIMEM.SYS, EMM386.EXE,

CONFIG.SYS — A typical MS-DOS 5.0 CONFIG.SYS File Command List For 386/486 Computers

DEVICE=C:\MSDOS5\HIMEM.SYS /machine:X	If you have a clone such as Zenith Data Systems, check page 611 of the MS-DOS 5 manual or 15.22 of the ZDS version to see if your brand is listed. If it is, you should add /machine:X, where X is the number listed for your clone manufacturer. Zenith Data Systems is listed as 10.
DOS=HIGH	Loads a portion of DOS above 1 MB in the 64K XMS memory area. Requires a 2nd MB of RAM (Extended).
DOS=UMB	Tells HIGHMEM.SYS to control the 640K — 1024K area so you can use the DeviceHigh and LoadHigh commands to put drivers in the upper memory area.
DEVICEHIGH=C:\MSDOS5\EMM386.EXE	Expanded memory driver required to access the upper memory area, to convert Extended memory into Expanded (paged) memory required by some programs like Lotus 1-2-3, and/or to use the DeviceHigh and LoadHigh commands. Use the following switches as your needs dictate:
/auto	Enables expanded memory support only when a program calls for it.
/memory XXXX	Specifies the amount of Expanded (LIM EMS 4.0) memory in Kilobytes reserved for programs from Extended memory. This amount of Extended memory must be available after subtracting the 64K high memory DOS area. Remaining memory is available for Windows.
XXXX /ram	Provides support for both the upper memory area and XXXX kilobytes of Expanded memory created from Extended memory.
/noems	Specifies no Expanded memory reserved, only upper memory block support.
FILES=30	Increases the number of allowed open files to 30
BUFFERS=40	Increases the number of allowed buffers to 40.
STACKS=0,0	Decreases the default number of stacks to 0, saving a small amount of memory. If your programs require stacks, use the numbers the program recommends.
SHELL=COMMAND.COM /E:256 /p	Increases the environment memory reserved for device drivers from the default 128 bytes to 256.
DEVICEHIGH=C:\MSDOS5\ANSI.SYS	ANSI driver for DOS screen colors and keyboard escape sequences. Required by some programs.
DEVICEHIGH=C:\MSDOS5\RAMDRIVE.SYS XXXX /E	Creates a logical drive in RAM that can store data up to the size specified in XXXX. Be sure to use either the /A switch to use Extended (above 1 MB) or /E switch to use Expanded (EMS) memory. Note: Using no switch allocates the specified amount out of 640K base memory which you do not want to happen. A RAM drive can speed up certain programs that store temporary files on the hard drive. Assigning the temporary file location to a RAM drive instead speeds up some of these program operations.
DEVICEHIGH=C:\MSDOS5\SMARTDRV.SYS XXXX	A hard drive RAM cache of XXXX kilobytes that speeds up apparent file retrieval of repeatedly accessed files. Using no switch defaults to creation of the cache in Extended memory which is fastest.
/a	Specifies drive cache in Expanded memory. Use only if Expanded memory is hardware specified and no Extended memory is available.
DEVICEHIGH=C:\MSDOS5\SETVER.EXE	Loads a table of programs that are known to run only under previous versions of DOS and fools the program into thinking MS-DOS 5 is that version.

RAMDRIVE.SYS and SMARTDRV.SYS. Microsoft addresses that in the manual. You should always use the newer version of these files (file date). Since MS-DOS 5.0 has the newest versions right now, these are what you should use. Windows uses base memory as well as Extended Memory for both its files and the programs and their created files. The more of both types of memory available to Windows, the faster it will run. MS-DOS 5.0 provides more base memory for Windows while reducing the Extended memory by the 64K XMS area. The base memory will help Windows, particularly if you had been getting out of memory messages while loading Windows. Also, be sure the MS-DOS 5.0 file WINA20.386 file is present in the root directory to provide upper memory management for Windows. Configuring the EMM386.EXE gave me the most trouble. There is probably more I don't know about.

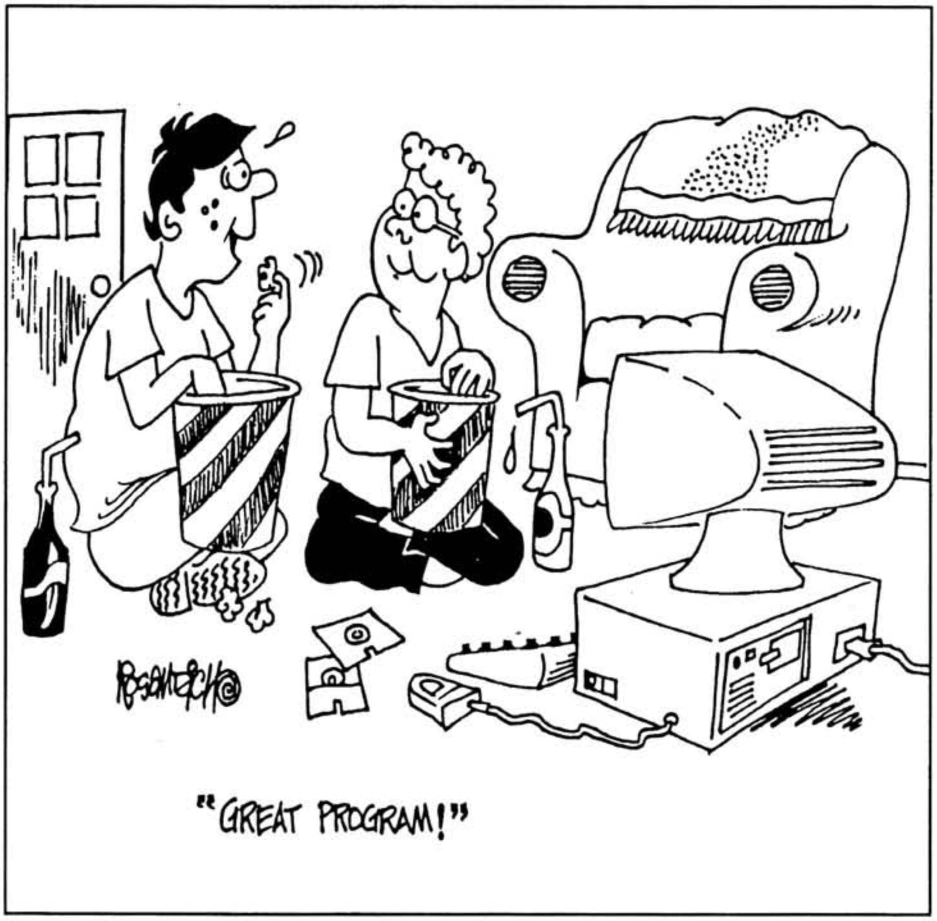
Final Comments

This article grew to be larger than planned. I've covered a lot of information that I hope will save you some of the frustration I

AUTOEXEC.BAT — Typical MS-DOS 5.0 AUTOEXEC.BAT File Listing

@echo off	Stops this and following commands from showing on screen.
Path=C:\;C:\MSDOS5;C:\BATCHES	Establishes the order of which directories DOS will search for a file to execute. Last command has no ";".
Prompt pg	Tells DOS to add the directory path you are in and a ">" to the DOS prompt.
SET TEMP=D:\WINDOWS\TEMP	Designates storing Windows temporary files in RAM drive D:. This provides faster Windows operation.
Copy Command.com D: Set Comspec=D:Command.com	These two command lines copy COMMAND.COM to a RAM drive and tells DOS to look for it there. This will speed up return of the DOS prompt after exiting a program when command.com must be reloaded into RAM. It loads faster from a RAM drive than any hard drive.
C:\UTILITIE\Numlokof.com	Runs a utility program previously discussed in REMark that turns off the Num Lock so that the keypad keys are in cursor mode after boot up.
LOADHIGH C:\UTILITIE\MOUSE\MOUSE.COM	Loads the mouse driver in upper memory.
LOADHIGH C:\NORTON\NC	Loads the Norton Commander (or any other named TSR program) in upper memory. Note: Loadhigh requires 1 MB of computer memory and Device=Himem.sys, DOS=UMB, and Device=EMM386.sys commands in the Config.sys file to work.

went through to get MS-DOS 5.0 properly configured. If all my troubles intimidates you, you might want to wait for MS-DOS 5.1. Microsoft says they will have an installation program that will check your hardware and automatically optimize your CONFIG.SYS file. If you need help with Zenith Data's MS-DOS 5.0, their manual instructs you to contact the dealer where you bought it. Results can be underwhelming in my experience. ZDS still refuses to provide user help directly. Interestingly, ZDS doesn't say where to get help if you buy the upgrade directly from them. If you buy the Microsoft version, you can call 206-646-5104 and give them your registration number from the inside back page of the user reference. They will then register you for 90 days of free support and immediately transfer you to a help-line person. ✻



MS-DOS 5.0

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

I have been getting lots of letters about upgrading to MS-DOS 5.0, and nearly everyone wants to know if it is something they should do. The real answer to that question depends on your particular situation. In some cases, this new release of MS-DOS may provide you with some features that you can really use, such as more available conventional memory. In other cases, you will find that this release of MS-DOS is somewhat more "limited" than previous versions, and I am not particularly pleased with these limitations.

Although the "generic" release (from Microsoft) of MS-DOS 5.0 has been available since about last June, I received my Zenith Data Systems MS-DOS 5.0 in early October. I have spent the last couple of months working with it and have discovered some surprising things as you will see. And since several people have also asked me about the generic MS-DOS 5.0, I will also give you some insight on that because I have installed that several times as part of my consulting business.

Installation

The first thing I noticed about installing Zenith Data Systems MS-DOS 5.0 is that the installation program is quite different from previous versions. In fact, it resembles the Windows installation in that there is a "gas gauge" which shows you how the installation is proceeding (in percent installed).

Unlike previous versions, it does not even try to FORMAT your hard drive, and it will basically install the new BIOS (IBMBIO.COM) and Kernel (IBMDOS-.COM) without deleting the old version. I found that particularly interesting because I ended up with two additional "hidden" files on my hard drive for the previous version.

The good news is that this new install program will actually allow you to choose a subdirectory name for the DOS files. But, other than that, there are some rather unusual installation limitations in the ZDS version, compared with the generic MS-DOS version available from Microsoft.

Perhaps the most interesting limitation in the ZDS MS-DOS version is that there is no "uninstall" feature. In the generic version I worked with, the installation requested a floppy disk for the specific purpose of removing the 5.0 upgrade. That is not particularly significant since one generally will not want to back out a new DOS version unless there is some major problem, but it is a difference.

I also found one other unusual feature of the installation. The SETUP program created a CONFIG.SYS file on my system that contained the following:

```
DEVICE=C:\DOS\SETVER.EXE
FILES=30
```

As you can see, there are only two lines in that file, and my observation is that most of today's software will not run with only those command lines. As most of you may know, a lot of today's software also requires at least a BUFFER= command with a minimum specification of 20 or so. Some software requires more, but 20-25 is a common requirement. If one reads the manual, it says (page 14.20 in mine) that the default is 15 if the computer has between 512 K and 640 K of RAM. Even though my computer (Z-386/16) has 640 K of conventional memory, the default of 15 buffers is NOT enough to run most of my software. My system usually has a BUFFERS= command of 25-30 (depending on what I am doing), and my FILES= command is usually set to 40. The good news is that the SETUP program "saves" your old CONFIG.SYS file as CONFIG.OLD, so you should be sure to manually update the installed one with the same number of files and buffers as the old one. That will avoid problems with software that has been installed for some time. Interestingly enough, this failure to install an appropriate number of buffers seems to be a major flaw in the installation program which does not exist in the generic version.

The generic version includes a lot of command lines in the CONFIG.SYS file, and perhaps the one that causes the most problems is:

```
SHELL=C:\DOS\COMMAND.COM/E:256/P
```

Note that this command line was installed in the generic version, not the Zenith Data Systems' version. I mention this specifically because I have received some letters about software not working correctly with the generic version, even though the same flaw does NOT exist in the ZDS version.

Here's the problem: In the generic install, COMMAND.COM is only copied to the subdirectory containing the rest of the DOS programs. It is NOT copied to the root directory of the boot drive, normally drive C on a hard drive system, which has been the standard since DOS was released for microcomputers. That all sounds great, and you will even find that the COMSPEC parameter (use the SET command to see it) correctly displays the path as defined by the SHELL= command. BUT... many programs, especially games and some utilities, NEVER check the COMSPEC parameter because the standard has always been to locate the Command Interpreter, COMMAND.COM, in the root directory (NOT a subdirectory) of the boot drive. Now along comes at least one generic version of MS-DOS which really uses the COMSPEC parameter, and guess what? Some programs cannot find COMMAND.COM (because it's not in the root directory), and all kinds of interesting error messages are displayed, depending on what program you are running.

The simple cure is to copy COMMAND.COM to the root directory and change CONFIG.SYS to show where the Command Interpreter is located. Since there is no reason to have two copies of COMMAND.COM, I suggest deleting the one in the subdirectory. Of course you will need to reboot after changing CONFIG.SYS to be sure that everything is working correctly, and it will have no effect on anything else you do. Again, let me stress that I have found this problem in at least one release of generic MS-DOS, not the ZDS version.

Of course the SETVER command is new, but I have removed that from my CONFIG.SYS since I generally have no need for it. SETVER is new in version 5.0, and its basic function is to define the DOS version number that is reported (by DOS) to a program. This can be helpful if a program is "version sensitive", but I have no need for it as of the writing of this article. You may or may not need it, but keep in mind that it is a memory-resident program that does require some conventional memory. That's the primary reason I do not use it, aside from the fact I have no need for it.

All in all, the basic installation of MS-DOS 5.0 went smoothly, but it did not set up CONFIG.SYS appropriately for my system. In fact, it really was not close enough to run all of my usual software. To correct this, I recommend that the minimum CONFIG.SYS file include at least the following three command lines:

```
SHELL=C:\COMMAND.COM /E:256/P
FILES=30
BUFFERS=25
```

If your existing CONFIG.SYS file has a higher number for the FILES= or the BUFFERS= command, be sure to update the new CONFIG.SYS file to contain those higher numbers or some of your software may not run correctly. Now let's take a look at some of the new MS-DOS 5.0 features.

New Features and Enhancements

Perhaps the most significant change in MS-DOS 5.0 is in the area of memory management. There are a variety of new device drivers that provide some of the features that were only previously available in third-party memory managers such as QEMM386. Some of these new features require the use of extended memory, which of course means that they cannot be implemented on any computer that uses an 8088 CPU. Extended memory can only be installed on a computer that has at least an 80286 CPU, which also includes the 80386 and 80486 as well.

Before we get too far along in this discussion of the new memory management features, let's take a moment to review some of the terminology. If you are interested in the details of all this, you can also refer to my October 1991 article which goes into considerable detail on the locations and contents of memory.

Conventional memory, sometimes known as base memory or DOS memory, is that memory located from 0 K to a maximum of 640 K. This is occasionally called standard memory because all DOS-based computers have at least some memory in this area. For best performance, you should have the maximum of 640 K installed in any computer you are using.

The next "piece" of memory is located between 640 K and 1024 K (1 MB). It contains six, 64K blocks of memory called Upper Memory Blocks or UMBs. This entire 384 K chunk of memory is called the Upper Memory Area or UMA.

Extended memory is located above one megabyte, and the first 64K memory block is called the High Memory Area or HMA. Don't fall into the trap of calling extended memory the HMA because that is not correct — the HMA is only the first 64K. Unlike conventional memory, extended memory can ONLY be used by programs specifically designed to access it, such as Windows 3.0.

Expanded memory can be used to "expand" conventional memory beyond the 640 K limit, and it can generally be "installed" on most computers. In all cases, the availability of expanded memory always requires that an Expanded Memory Manager (e.g., EMM.SYS or EMM386.EXE supplied with MS-DOS) be defined, usually by a DEVICE= command in the CONFIG.SYS file. For some computers, you must install an expanded memory board, and the appropriate expanded memory manager should be included by the manufacturer of that board and installed as specified by the manufacturer's instructions. Like extended memory, expanded memory can only be used by programs specifically designed to access it, such as Quattro Pro 3.0. Expanded memory has the specific advantage that it can be installed on nearly all computers, including those with an 8088 CPU. As previously stated, extended memory can only be installed on computers with at least an 80286 CPU or later. We will take a look at some of the specific changes that involve the various types of memory when we look at changes to commands and device drivers in the CONFIG.SYS file later in this article.

If you have used MS-DOS version 4.0, you may remember that you had to have the SHARE command installed in CONFIG.SYS if you had a hard drive partition larger than 32 megabytes; otherwise, you would see a message: "WARNING! SHARE should be loaded for large media." In version 5.0, you do not need to load SHARE because this version has the inherent capability to support hard drive partitions up to two GB (gigabytes) without SHARE (and the memory required to support it). Since this may be the first time you have seen a disk space capacity expressed in gigabytes for a microcomputer, a gigabyte (GB) is 1,024 megabytes. And yes, you can buy a hard drive with a gigabyte of storage if you have several thousand dollars, but those are more commonly used on a file server where lots of disk space is really required. For most users, the biggest advantage of this enhancement is that you will not need the additional memory required when SHARE is memory-resident.

In case you haven't heard about it yet, there is ANOTHER floppy disk format: 2.88 MB, which of course is twice the capacity of the existing 1.4 MB, high density (HD) floppy. Given that the 720 K format is double density and the 1.4 MB format is high density, I suppose we could call the 2.88 MB format Wazoo Density (WD), but that term is strictly from my own imagination. I think that Microsoft should really look at how floppy disks are currently labeled and sold before they include something weird like that in a manual.

Page 6.2 of my ZDS MS-DOS manual says that a 5.25-inch, double sided/quad density disk is 1.2 MB, and a 3.5-inch, double sided/quad density disk is 1.44 MB (usually shown as just 1.4 MB). While it may be technically true that these disk formats are really quad density in a literal sense, you will have a REAL difficult time in the real world finding disks that are labeled or referred to as quad density. Most of you probably know that both of those formats are already called high density and are generally labeled as HD. And I doubt that floppy disk manufacturers will change their current labeling (which would confuse EVERYONE) to accommodate something written by Microsoft, so I expect that the 2.88 MB format will be labeled WD or something like that to differentiate it from the current HD disk formats. Now let's take a look at some of the new DOS-level commands.

New Commands

Perhaps one of the best features is the new help "system" in version 5.0. I call it a system because there are two different ways you can use it. First, you can use the HELP command by itself or follow it with a specific DOS command, such as HELP FORMAT. The second way to invoke the help system is type the specific command name followed by the /? switch, such as FORMAT /?.

Finally, MS-DOS has a reasonable ASCII file editor which has been implemented as the EDIT command, although the old line editor, EDLIN, is still included in version 5. If you have ever used WordStar or any of the numerous programs that have implemented some of the CTRL keys, it may take you as long as three seconds to figure out how to use EDIT. Of course EDIT is only for ASCII files, so there are no heavy-duty word processing capabilities, but you will find there are sufficient features implemented to make basic file editing easy. For example, many of WordStar's CTRL-Q (Quick menu) commands are implemented, such as CTRL-QF (Find) and CTRL-QA (find and replace). The famous WordStar diamond is, of course, implemented, although you can always use the arrow keys too. And many of the other standard CTRL keys are clearly copied from WordStar, such as CTRL-T (delete word), CTRL-Y (delete line), CTRL-G (delete character at cursor, same as DEL), and CTRL-H (delete character before cursor, same as BACKSPACE). As a side note, I have taken the time to "fix" my Word Perfect 5.1 to implement all of the single-key CTRL-keys (with macros) so that I don't have to move at least one hand from the main keyboard. It's much easier, not to mention faster, for me to use CTRL-T to delete a word or CTRL-Y to delete a line. In any case, the EDIT command is a nice addition to this version of MS-DOS.

As I was organizing material for this article, I came across three new commands that looked suspiciously familiar from somewhere else: MIRROR, UNDELETE, and UNFORMAT. I probably would not have suspected anything if the MIRROR command had not been included, and I believe the "somewhere else" is the famous PCTOOLS because it happens that it also has those same three identical commands. It looks like Microsoft has finally recognized that at least one other software manufacturer has some good ideas and has included these three important programs in the current MS-DOS release.

The MIRROR command basically saves directory and FAT information to a file so that you can successfully UNDELETE it, assuming you use the UNDELETE command immediately after deleting the file of course. MIRROR can also be used to save hard drive partition information to a floppy disk so that the UNFORMAT command can be used to rebuild a hard drive partition. Based on my knowledge of PCTOOLS and the information contained in the MS-DOS 5.0 manual, it also looks like the FORMAT command has been modified so that it works something like the PCTOOLS "safe" FORMAT command.

Although the UNDELETE command will work without the file created by MIRROR, keep in mind that it is critical to use the UNDELETE command *IMMEDIATELY* after deleting a file or it may not be recoverable. If you do not UNDELETE a file immediately after deleting it, it may not be possible to recover it because one or more of the clusters used for the deleted file may have been overwritten if you add or change any other file before attempting the UNDELETE.

One other command in this new MS-DOS version also looks familiar: LOADHIGH. If you use QEMM386, you will find that it has the LOADHI command, which essentially performs the same function as this new DOS command. The DOS LOADHIGH command allows you to load a program (e.g., a TSR) into the Upper Memory Area, but you must have previously loaded either UMM.SYS or EMM386.EXE before you can use it. Because of that, there are some restrictions on using LOADHIGH as you will see in the discussion of the changes to the CONFIG.SYS file commands and device drivers. Specifically, you must have extended memory to load either one of those device drivers, so the DOS LOADHIGH command has a restriction that QEMM386's LOADHI does not have.

If you like BASIC, you will probably enjoy using the QBASIC command to run some of the sample programs included with this version. Two games, GORILLA and NIBBLES, and a simple money manager, MONEY, are included along with a utility (REMLINE) that removes line numbers from Microsoft Basic Programs.

Last, but certainly not least, is the DOSKEY program. I know I have seen a shareware program by the same name, but I am not absolutely certain whether this is the same program or not. Regardless, the program I remember performs the same functions as DOSKEY, and this is a really useful program if you use the DOS command line as much as I do.

DOSKEY is a memory-resident program that "remembers" commands, allows you to edit them, and also can create some sophisticated macros that can perform some sophisticated processing like batch files. Now that I've seen this program again, I wish I had found a copy before now since I use the command line a lot for testing and such.

The EXPAND Command

EXPAND is a new command in version 5.0, and its use is not obvious until you take a look at the directory for a distribution disk. As a personal preference, I now order nearly all software on 3.5-inch disks, and I ordered my Version 5.0 upgrade in that format. Figures 1, 2, and 3 are the DIR listing for the three, 3.5-inch Zenith Data Systems MS-DOS distribution disks that I received.

Again, please note that these Figures are the DIR listings for the 3.5-inch disk format. If you ordered the 5.25-inch format, obviously the contents of your distribution disks will be different.

I have included these directory listings

```
  Volume in drive B is 890-2583-0U
  Volume Serial Number is 2583-1613
  Directory of B:\

 COMMAND  COM      47845    06-03-91 5:00a
 COUNTRY  SYS      17069    06-03-91 5:00a
 HIMEM    SY_      7937     06-03-91 5:00a
 KEYB     COM      15634    06-03-91 5:00a
 KEYBOARD SYS      34697    06-03-91 5:00a
 MODE     CO_      16834    06-03-91 5:00a
 NLSFUNC  EXE      7052     06-03-91 5:00a
 SETVER   EX_      7476     06-03-91 5:00a
 PACKING  LST      3138     06-03-91 5:00a
 CONFIG   SYS      13       06-03-91 5:00a
 AUTOEXEC BAT      36       06-03-91 5:00a
 SETUP    EXE      77114    06-03-91 5:00a
 SETUP    INI      2066     06-03-91 5:00a
 CV       COM      716      06-03-91 5:00a
 FORMAT   COM      32911    06-03-91 5:00a
 EGA      CP_      19714    06-03-91 5:00a
 LCD      CP_      3470     06-03-91 5:00a
 DISPLAY  SY_      11186    06-03-91 5:00a
 APPEND   EX_      8475     06-03-91 5:00a
 DOSKEY   CO_      4730     06-03-91 5:00a
 EXPAND   EXE      14563    06-03-91 5:00a
 FDISK    EXE      58170    06-03-91 5:00a
 PRINT    EX_      11061    06-03-91 5:00a
 SHARE    EX_      9312     06-03-91 5:00a
 SYS      CO_      9661     06-03-91 5:00a
 ANSI     SY_      7185     06-03-91 5:00a
 DRIVER   SY_      4577     06-03-91 5:00a
 EGA      SY_      4107     06-03-91 5:00a
 EMM      SY_      8687     06-03-91 5:00a
 EMM386   EX_      47585    06-03-91 5:00a
 RAMDRIVE SY_      3765     06-03-91 5:00a
 SMARTDRV SY_      6295     06-03-91 5:00a
 UMM      SY_      2823     06-03-91 5:00a
 APPNOTES TXT      8672     06-03-91 5:00a
 README   TXT      13444    06-03-91 5:00a
 ASSIGN   CO_      5219     06-03-91 5:00a
 FASTOPEN EX_      8927     06-03-91 5:00a
 GRAFTABL CO_      6903     06-03-91 5:00a
 GRAPHICS CO_      12909    06-03-91 5:00a
 GRAPHICS PR_      4866     06-03-91 5:00a
 JOIN     EX_      13902    06-03-91 5:00a
 SUBST    EX_      14495    06-03-91 5:00a
 42 File(s)   37888 bytes free
```

Figure 1
MS-DOS 5.0 Distribution Disk 1

```
Volume in drive B is 890-2584-0U
Volume Serial Number is 2584-1603
Directory of B:\

4201     CP_      2605     06-03-91 5:00a
4208     CP_      368      06-03-91 5:00a
5202     CP_      261      06-03-91 5:00a
LASER    CP_      49658    06-03-91 5:00a
PRINTER  SY_      12509    06-03-91 5:00a
EXE2BIN  EX_      6886     06-03-91 5:00a
MACHINE  EX_      10885    06-03-91 5:00a
SHIP     CO_      731      06-03-91 5:00a
ATTRIB   EX_      11801    06-03-91 5:00a
BACKUP   EX_      25727    06-03-91 5:00a
CHKDSK   EX_      12705    06-03-91 5:00a
COMP     EX_      10513    06-03-91 5:00a
DEBUG    EX_      16898    06-03-91 5:00a
DISKCOMP CO_      7661     06-03-91 5:00a
DISKCOPY CO_      8396     06-03-91 5:00a
EDLIN    EX_      10441    06-03-91 5:00a
FC       EX_      13505    06-03-91 5:00a
FIND     EX_      5560     06-03-91 5:00a
LABEL    EX_      6929     06-03-91 5:00a
MEM      EX_      29726    06-03-91 5:00a
REPLACE  EX_      14417    06-03-91 5:00a
RESTORE  EX_      26085    06-03-91 5:00a
SORT     EX_      4996     06-03-91 5:00a
UNDELETE EX_      9391     06-03-91 5:00a
XCOPY    EX_      11868    06-03-91 5:00a
LOADFIX  CO_      704      06-03-91 5:00a
MORE     CO_      2319     06-03-91 5:00a
TREE     CO_      5445     06-03-91 5:00a
UNFORMAT COM      18576    06-03-91 5:00a
GORILLA  BA_      11991    06-03-91 5:00a
MONEY    BA_      14942    06-03-91 5:00a
REMLINE  BA_      5026     06-03-91 5:00a
EDIT     COM      413      06-03-91 5:00a
QBASIC   EX_      213223   06-03-91 5:00a
NIBBLES  BA_      8266     06-03-91 5:00a
HELP     EX_      8534     06-03-91 5:00a
DOSHELP  HL_      2994     06-03-91 5:00a
EDIT     HL_   17419 06-03-91 5:00a
         38 File(s)  87040 bytes free
```

Figure 2
MS-DOS 5.0 Distribution Disk 2

```
Volume in drive B is 890-2585-0U
Volume Serial Number is 2585-1603
Directory of B:\

DOSSHELL HL_      54344    06-03-91 5:00a
QBASIC   HL_      123120   06-03-91 5:00a
DOSSHELL CO_      2231     06-03-91 5:00a
DOSSHELL EX_      155854   06-03-91 5:00a
DOSSWAP  EX_      15486    06-03-91 5:00a
CGA      GR_      1768     06-03-91 5:00a
EGA      GR_      2729     06-03-91 5:00a
EGAMONO  GR_      2564     06-03-91 5:00a
HERC     GR_      1809     06-03-91 5:00a
MONO     GR_      301      06-03-91 5:00a
VGA      GR_      3593     06-03-91 5:00a
VGAMONO  GR_      3605     06-03-91 5:00a
CGA      IN_      3330     06-03-91 5:00a
EGA      IN_      3333     06-03-91 5:00a
MONO     IN_      2990     06-03-91 5:00a
CGA      VI_      6920     06-03-91 5:00a
EGA      VI_      7012     06-03-91 5:00a
HERC     VI_      6998     06-03-91 5:00a
VGA      VI_      7236     06-03-91 5:00a
MIRROR   CO_      12781    06-03-91 5:00a
MSHERC   CO_      4444     06-03-91 5:00a
RECOVER  EX_      7536     06-03-91 5:00a
WINA20   38_      3214     06-03-91 5:00a
         23 File(s)  283648 bytes free
```

Figure 3
MS-DOS 5.0 Distribution Disk 3

as part of the EXPAND command discussion because you will note that many of the files have an underscore as the last character of the file type (also called extension). Any file with an underscore is actually a "compressed" file, and it cannot be used unless it is "uncompressed" with the EXPAND command. That is automatically done when you run the SETUP program, but you can also use EXPAND as a standard DOS command in case you clobber a DOS program on a hard drive.

New CONFIG.SYS Commands and Device Drivers

A new device driver, UMM.SYS, is the upper memory manager that is used to control the Upper Memory Area for computers that have an 80286 CPU or later. This device driver will give you access to approximately an additional 96 KB of the UMA: from C800h to CFFFh (32 KB) and from E000h to EFFFh (64KB). That's the good news. The bad news is that you must have extended memory installed in your computer because HIMEM.SYS must also be installed BEFORE UMM.SYS. That also explains why UMM.SYS will only work on an 80286 or later CPU, although the manual specifically mentions that UMM.SYS provides UMB support according to the eXtended Memory Specification (XMS) 2.0. As you can see, this device driver is not much help if you have a SupersPort 286 laptop that only has the standard 1 MB of memory installed.

Another device driver, EMM386.EXE (not SYS), can be used on 80386 or later computers to "convert" extended memory to expanded memory and provide support for the UMA (like UMM.SYS). Like UMM.SYS, EMM386.EXE also requires that extended memory be installed, and HIMEM.SYS must also be installed as well. This device driver is not much help on a SupersPort 386SX laptop that only has the standard 1 MB of memory installed.

The DEVICEHIGH= command is similar to the basic DEVICE= command, except that DEVICEHIGH= is used to load a device driver into the Upper Memory Area. Note that this can "recover" some conventional memory that need not be used for one or more device drivers, assuming that there is enough free space in the UMA to load what you need of course. Unfortunately, the DEVICEHIGH= command requires that either UMM.SYS or EMM386.EXE be installed first, so of course you cannot use this new command if you only have the standard 1 MB of memory in your computer, such as a SupersPort laptop. Remember that the DEVICE= command will only load a device driver into conventional memory.

The DOS= command is a new command that generally allows you to load part of MS-DOS in extended memory (specifically the HMA) to save conventional memory space. Of course you must have an 80286 or later system, and extended memory must be installed so that HIMEM.SYS can be loaded first.

For those of you unfamiliar with HIMEM.SYS, it is used to generally manage extended memory, including the HMA in accordance with XMS 2.0. This primarily prevents conflicts arising from the use of programs which access extended memory, including extended to expanded memory "converters" such as EMM386.EXE.

RAMDRIVE.SYS appears to be little more than an updated VDISK.SYS with a new name. It is a device driver that uses memory to emulate a disk drive, and since nearly everyone has probably experimented with that at some time or other, there is no additional explanation required.

Although ZCACHE.SYS seems to be missing in action, it has apparently been replaced by a new disk caching program, SMARTDRV.SYS. If you have worked with Windows 3.0, you are probably familiar with this device driver which requires the use of either expanded or extended memory. Note that SMARTDRV.SYS cannot be used with conventional memory, which is a feature that was supported by ZCACHE.SYS.

The SETVER.EXE device driver is also new in this version, and it is used to define the DOS version number that is reported to a program. Some programs have been written to check the DOS version number and will run only if the version number is reported as expected. If a program is not compatible with this version of MS-DOS, and it may not be for a lot of reasons (e.g., copy protected programs), then the SETVER command will not change the compatibility problem. I mentioned earlier that the SETVER command was installed in the CONFIG.SYS file during my installation of version 5, but I deleted that line because there is absolutely no reason to use any memory on my system for that. On my system, CHKDSK noted that 591,248 bytes of memory were free with SETVER installed,

```
WIN200.BIN        3.40
WIN100.BIN        3.40
WINWORD.EXE       4.10
EXCEL.EXE         4.10
HITACHI.SYS       4.00
MSCDEX.EXE        4.00
REDIR4.EXE        4.00
NET.EXE           4.00
NET.COM           3.30
NETWKSTA.EXE      4.00
DXMA0MOD.SYS      3.30
BAN.EXE           4.00
BAN.COM           4.00
MSREDIR.EXE       4.00
METRO.EXE         3.31
IBMCACHE.SYS      3.40
REDIR40.EXE       4.00
DD.EXE            4.01
DD.BIN            4.01
LL3.EXE           4.01
REDIR.EXE         4.00
SYQ55.SYS         4.00
SSTDRIVE.SYS      4.00
ZDRV.SYS          4.01
ZFMT.SYS          4.01
TOPSRDR.EXE       4.00
```

Figure 4
SETVER Version Table Listing

and 591,664 bytes were free without SETVER installed, which means that 416 bytes of memory were required. Even though SETVER only required a very small amount of memory, I did not think it was a worthwhile use of even that. Let's take a moment to look at that in more detail.

If you enter the SETVER command by itself, you will see a list of programs and the appropriate DOS version that should be reported. According to the manual, Microsoft has apparently verified the programs in the existing SETVER table are version sensitive, so the obvious thing to do is see what the programs are. For discussion purposes, I have included the list of programs displayed by SETVER as Figure 4.

After examining the list shown in Figure 4, I have never used any of the programs listed, so it seemed silly to use any memory for that kind of thing. I'll leave it to you to determine whether you use any of these programs on your system, but if you don't, I recommend deleting the SETVER command from your CONFIG.SYS file because there is no need to waste memory on something you don't need.

That covers the new commands and device drivers for MS-DOS 5.0, but I should mention that EMM.SYS is still provided with this version. That may be important in your system because EMM.SYS can still be used to support expanded memory in systems that only have 1 MB of standard memory installed, such as the SupersPort 386SX. On my SupersPort 386SX laptop, EMM.SYS supports 224 K (229,376 bytes) of expanded memory within the standard 1 MB of memory I have.

Enhanced Commands

Whether or not a command has been enhanced (i.e., "improved") depends on your perspective. If you use a specific command a lot, and something has been added to make it better or easier to use, then it is appropriate to call it an enhanced command. For the most part, it looks like most of the MS-DOS 5.0 command enhancements have really been improvements, but there are some interesting changes I noted during my testing.

One of the most often used commands is FORMAT, and it has apparently been enhanced to reflect the addition of the UNFORMAT command mentioned earlier. Specifically, FORMAT begins by displaying a message that it is "Checking existing disk format", then it displays another message to the effect that it is "Saving UNFORMAT information."

The enhanced FORMAT command has some new and different switches, which are clearly due, in part, to the UNFORMAT capability. I noticed that the /N (Number of sectors) and /T (Tracks) switches are still supported in this version, but the /F (Floppy format) switch is easier to use because you don't have to remember the number of tracks and sectors when you are trying to FORMAT a 720 K, 3.5-inch floppy disk in a 1.44 MB drive. To do that, you simply enter: FORMAT A:/F:1.44 (or various other equivalents). Although the /F switch was available in version 4.0, the latest version also includes the new 2.88 MB format mentioned earlier, assuming you can find a floppy drive and some disks for that capacity.

One of my biggest gripes about "enhanced" programs occurs when a command or switch in the previous version does something completely different in a new version. Such is the case with the /Q (Query) switch, which was used to suppress prompts and statistics in version 4.0. Even the old manual mentions that this switch may be useful in a batch file. In 5.0 however, the /Q (Quick format) switch deletes the FAT and root directory of a previously formatted disk, which essentially just speeds up the format because the disk is not re-tested for bad sectors. I mention that in case you have created some batch files for use with the FORMAT command, and there is no substitute for version 4.0's query function.

The /U (Unconditional format) is used when you know you will not want to recover any data from a formatted disk. In other words, you cannot use the UNFORMAT command after you specify the /U switch because it destroys all data on a disk just like the old FORMAT command did. The other switches, /V (Volume label), /B (to reserve space for the BIOS and DOS Kernel), /S (System), /1 (Single-sided format), /4 (360 K format in a 1.2 MB drive), and /8 (8-sector format) are still supported in version 5.0.

The ATTRIB command has been enhanced to allow you to change both the System and Hidden file attributes, in addition to Archive and Read-only.

The DEL (or ERASE) command also has a new switch, /P for Prompt. Now you can enter a command like *DEL *.DOC/P*, and you will see a prompt for EACH file in the form of: filename.typ Delete (Y/N)? Looks like a DOS programmer finally saw the famous SWEEP program that I originally used on my H-89 with the CP/M operating system.

Even the DIR command has been enhanced to finally add the capability to see all files, regardless of attributes. For some reason, the DIR command could not previously display any files that had either the Hidden or System file attribute (such as MSDOS.SYS, IO.SYS, IBMBIO.COM or IBMDOS.COM), so you could not use DIR to see exactly what files were occupying how much disk space.

The MEM command has a new switch, /C (Classify), which displays the status of programs loaded into conventional memory and the upper memory area.

The MODE command has a new feature which allows you to change the keyboard's typematic rate, which can help make your keyboard more responsive. And based on a quick look through the manual, it looks like you might need to use the MODE command for printer configuration because the CONFIGUR command is not provided with this new MS-DOS version.

Where Is?

As I was going through the Zenith Data Systems MS-DOS version 5.0 manual, I noticed that one specific command was missing: DSKSCAN. If you have version 4.0, you probably know that DSKSCAN scans a hard drive partition for bad sectors and records bad sectors in the File Allocation Table. I was disappointed to find this program was missing because I use it regularly, but then, upon doing some more checking, it turns out that there are more of the most common programs "missing."

CONFIGUR is perhaps the oldest Zenith Data Systems' utility, and it dates back to the old CP/M days. It's gone, although part of its capabilities can be entered with the MODE command. I dislike the MODE command because it is memory-resident, but such is life.

DSKSETUP is also missing in action. The hard drive utilities PART and PREP are nowhere to be found, but PART has apparently been replaced by FDISK in version 5.0. And for those of you interested in such things, both ZCOM and ZSPOOL have disappeared. And ZCACHE.SYS has apparently been replaced by SMARTDRV.SYS.

What About MS-DOS 5.0?

Although there are a lot of new features in MS-DOS 5.0 that are nice, I cannot recommend it as an upgrade that every computer user should have for several reasons. Perhaps the most important is that you will not be able to take advantage of the new memory management features unless your computer has extended memory installed. As I mentioned earlier, my SupersPort 386SX only has the standard 1 MB of memory, so it really is not worthwhile for me to buy DOS 5.0 for it. And even though DOS 5.0 has some new programs like UNDELETE, I don't need those either because I can use something like PCTOOLS or Norton Utilities which have had that for years. In short, MS-DOS 5.0 really has nothing to offer for me on either of my laptops.

The fact that these new memory management features require extended memory also makes it impossible for me to recommend MS-DOS 5.0 for anyone who does not have a computer with at least an 80286 CPU or later. If you are using something like a Z-151 or an eaZy PC, then I do not think that this DOS version will add much to your system.

If you do have a computer that has an 80386 or 80486 CPU, but has only the standard 1 MB of memory, then I think you will find QEMM386 is a much better value because it is more flexible. In the short term, you can use QEMM386's LOADHI program to load TSRs in the UMA without any additional memory required. And if you decide to add additional memory later on, then QEMM386 can support that, too.

Many of the other new MS-DOS 5.0 features have been available in other software for years, such as PCTOOLS and Norton. I expect that many of you probably have at least one general utility program, and even if you decide to upgrade to this new DOS version, keep in mind that you will also have to upgrade your utility program for compatibility reasons. For example, PCTOOLS version 6 is compatible with MS-DOS version 4.0, but it is NOT compatible with MS-DOS 5.0. The point is that it may be more expensive to upgrade to MS-DOS 5.0 because you will also have to upgrade your utility software. This requirement has existed for the last couple of versions of MS-DOS, so it is not new to version 5.0, but I think it is something you should consider.

I should also point out that you will also need to upgrade to version 6 of QEMM386 if you upgrade to MS-DOS 5.0. If you are currently using another memory management utility with an older DOS version, you will probably need to upgrade that, and I suggest you check with the manufacturer for version 5.0 compatibility and upgrade costs.

On the other hand, you will probably want to upgrade to 5.0 if you are also running Windows 3.0. The memory management features of MS-DOS 5.0 may help speed up Windows, which needs all the help it can get. Whether or not this upgrade will help you much really depends on what hardware you have. The Zenith Data Systems MS-DOS 5.0 (or the generic version) upgrade is conditionally recommended if you have extended memory and need some or all of the new features. Also, keep in mind that you will also need to upgrade any disk utility and memory management software if you upgrade to this new MS-DOS version.

Powering Down

I have not attempted to cover every single new and enhanced feature of the Zenith Data Systems MS-DOS version 5.0, but I have mentioned those which I believe most computer users will find most helpful. I have noted all of the new commands to give you an idea of what you might expect to find.

For help in solving specific computer problems, be sure to include the exact model number of your system (from the back of the unit or series from the Owner's Manual), the ROM version you are using (use CTRL-ALT-INS to find it, except for the eaZy PC), the DOS version you are using (including both version and BIOS numbers from the VER command), and a list of ALL hardware add-ons (including brand and model number) installed in your computer. The list of hardware add-ons should specifically include memory capacity (either added to an existing board or on any add-on board), all other internal add-on boards (e.g., modem, bus mouse or video card), the brand and model of the CRT monitor you have, and the brand and model of the printer with the type of interface (i.e., serial or parallel) you are using. Also be sure to include a listing of the contents of the AUTOEXEC.BAT and CONFIG.SYS files unless you have thoroughly checked them out for potential problems (e.g. TSR conflicts). If the problem involves any application software, be sure to include the name and version number of the program you are running when the problem appears.

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

Products Discussed

Powering Up (885-4604)	$12.00

Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3463 (ZUG Software only)

Manifest	$60.00
QRAM	80.00
QEMM386	100.00
DESQview	130.00
DESQview386	220.00

Quarterdeck Office Systems
150 Pico Boulevard
Santa Monica, CA 90405
(213) 392-9701

"I CAN REMEMBER WHEN WE USED TO GIVE THEM APPLES!"

Upgrade of Hard Drives on the SupersPORT 286

LTC Charles F. Santose
US Army Reserve
1190 Hunters' Trail
Broadview Heights, OH 44147

In January 1991, my US Army Reserve unit was activated for deployment during Operation Desert Shield. We'd be unable to take our Z-248s, so I transferred all the important software onto the 20M hard drive on my personal SupersPORT 286e computer.

The SupersPORT 286e did yeoman duty for the next few weeks as we prepared for special duty in the Gulf. Everything ran well until the hard drive died! Just died, no whimpering, just dead. By this time the air war had started and we knew ground forces would be close behind. We needed a new hard drive and would have to get it from non-Army sources.

I had already looked at upgrading the small 20M drive in the computer to something larger and knew the Connor 40M CP-3044 drive would work, as it was an option on new SupersPORTs. Since the Connor CP-30104 drive has worked in the ZDS TurbosPORT series, I surmised it would also work in the SupersPORT 286. All ZDS sources checked said "No way! It takes a special controller, and you can only get that from us." Taking that as a challenge I ordered a CP-30104 from Hard Drives International and had it shipped Federal Express overnight. (OK, so I cheated and ordered both a CP-30104 and a CP-3000 in case the 122M wouldn't work! HDI took the unopened CP-3000 back without even a question.)

To make a long story short, the 122M Connor CP-30104 just slid into the SupersPORT 286e and has been running happily ever since. It isn't quite that simple on the other SupersPORT 286s. The details on how to do this to any other SupersPORT 286s follow. Since the Heathkit H-2860 and H-2862 are kit versions of the ZDS SupersPORT 286 the conversion works for them as well.

The key to putting either the 40M CP-3044/3000 or 122M CP-30104 drive in the SupersPORT 286 is a few ZDS parts. If your SS-286 is the VGA (-e) model, or one of the later CGA (non-e) models you only need the drive. If you have an earlier model, with the older Connor CP-3xx series IDE drive (3 digit drive model number on the drive) you need a couple of ZDS parts.

If you're installing the 122M drive you need a ROM set that contains Type 61. The 40M drive is Type 44 and it's in all SS-286 ROMs.

To install a hard drive in the Heathkit H-286x series that doesn't currently have a hard drive, you only need to add a hard drive in the left drive bay pretending you originally had the older series Connor (you need all the parts if you threw away the Winchester bracket that was standard with the H-2862).

Note: It has been reported that this conversion or upgrade may not work on very early SS 286 CGA motherboards (revision -02 or less). Very early Connor IDE drives were not true IDE specification and ZDS accounted for this on some desktops with a jumper. ZDS accounted for this on the SupersPORT 286 in the motherboard circuitry which changed at -03. A jumper is not present in the SupersPORT 286. If you have a very old SupersPORT 286 you should insure you can get a refund on any drive you purchase that won't work!

Connor Drives

Drives known to install in the SupersPORT 286 series are:

Connor		ROM
Model No.	**Capacity**	**Drive Type**
CP-344	42M (factory installed old models)	44
CP-3044	42M (factory installed new models)	44
CP-3000	42M (replacement for 3044)	44
CP-30104	122M	61

Other brands (Quantum, Maxtor) have been tried without success. Stick with a Connor drive.

Zenith Data Systems Parts Required

Hardware

For ALL installations:

969-1917 Cable, Hard Drive, Data, SS286e (40 conductor)(Optional, but highly recommended)

For "older" series SS 286, and H-2860 (CGA models):

969-1814 Bracket, Hard Drive, SS286e, or
969-181-D (old model number, same bracket), or
232-183-1 (even older model number!)
969-2345 Cable, 4C, Power, SS286 (4-pin to 3-pin)

For H-2862 (VGA model) without a hard drive:

969-1812 Cable,. Power, SS286e (3-pin to 3-pin)

Firmware

For Connor CP-30104 drive in SS-286 and H-2860 (CGA models):

444-671-05 (As of Sep 90. The -05 is the revision number
444-672-05 and the version you get may be higher.)

Note: There are reported conflicts with the -05 ROM (version 3.4E), and EXTended memory. The conflict is a refusal to soft boot from the keyboard. The computer functions normally expect for this problem.

Source for Hard Drives and Parts

There are several sources for hard drives. Check PC-Magazine or Computer Shopper for prices. I've purchased a number from Hard Drives International (800-927-2907) which ships FedEx so you'll have the drive in only a couple of days. You only need the "bare drive" and the 40-conductor cable. If you have to order a ZDS mounting bracket, or ROMs, go with the ZDS data cable listed above as it simplifies installation. Otherwise you can rework the 40-conductor data cable that comes with the drive.

Zenith Data Systems parts can be obtained at a ZDS dealer or service center. One is Qualitech in Delaware (302-427-2600; ask for parts).

Installation

Once you've assembled all the parts it's time to put them inside the computer. You'll need a special tool: a Phillip's screw driver.

If your current hard drive is still working you can simplify reinstallation of the software by performing a BACKUP. Don't try to use FastBack. Earlier versions don't work on the SS 286 due to DMA conflicts, and I would not trust a newer version unless it was tested extensively. FastBack assures me that it's made a verified Backup. but Restores always fail.

If your current hard drive is dead you'll have to rely on your last backup.....

a. Gather all parts and special tools. Set aside about 90 minutes for the project. Remove small hands and cats from the area.
b. Close the computer cover and unplug the battery case and power connectors. Tape the cover closed so it doesn't pop open unexpectedly and damage the case (don't tape over the lower half of the bottom case as you're going to remove that part).
c. Invert the unit on a towel to keep from scratching the case and remove the seven (7) screws holding the lower case together. Keeping the case together, rotate the unit upright. Lift slightly and rotate the top half of the case and the screen to the rear and lay it flat. There is enough slack in the video cables to allow you to leave them connected.

Examine the video connector where it plugs into the main unit. If it's not held in place with a bracket or clip, get one from your dealer.

d. Remove keyboard (2 screws). Just move it slightly out of the way, don't disconnect the cable unless you're also installing new ROMs.

To remove the keyboard cable lift up both ends of the bracket on the motherboard until it pops up enough to free the cable. When you replace it just stick the cable in the slot and snap the connector back down.

e. Remove the hard drive (3 screws). If the hard drive is connected to another circuit card you've got the CP-3xx series with the separate controller. Remove the controller. Disconnect the drive or controller from the motherboard 40-pin connector. Disconnect the power cable from the drive noting the position of the wires.
f. If you're installing new ROMs in the non-VGA versions, and there is only one access hatch on the bottom of the computer for the 80287 chip, you'll also need to remove the floppy drive (3 screws), the power supply, and motherboard. The ROMS are on the bottom of the motherboard. (If you've got two hatches don't disassemble further, just remove/replace the ROMs through the bottom hatch!)

ROMs on the SupersPORT 286e (VGA version) are on the TOP of the motherboard, under the keyboard. All '286e ROMs I've seen already include Type 61.

Remove the old ROMS and replace. Insure all pins go into the sockets. Reassemble the motherboard, floppy drive, and power supply.

g. Mount the data cable to the drive. The ZDS data cable is optional, but highly recommended. You can use the cable that comes with your new drive with a little care and modification. My cable came with two connectors on the disk drive end so I cut one off. Also separate the flat cable into pairs to increase it's flexibility. Do this carefully so you don't disturb any connections or break any wires.
h. Install the new drive on the bracket with four screws (first removing the old drive if necessary!).
i. Install the (new) power connector to the power supply, if necessary, and drive. The brown wire goes toward the center of the drive.
j. Connect the 40-pin cable from the drive to the motherboard.
k. Find a piece of cardboard and place it under the drive approximately over the hard drive bay. We want to pre-test the installation before we reassemble the computer and the cardboard will insulate the drive from the computer innards so we don't get exciting sparks and blue smoke.
l. With the drive safely isolated from the computer, plug the power cable in the rear panel and switch on the computer. The floppy should wind up and buzz, and you should hear the drive motor on the hard drive running. If you don't hear both drives running POWER OFF and check your work! The screen will probably display a error to the effect that you can't boot this drive. Turn everything back off and unplug the power cord.

Reassembly

Reassembly is the reverse of disassembly:

a. Install the drive. Neatly fold the 40-conductor drive cable in such a manner that it exits the bottom of the drive and then comes between the two bracket feet. Place the drive/bracket assembly into the computer and arrange the cable so it isn't pinched or under the bracket. There's plenty of room under the keyboard for any excess. Keep the cable out of the modem compartment.

Secure the drive bracket with three (3) screws. The screw in the back left corner has a black ground wire connected from the keyboard. Run this wire where it won't be pinched. The screw hole nearest the modem compartment is left open as a case-screw will go through here. Neatly run the power cable between the floppy and hard drive. If the drive and bracket won't fit you'll have to move the drive on the bracket (the mounting holes are slotted). It's tight, but it all fits.

b. Replace the keyboard, making sure all the drive cable is folded neatly underneath. If you removed the keyboard cable from the motherboard, be sure to snap the bracket back down to secure the cable. Be sure the lugs at the top of the keyboard go into the slots and secure with the two small screws.
c. Replace the top half of the case (and screen). Don't forget to put the hard drive hole blanking plate in place. The case takes a little jiggling sometimes. Make sure no wires are sticking out. Replace the seven screws in the bottom of the case.

Testing and Software Installation

Power up the computer, listening for the drive motors and watching the LEDs. If your computer is set to boot from the hard drive you'll either get the message "Not a bootable partition" or nothing at all. Either is normal, the first is more common.

a. Hit Ctrl-Alt-Ins and enter the monitor ROM. Select the proper Hard Disk Drive Type. Save your SETUP.
b. Reboot the computer from the floppy. I recommend you use your MS-DOS or DR-DOS Disk #1. This will run the setup program (various names depending on your version). Go to the hard drive part of the menu and see if the format option is there. If the format option is there you can proceed if you want the drive to be all one partition.

If you've installed the CP-30104 I recommend you PARTition it into two 60M partitions. If you use MS-DOS 3.3+ or 4.01 you can make both partitions bootable. If

Continued on Page 43

Lotus 1-2-3 Release 3.1

Part 1

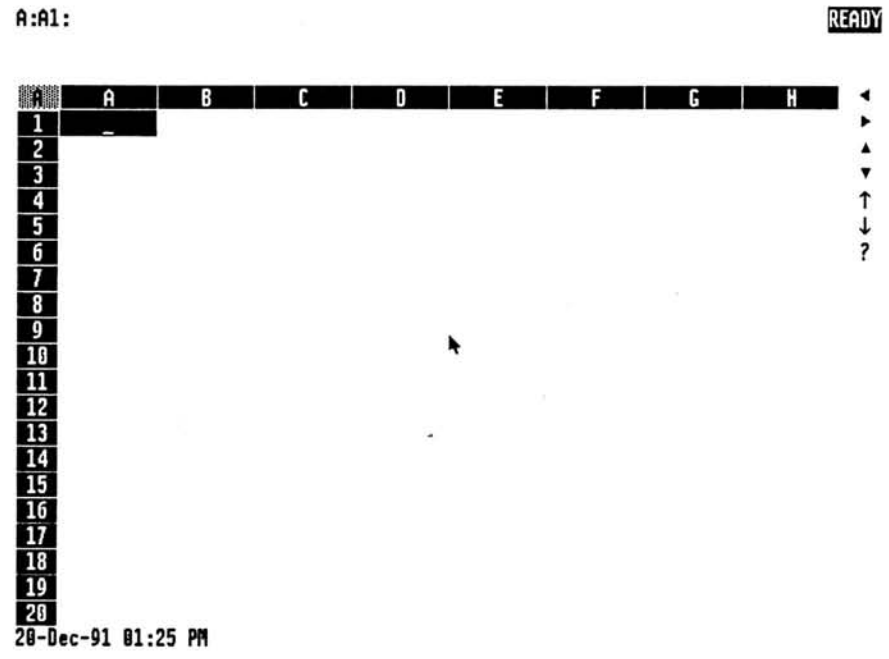
Craig S. Stevenson
Stevenson Technical Services, Inc.
108 2nd Avenue West
Bertha, MN 56437-0297

Spreadsheets From Primitive to Powerful

I purchased my first "real" computer system the "right way." I found the software that met my needs, and bought the hardware to run it. I knew that Lotus 1-2-3 would do what I needed, and a Zenith Data Systems Z-100 was the computer I chose for my hardware platform. Previously, I had run the gamut of primitive spreadsheets including VisiCalc, AceCalc, and even SuperCalc (CP/M) for the Apple and Franklin computers. Lotus release 1.0 and release 1A had the power I needed, and met my needs for several years. As my requirements increased, I felt the need to upgrade. Unfortunately, Zenith decided to drop the Z-100 line, forcing me to move into the MS-DOS area. I purchased Lotus 1-2-3 release 2.01, upgraded to release 2.2, and now use release 3.1.

Be forewarned that Lotus 1-2-3 takes a lot of horsepower to run effectively. Lotus recommends at least a 286 with 2 megabytes of memory. You also need at least 5 megabytes of free space on your hard disk. If you just create small, simple spreadsheets, you might get by with the minimum configuration, but when you start to use the macros, database, and file linking capabilities, you will want a fast machine with a lot of memory. One of my worksheets contains the names, addresses, and other pertinent information on over 2,250 people and consumes 344,902 bytes of storage! Another worksheet that I use in my business for depreciation purposes is spread over 11 individual files that are linked together into a summary report and occupy 291,767 bytes on my hard disk for the .WK3 files alone! I also use macros to automatically set up the graphs and do the printing of my spreadsheets. One worksheet that I set up uses an extensive macro to link three separate spreadsheets together, create a sorted master report of the information, and print the report with WYSIWYG enhancements. Then three separate graphs are generated based on the information in the master report, and each graph is printed, all under macro cọntrol. The entire process takes about 20 minutes to complete, even when using a 25MHz 386 computer and a laser printer. The most important point is that 1-2-3 is powerful if given the right hardware.

Although there are alternatives to Lotus these days, Lotus 1-2-3 remains the de facto standard by which all other spreadsheets are compared. The examples that follow are specific to Lotus 1-2-3 release 3.1, but the principles may be applied to other spreadsheets as well.



```
A:A1:                                              READY

20-Dec-91 01:25 PM
```

Figure 1

Lotus 1-2-3 release 3.1 main features:

- Spreadsheet size is 256 columns by 8192 rows.
- Spreadsheet is 3-dimensional with up to 256 sheets.
- WYSIWYG (What You See Is What You Get) interface.
- Expanded macro capabilities.
- Formula to file linking capability.

In this series of articles, I will guide you through some basic spreadsheet concepts, as well as some of the more complex features of Lotus 1-2-3 such as macros, file linking, graphing, and the WYSIWYG interface.

Getting Started

If you have a menu system (doesn't everybody?), load 1-2-3 as you normally would. If you don't have a menu system, change to the 1-2-3 sub-directory, (mine is 123R3), and type LOTUS. This brings you into the Lotus Access System. As is consistent throughout 1-2-3, you may type the first character of a menu option to select it, or you may use the arrow keys to highlight the option you wish to choose and press [ENTER]. If you are a touch-typist, it is much faster to just press the first character of your choice. At this point either press [ENTER] when 123 is highlighted or press 1 to select 123. Please note that it is not necessary to use the Lotus Access System at all. You could simply type 123 at the DOS prompt to take you directly into 1-2-3. I generally enter through the access system because there are times when I want to modify my configuration or access the translate utility. You should now be looking at the screen shown in Figure 1.

Depending on how you installed 1-2-3, your screen might look a bit different. For this article, 1-2-3 was set up in 80x25 VGA Monochrome mode, so that the screen captures are readable. Normally, I use 80x34 VGA Color mode. 1-2-3 also has an 80x50 VGA Color mode, but I find that the text is a bit too small for my tastes. I have also set my system to load the WYSIWYG add-in automatically. I have found 1-2-3 to be much easier to use with WYSIWYG loaded, and some of the features I will be discussing require that you be in WYSIWYG mode. If you don't have WYSIWYG loaded, load it now. If you don't know how to load WYSIWYG, don't worry. I'll tell you how later in this article.

A Look Around

The top three lines of the screen are called the control panel. The top left corner of the control panel contains the address of the cell pointer, and is used to indicate its current location on the worksheet. The mode indicator is located at the top right corner of the control panel, and is used to describe the current mode of operation. This indicator changes to show that you are entering a value or label, accessing the menu, editing a cell, or if you have made an error. The mode indicator now displays READY, showing you that 1-2-3 is ready for you to enter data or select a command from the menu. The second line of the control panel contains the 1-2-3 command menu. The third line of the control panel contains either submenus or command descriptions, depending on the particular command that is highlighted. The file and clock indicator is located in the lower left corner of the screen. This indicator shows the file name of the current worksheet if you have saved it to disk. If you have not saved your worksheet, the indicator displays the current date and time.

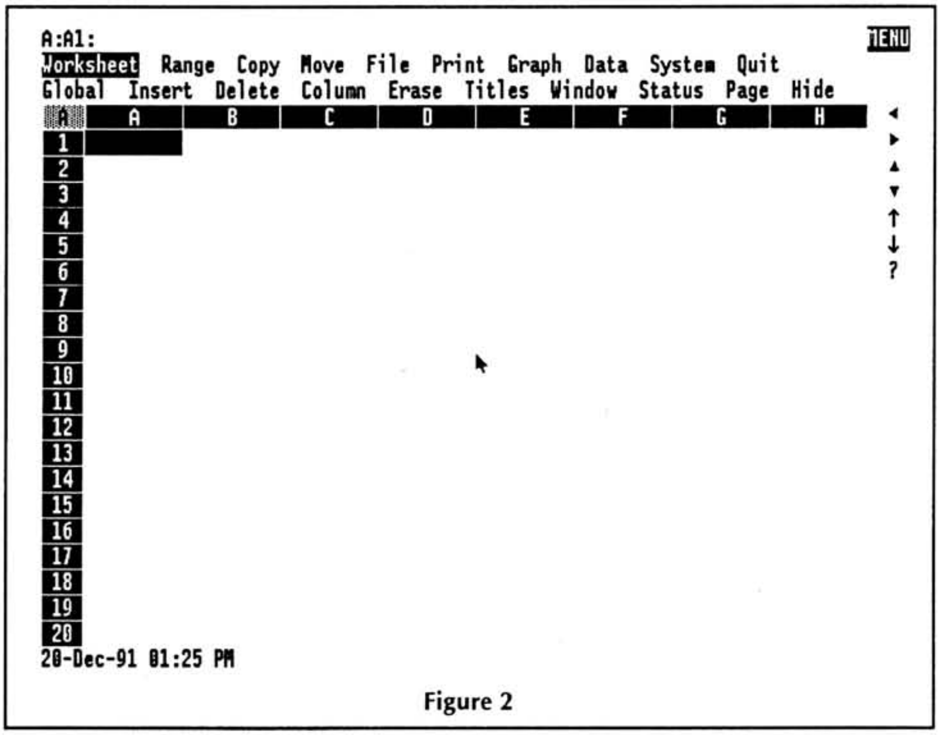

A:A1:
MENU
Worksheet Range Copy Move File Print Graph Data System Quit
Global Insert Delete Column Erase Titles Window Status Page Hide
A
A
B
C
D
E
F
G
H
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
?
20-Dec-91 01:25 PM

Figure 2

The dark horizontal bar near the top of the screen is divided into columns labeled A through H. The dark vertical bar on the left side of the screen is divided into rows labeled 1 through 20. The cell pointer is positioned at the intersection of column A and Row 1. Every intersection point is called a cell, and is labeled with the Column (letter) preceding the Row (number). On this screen, the cell pointer is located in cell A1. The cell that contains the cell pointer is called the current cell. When you move the cell pointer, the address will be updated in the control panel. If any data is in the current cell, it will be displayed following the cell pointer address in the control panel. The small shaded box at the intersection of the dark bars is the worksheet letter. You are currently on sheet A. As I stated earlier, 1-2-3 has 3-dimensional capability. The top sheet is sheet A. If you insert additional sheets underneath the top sheet, they become sheet B, sheet C, and so on. If you look at the cell pointer address in the top left corner of the control panel, you will notice that it reads A:A1. This means that the cell pointer is located on sheet A (A:), and at cell A1 (A1:). You use the arrow keys to move the cell pointer around the worksheet. Don't be afraid to move off the screen. 1-2-3 will scroll the screen to follow your cell pointer. The [PgUp] and [PgDn] keys also may be used to move up and down the length of the screen. To move right and left the width of the screen use [CTRL-Right Arrow] and [CTRL-Left Arrow]. If you want to return to cell A1 at any time, press the [HOME] key.

1-2-3 has status indicators at the bottom of the screen that indicate whether your [CAPS lock], [NUM lock], or [SCROLL Lock] keys are currently activated. There are additional status indicators that come on when you are running low on memory, have something printing, need to recalculate your worksheet, or have a circular reference in your formulas.

The 1-2-3 Menu System

The entire 1-2-3 menu system may be accessed from the keyboard. As a touch-typist, this is one reason that I stick to 1-2-3. There is really no need to move your hands from the keyboard. Mice have their place, but you don't enhance productivity by having a good typist remove their hands from the keyboard to drag a mouse around the desktop. The 1-2-3 menu access key is the slash (/) key, which is the key under the question mark, on the right side of your keyboard. If you press the slash key now, you should see the top level of the 1-2-3 menu system. Your screen should now look like Figure 2.

1-2-3 menus are arranged in a structure similar to an upside-down tree. You start at the top level or trunk, and proceed down through the branches until you get to the menu selection that you want. This may appear cumbersome at first, but in practice it is extremely fast since your hands never leave the keyboard. If you are not sure where your selection is going to lead you, move the highlight bar over your selection with the arrow keys. On the line below the menu choices will appear the choices for the next menu down the tree. If the tree does not extend any farther down, you will be shown a description of what this selection will do. Note in Figure 2 that when Worksheet is highlighted, the line below reads, "Global Insert Delete Column Erase Titles Window Status Page Hide." These are commands that are accessed through the Worksheet selection from the top menu. In order to "back up" through the menu tree, use the [ESC] key. [ESC] also may be used to "Escape" from trouble. If you make an error and don't know how to clear it, keep pressing [ESC] until you are returned to the spreadsheet. Some areas of the menu tree also allow you to Quit back to the spreadsheet by highlighting Quit and pressing [ENTER] or simply by pressing the Q key. If you are deep into the 1-2-3 menu tree and want to return quickly to the spreadsheet, press [CTRL-BREAK]. If you want to quit 1-2-3 entirely, press the slash key and then the Q key. 1-2-3 will ask you to confirm that you want to quit. Press the Y key and you will be back to the Lotus Access System. Press E to exit and you should be back to your menu or at a DOS prompt.

For the rest of this article I will show 1-2-3 commands in their long format. It is up to you to remember to use the shortcuts. Words and formulas enclosed in double quotes ("Example") should be typed exactly as they appear, but the double quotes should not be entered.

In the following section, I will guide you through the process of setting up a simple worksheet in which you will use many of the features of Lotus 1-2-3.

A Sample Spreadsheet

If you started 1-2-3 from scratch, you should have a blank worksheet now. If you accidently entered some characters into the worksheet, clear it by entering the command /Worksheet Erase Yes. Remember that the shortcut is /WEY. You do not need to press [ENTER] when using the shortcut keys to access the 1-2-3 menu system. Move the cell pointer to location A1. Type "XYZ Company" and press [ENTER]. Move the cell pointer to location A2, type "'1991 Monthly Sales Summary Report", and press [ENTER]. Remember to put a single quote (') before the first 1 in 1991. This tells 1-2-3 to treat the number 1991 as a label, instead of a value. If you begin a cell with a letter, 1-2-3 treats the cell as a label. If you begin a cell with a number or mathematical symbol, 1-2-3 treats the cell as a value. Notice that the text automatically flows into the next cell. Those of you that have used older spreadsheets will appreciate that you no longer have to manually split words over multiple cells. Now move the cell pointer to cell A5 and type "^Month" and press [ENTER]. Notice the caret (^) right before Month. This is called a label prefix. 1-2-3 recognizes several label prefixes and aligns the text differently for each one. If you begin your text with a single quote ('), 1-2-3 left-justifies the text you enter. This is the default, so if you enter text without a label prefix, 1-2-3 will left-justify it. If you use a double quote (") as your label prefix, 1-2-3 will right-justify your text. If you use a caret (^) as your label prefix, 1-2-3 will center your text within the cell. Even if you change the column width, the text will remain centered within the cell. Move the cell pointer to cell A6. If you want to put some sort of divider between the header and the data you could try typing in several hyphens, but this won't work, since 1-2-3 interprets the first hyphen as a negative number indicator. To compensate for this, you could begin with a single quote (') to indicate that you want this to be a left- aligned label, but there is a more elegant way to do it. Type "\-" and press [ENTER]. Note that this is a backslash, not a regular slash. The backslash (\) tells 1-2-3 that this cell is going to be a repeating label. Whatever character or characters follow the backslash will be repeated to fill the entire cell. This saves time when you are constantly changing the column widths of your worksheet.

Complete the shell of the worksheet by entering the proper labels in the following cells:

A7	–	"Jan"
A8	–	"Feb"
A9	–	"Mar"
A10	–	"Apr"
A11	–	"May"
A12	–	"Jun"
A13	–	"Jul"
A14	–	"Aug"
A15	–	"Sep"
A16	–	"Oct"
A17	–	"Nov"
A18	–	"Dec"
B4	–	"^Chicago"
D4	–	"^Los Angeles"
F4	–	"^New York"
B5	–	"^Sales"
C5	–	"^Profit"

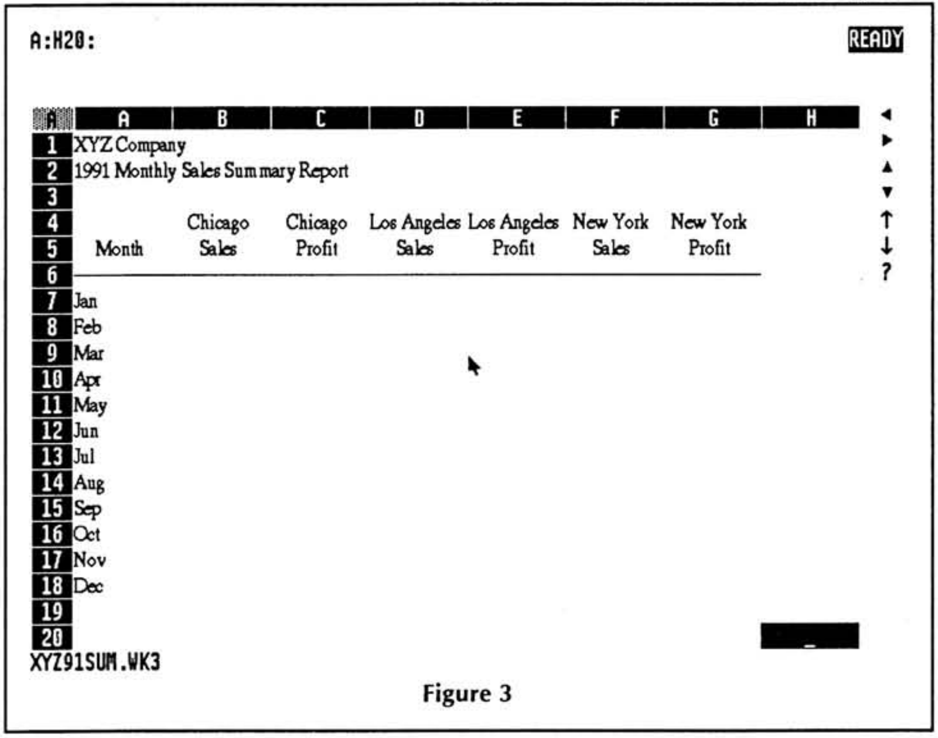
If you make a mistake when entering these labels, move the cell pointer to the cell containing the error, type a new entry, and press [ENTER]. You also may edit individual cells by highlighting the cell you wish to edit and pressing [F2]. This puts 1-2-3 in EDIT mode, which allows you to edit the entry in the second line of the control panel. You may use the left and right arrow keys as well as [HOME] and [END] to move the cursor when editing. The [HOME] key moves the cursor to the beginning of the entry, while the [END] key moves the cursor to the end of the entry. To delete characters, use [BACKSPACE] to erase the character to the left of your cursor, or [DELETE] to delete the character directly over your cursor. When you have made the necessary corrections, press [ENTER] and the edited entry will be placed in the worksheet.

Shortcuts That Speed Up the Process

Instead of manually entering the other labels, you can use 1-2-3 to copy the contents of one cell to another. Move the cell pointer to cell B4. Enter /Copy. 1-2-3 now asks, "Enter range to copy FROM:" Since you only want to copy the label under the cell pointer, press [ENTER]. Remember that a range of cells also may be just one cell. Now 1-2-3 asks, "Enter range to copy TO:" You want to copy to cell C4, so you could type "C4" and press [ENTER], or you can move the cell pointer to your destination and press [ENTER]. Since it is easier to point, move the cell pointer to cell C4 and press [ENTER]. The contents of cell B4 should now appear in cell C4. One advantage of copying cell contents this way is that the format of the cell is also copied. Notice that Chicago is centered in the cell at C4 just as it is in B4. Now use the same procedure to copy the contents of cell D4 to E4 and F4 to G4.

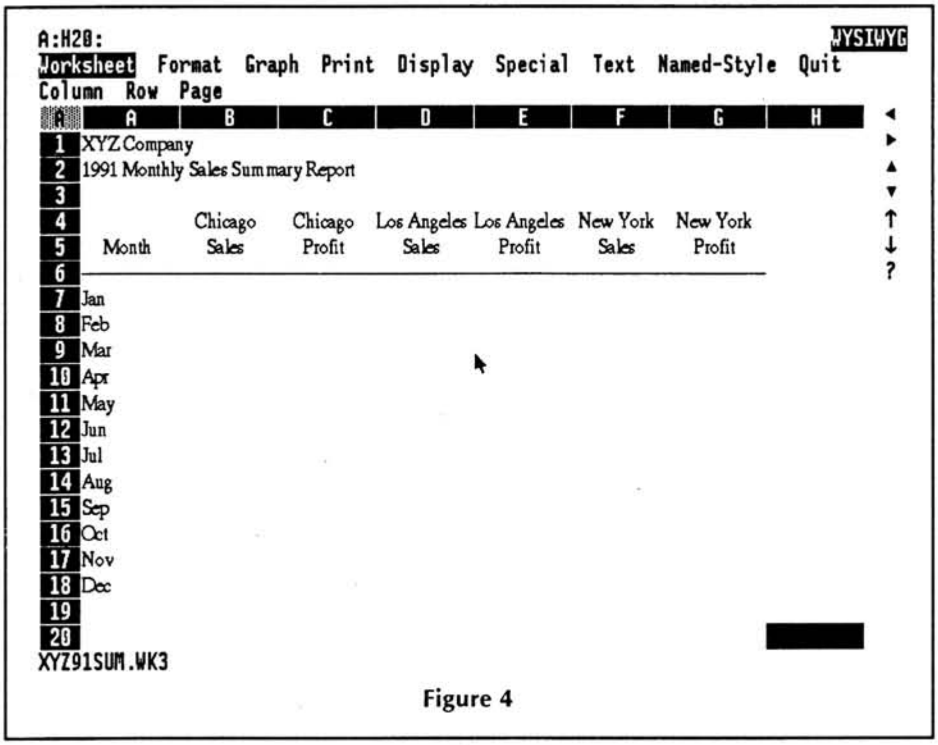
Move the cell pointer to cell B5. This time you can use the range copying capability of 1-2-3 to speed up the process of copying these labels. Enter /Copy, and when 1-2-3 asks, "Enter range to copy FROM:", move the cell pointer to C5, which is the end of your source range. Notice that both cells B5 and C5 are highlighted. Press [ENTER]. When 1-2-3 asks, "Enter range to copy TO:", move the cell pointer to cell D5, which is your destination or target range. There is no need to specify a complete target range in this case. 1-2-3 simply requires a starting point, and will copy as many cells as you highlighted in your source range. Press [ENTER], and 1-2-3 will complete the procedure by copying the contents of cells B5 and C5 to cells D5 and E5 respectively. Use the same procedure to copy the contents of cells B5 and C5 to cells F5 and G5.

Move the cell pointer to cell A6. Enter /Copy, and when 1-2-3 asks, "Enter range to copy FROM:", press [ENTER], since you only want to copy the contents of one cell. When 1-2-3 asks, "Enter range to copy your worksheet should look like Figure 3.



```
A:H20:                                                                READY

  A        A          B          C          D          E          F          G         H
  1  XYZ Company
  2  1991 Monthly Sales Summary Report
  3
  4                Chicago    Chicago   Los Angeles Los Angeles New York   New York
  5     Month      Sales      Profit     Sales      Profit      Sales      Profit
  6  -----------------------------------------------------------------------------
  7  Jan
  8  Feb
  9  Mar
 10  Apr
 11  May
 12  Jun
 13  Jul
 14  Aug
 15  Sep
 16  Oct
 17  Nov
 18  Dec
 19
 20
XYZ91SUM.WK3
```

Figure 3



```
A:H20:                                                                WYSIWYG
Worksheet  Format  Graph  Print  Display  Special  Text  Named-Style  Quit
Column  Row  Page
  A        A          B          C          D          E          F          G         H
  1  XYZ Company
  2  1991 Monthly Sales Summary Report
  3
  4                Chicago    Chicago   Los Angeles Los Angeles New York   New York
  5     Month      Sales      Profit     Sales      Profit      Sales      Profit
  6  -----------------------------------------------------------------------------
  7  Jan
  8  Feb
  9  Mar
 10  Apr
 11  May
 12  Jun
 13  Jul
 14  Aug
 15  Sep
 16  Oct
 17  Nov
 18  Dec
 19
 20
XYZ91SUM.WK3
```

Figure 4

TO:", you must now specify a range of cells, since you want the contents of one cell to be copied to many cells. Move the cell pointer to cell B6 and press the period (.) key, which tells 1-2-3 that this cell is the beginning of the target range. Move the cell pointer to cell G5, which is the end of the target range, and press [ENTER]. 1-2-3 will complete the process, copying the contents of cell A6 to the entire range of cells between B6 and G6.

After you have completed these steps,

Loading WYSIWYG

Now that you have the basic shell of the worksheet finished, you can make it look more presentable by using the WYSIWYG interface. If you are not sure if you have WYSIWYG loaded, type ":". The colon (:) is the access key for the WYSIWYG interface. If WYSIWYG is loaded you should see the WYSIWYG menu system as shown in Figure 4. The mode indicator in the control panel should read WYSIWYG.

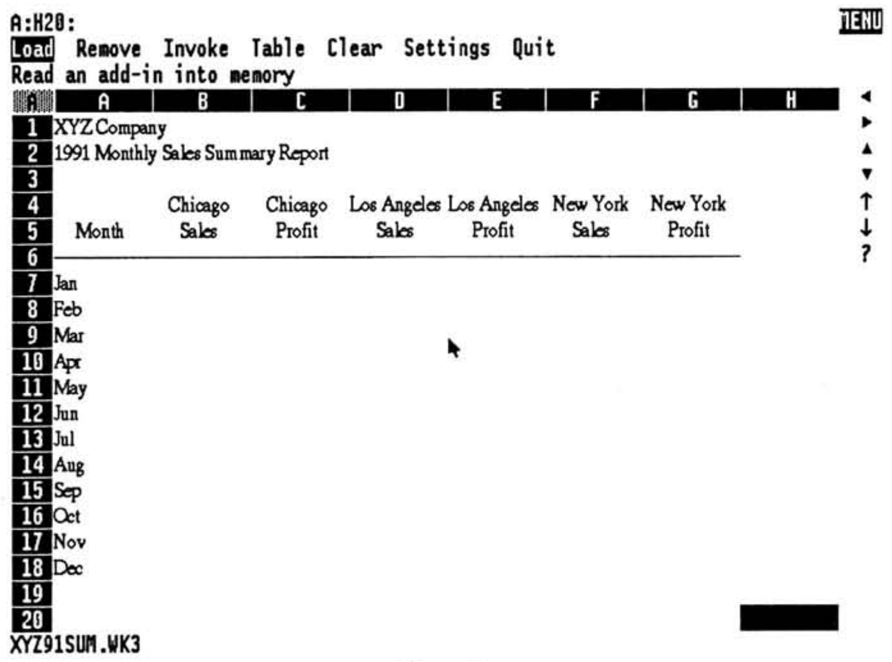
If your system already has WYSIWYG loaded, you may skip the next section, and proceed directly to the section titled "Using WYSIWYG."

If you need to load the WYSIWYG interface, press [ALT- F10]. You should now see the 1-2-3 add-in menu system, which looks like Figure 5.

Since you want to load the WYSIWYG add-in, highlight Load and press [ENTER] to read an add-in into memory. The WYSIWYG add-in should be stored in the ADDINS\ sub-directory, so highlight that sub-directory by using the arrow keys and pressing [ENTER]. Use the arrow keys to highlight the choice WYSIWYG.PLC and press [ENTER]. Now you are given the option to assign the add-in to a hot-key. Press [ENTER] when the No-Key option is highlighted, since you are going to set 1-2-3 to load this add-in automatically. You will now see the WYSIWYG opening screen and your worksheet will change its appearance slightly. Press Q to quit back to the worksheet. To configure 1-2-3 to load the WYSIWYG add-in automatically, enter [ALT-F10] Settings System Set ADDINS\ WYSIWYG.PLC Yes No-Key Update Quit Quit. The next time you start 1-2-3, the WYSIWYG add-in will be loaded and invoked automatically.

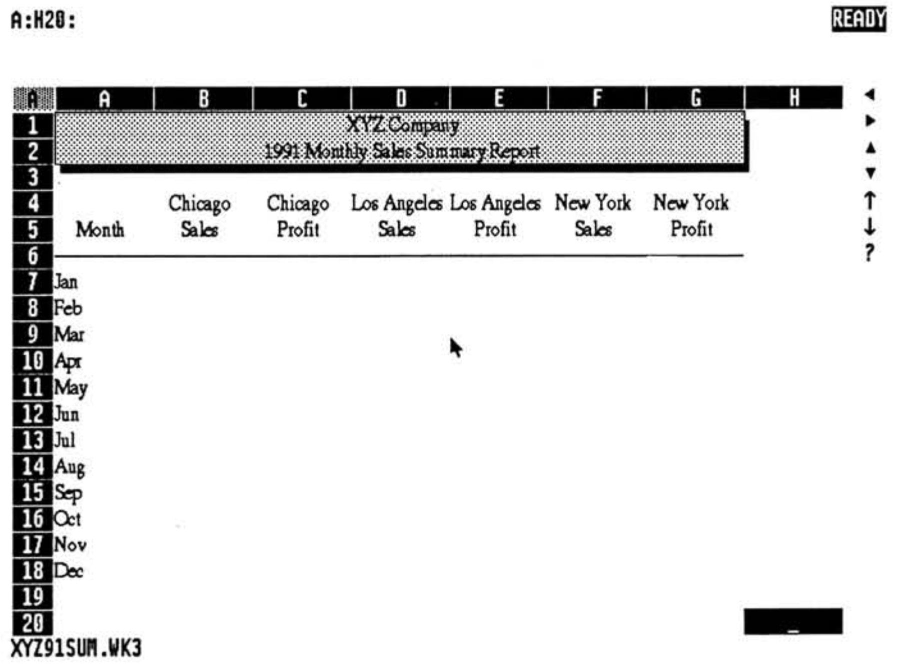
Using WYSIWYG

The WYSIWYG commands are entered in the same way you enter normal 1-2-3 commands, except you use the colon (:) key instead of the slash (/) key to access the menu system. You may still type the first letter of any command as a shortcut. If you refer to Figure 3, you will notice that rows 1 and 2 of your spreadsheet are not centered. Using WYSIWYG, centering is a snap. Move the cell pointer to location A1, and enter :Text Align Center. When 1-2-3 asks, "Select range to align:", move the cell pointer to cell G2. Notice that the entire range between cells A1 and G2 is highlighted. Press [ENTER], and 1-2-3 will center the text within this range. If you want to edit your text in the future, remember that the text was entered in cells A1 and A2. The text itself is still in those cells, even though WYSIWYG has centered it on the screen. With the cell pointer at location A1, enter :Format Lines Outline. Highlight the range between cells A1 and G2, and press [ENTER]. This draws an outline box around the range that you highlighted. To add a drop shadow to this box, make sure that the cell pointer is in cell A1 and enter :Format Lines Shadow Set. Highlight the range between cells A1 and G2, and press [ENTER]. A drop shadow should now appear below and to the right of your outline box. To add shading to the box, you would use :Format Shade Light, highlight the range of cells between A1 and G2, and press [ENTER]. If you completed these steps correctly, your worksheet should look like Figure 6.



```
A:H20:                                                              MENU
Load  Remove  Invoke  Table  Clear  Settings  Quit
Read an add-in into memory
 A      A        B         C          D          E         F         G        H
 1  XYZ Company
 2  1991 Monthly Sales Summary Report
 3
 4              Chicago   Chicago  Los Angeles Los Angeles New York  New York
 5    Month      Sales    Profit      Sales      Profit     Sales     Profit
 6
 7  Jan
 8  Feb
 9  Mar
10  Apr
11  May
12  Jun
13  Jul
14  Aug
15  Sep
16  Oct
17  Nov
18  Dec
19
20
XYZ91SUM.WK3
```

Figure 5



```
A:H20:                                                              READY

 A      A        B         C          D          E         F         G        H
 1                                XYZ Company
 2                    1991 Monthly Sales Summary Report
 3
 4              Chicago   Chicago  Los Angeles Los Angeles New York  New York
 5    Month      Sales    Profit      Sales      Profit     Sales     Profit
 6
 7  Jan
 8  Feb
 9  Mar
10  Apr
11  May
12  Jun
13  Jul
14  Aug
15  Sep
16  Oct
17  Nov
18  Dec
19
20
XYZ91SUM.WK3
```

Figure 6

The WYSIWYG interface makes it possible to create very professional looking reports and other documents with a minimum of effort.

Saving Your Work

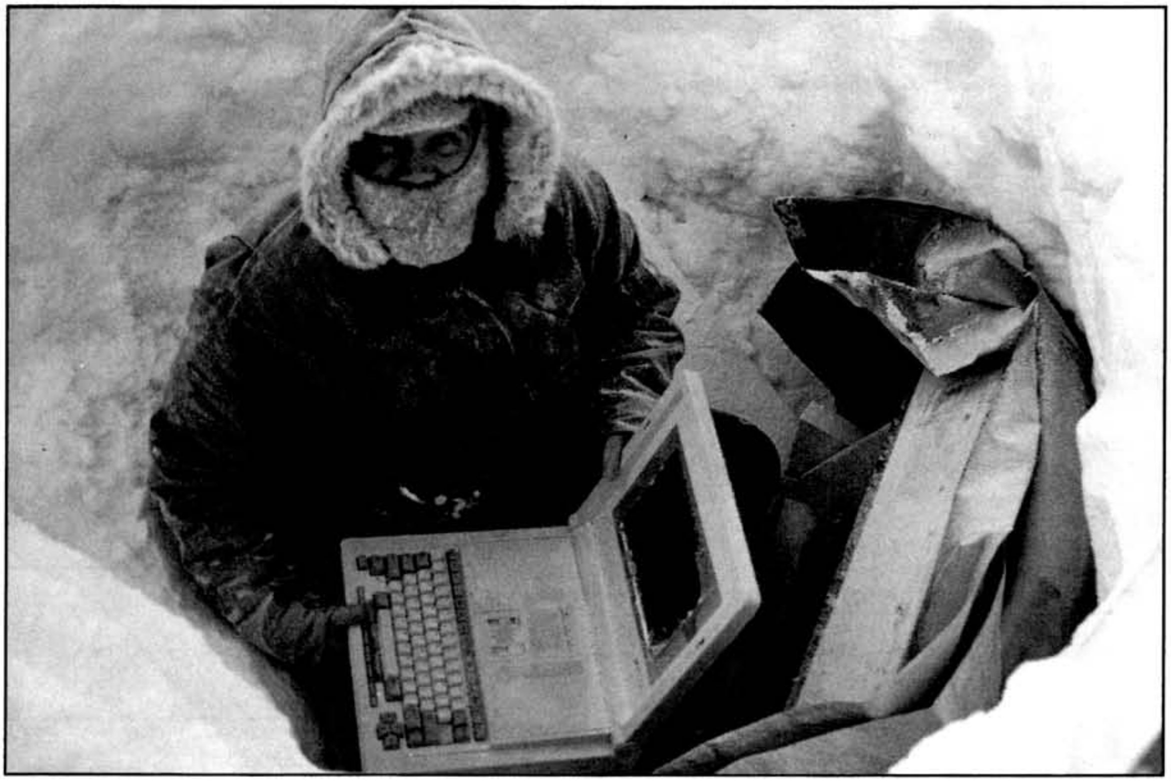
Now that you have spent considerable time creating your worksheet, it is time to learn how to save and retrieve it. To save a worksheet, enter /File Save. 1-2-3 will show you the subdirectory and file name under which this file will be saved. 1-2-3 will automatically supply a unique file name beginning with FILE0001.WK3. If this file already exists on your hard disk, 1-2-3 will increment the number portion of the file name until a unique file name is found. If you do not want to use the default file name, you must supply a file name of your own. When 1-2-3 asks, "Enter name of file to save:", type "XYZ91SUM" and press [ENTER]. You do not need to backspace over the old file name, although there is nothing wrong with doing so. 1-2-3 automatically adds the .WK3 extension and saves the file. To clear your worksheet out of memory enter /Worksheet Erase Yes. If you want to retrieve your worksheet again, enter /File Retrieve. When 1- 2-3 prompts you, "Enter name of file to retrieve:", type "XYZ91SUM" and press [ENTER]. Your saved worksheet will be retrieved from the disk and loaded into memory.

Quitting Time

To quit, enter /Quit Yes Exit. If you have not saved your work, 1-2-3 will warn you first before quitting.

Next Time

The purpose of Part 1 of this article was to give you some of the basics of Lotus 1-2-3, along with a taste of the more advanced features and how they are used. Future articles will cover the advanced features in much greater detail, and give you a sense of the sophistication of Lotus 1-2-3 release 3.1. ✻



Unusual Plant Life in the Arctic

David Dalton
Manager, Corporate Communications
Zenith Data Systems

George "Bub" Mueller uses a ZDS laptop computer to study a plant that grows in darkness rather than light.

The plant, Laminaria solidungula, is a type of kelp that grows primarily during the darkness of the Arctic winter. The plant has been found in only one place in the world, on the floor of the Beaufort Sea, off the northern Arctic coast of Alaska.

Mueller, a researcher at the University of Alaska Fairbanks Water Research Center, measures the amount of light that reaches the sea floor. Vegetation covers less than 1/10 of one percent of the sea floor and the kelp plant grows in a small area called the Boulder Patch. The patch contains stones and small boulders that serve as anchors for the plants against the sea's currents.

Mueller is interested in the area for two reasons. "The fact that it supports plant life and is also an oasis for other aquatic life is an unusual phenomenon," he said. "But, it is also located three miles from an oil-drilling rig, and the U.S. Environmental Protection Agency is concerned about the possible impact the rig might have on the plant."

As part of a five-year research project, Mueller and his colleagues have buried in the sea sensors that collect data on light availability. The unattended sensors take measurements for a year and then must be recovered. Each year, Mueller and his colleagues travel back to the sea, encountering temperatures as low as -20 degrees F, to retrieve the sensors.

Since 1986, the researchers have used a Zenith Data Systems' Z-183 laptop to download data from the sensors and take it back to Fairbanks for analysis. So far, the data have shown that the presence of the rig has not affected the plant's growth.

The ZDS laptop has withstood rough handling, sub-zero temperatures, blowing snow and sea spray. "Through all this extreme treatment, the laptop has never failed us," he said. ✻

Troubleshooting Windows:

Everything You Always Wanted to Know, But Didn't Know Who to Ask

Sanford Shapiro
654 Gravilla Street
La Jolla, CA 92037

Now, why would Microsoft, a major software company, come to my town and provide a free, lavish buffet lunch and a free Windows Technical Workshop for close to 1000 people? I have used many programs over the years, and never before has a company bought me lunch. Does this mean Microsoft is worried? Is their program, "Windows," in trouble?

Millions of copies of Windows have been sold, but many are no longer being used. People find Windows unstable and stop using it. Microsoft wants you to keep using their program, and they buy you lunch, hope you will be able to run Windows, and hope you will then support other Windows users. The Windows Technical Workshop is geared to the "Support Professional," people who are looking for "under the hood information." Support Professional is a fancy term for a company's local Window's expert. A whole new career has opened for anyone knowing how to get Windows up and running. And I thought "Navel Surgeon" was an exotic specialty!

A generation of computer users grew up using a keyboard for typing computer commands. A new generation of computer users, people uncomfortable typing commands, instruct their computers by using a mouse to point at pictures. The Apple Macintosh popularized the graphical-based user interface. Microsoft's Windows program has adapted the graphical user interface ("GUI") for the PC environment.

Windows puts a wall between you and your computer's operating system. On that wall are pictures that you point to and click on with your mouse. When it works, it is wonderful. You want to start a program? Just double click on the picture that represents your program – it is called an "icon." Or point to the heading on the "Menu Bar" called "File," and click (Figure 1). A menu of options will drop down (Figure 2). If you point to the "Open" command in this menu and click again, a list of files will appear (Figure 3). Just point to the file you want – click, and you are off and running. Or you can point to the "Run" command on the "File" menu and type in the name of your program.

When it doesn't work, you have a nightmare on your hands. Either you keep pointing and clicking until your screen is a hodgepodge of windows, or a window appears with the message: "Unrecoverable Application Error," the dreaded "UAE." Now you have to reboot your computer and start over. You can try clicking on the "Help" menu, but soon that too leads to an impenetrable maze of irrelevant information.

It used to be that you had a choice –

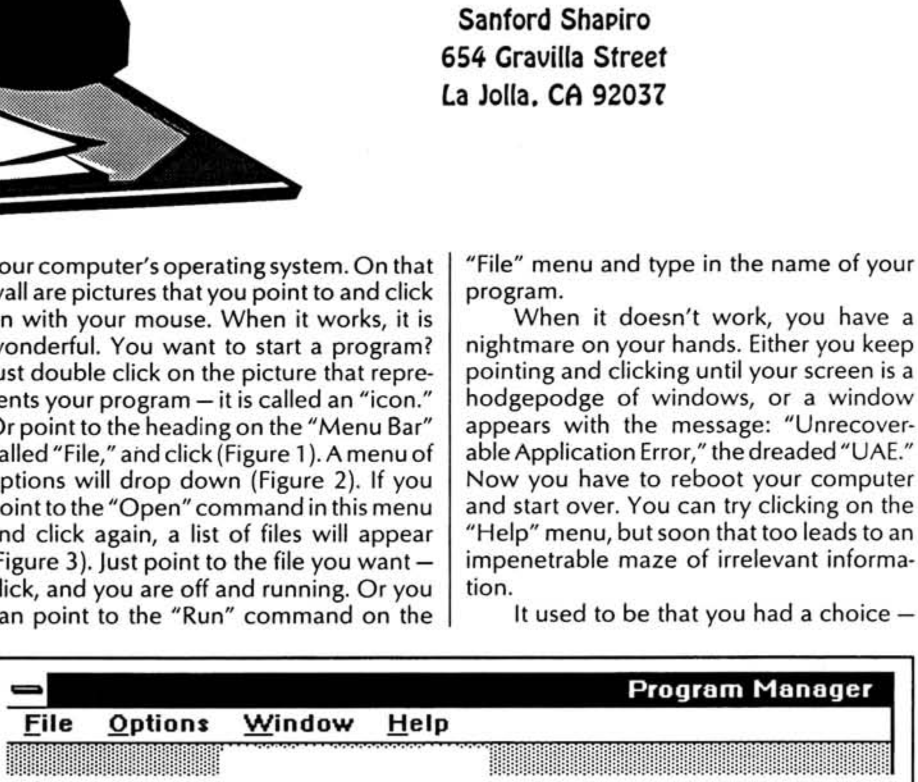


Figure 1

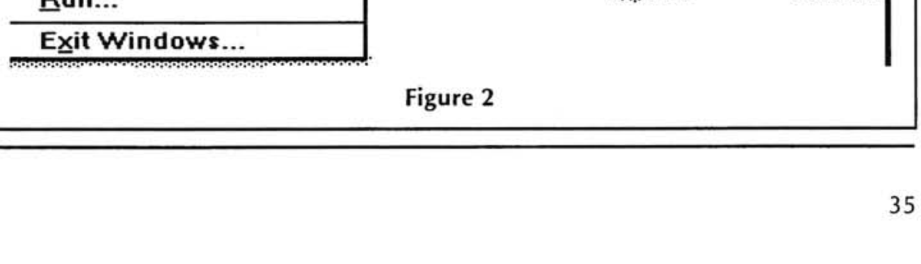
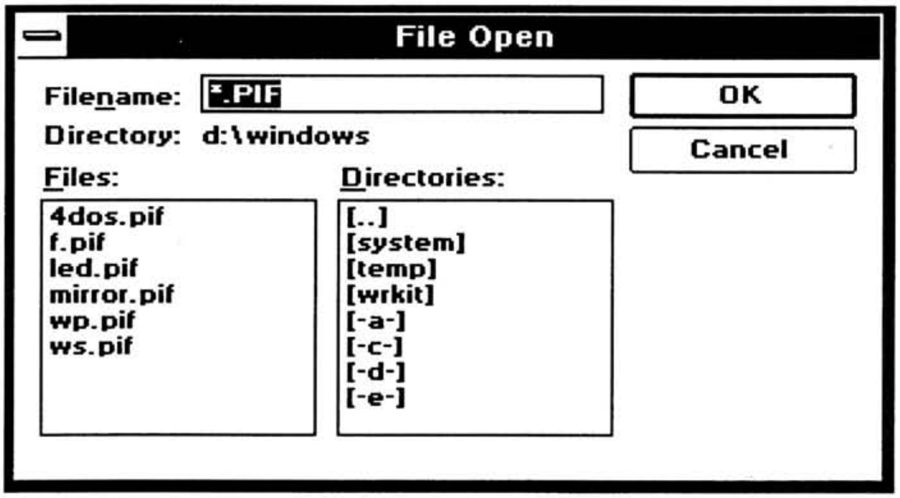


Figure 2



Filename: *.PIF

Directory: d:\windows

Files:
4dos.pif
f.pif
led.pif
mirror.pif
wp.pif
ws.pif

Directories:
[..]
[system]
[temp]
[wrkit]
[-a-]
[-c-]
[-d-]
[-e-]

Figure 3

if a program didn't work for you, you found another one that did. But not any more. Many new programs now can only be run under Windows. For instance, if you use the popular desktop publishing program "PageMaker," you have to run Windows. And my fondness for the spreadsheet program, "Excel for Windows," means I have to use Windows, too. So some of us are being dragged kicking and screaming into the world of Windows. If you can't fight them — join them. And that is why I went to the Windows Technical Workshop.

Do I now like Windows any better? No, but at least I have the satisfaction of knowing I can make it work. The Windows Technical Workshop covered five topics: Architecture and memory, Setup, Managing the GUI migration, Troubleshooting, and Optimization. (A sixth topic, Networking, I will skip because I lack personal experience.) The following information is based on my notes and on information from the manual: "Microsoft Windows 3.0 Technical Workshop, Fall, 1991."

The first thing our speaker told us was that Windows 3.0 is unstable — users get too many "Unrecoverable Applications Errors." A free maintenance release, Windows version 3.0a was available for the asking. When I got home, I called Microsoft at (206)454-2030 and asked for the upgrade. "Are you having a problem with Windows?" I was asked. "I get a lot of UAEs," I answered. "What size disks would you like?" They said. No further questions were asked, and forty-eight hours later Federal Express delivered the upgrade. I do get fewer UAEs now.

Architecture and Memory

Programs, or "applications" as Microsoft calls them, fall into two categories: Windows applications and DOS applications. Examples of DOS applications are WordStar and Lotus 1-2-3, programs not designed to share displays, memory, printers or other system resources. Windows applications are designed to share and, when running concurrently, take turns in using printers, displays and communications ports.

Windows operates in one of three modes: Real, Standard or Enhanced, and Windows checks your system to determine the best mode. I will describe the differences between these modes.

Real mode is similar to Windows 2.x but with a 3.0 interface. It works on 8088 type computers and is limited to 640K of memory for executing Windows code and data. Standard mode requires an 80286 or better computer and breaks the 640K barrier. In Windows standard mode, applications seamlessly access both conventional and extended memory as one single memory pool. Even if you have an 80386 computer, if you only run Windows applications, Standard mode will be 20% faster than Enhanced mode.

Enhanced mode is slower because its operations have more overhead. However, if you run DOS applications, then standard mode is slower because Windows and Windows applications are swapped to disk.

Enhanced mode requires an 80386 or better computer and introduces a new kind of memory: "Virtual Memory." [Other memory types: Conventional, Extended (XMS), Expanded (EMS), Upper (UMB) and High (HMA) have been described in other REMark articles and will not be repeated here.] Virtual memory is like pretend memory. Windows takes part of the storage area on your hard disk and creates a "Swapfile" where code can be swapped in and out of RAM as needed. This area of hard disk storage is added to your computer's pool of conventional and extended memory. To Windows and Windows applications, it is a single huge memory pool.

If you have DOS programs that use expanded memory outside of Windows, Enhanced mode also allows use of an expanded memory manager, such as EMM386.EXE, to make expanded memory available. Another feature of enhanced mode is the use of protected memory space — memory space created by the 80386 hardware — to create "virtual machines." A virtual machine is an area of memory made to look like conventional memory to DOS — like a separate computer. With several virtual machines, multiple DOS applications can be run concurrently. Each DOS application is contained within its own virtual machine, while Windows and all Windows applications are contained within one virtual machine.

Setup

An undocumented feature of Windows 3.0 is the Setup Information File ("SETUP.INF"), an editable ASCII file. If, for instance, you are installing Windows on multiple machines, you can customize your "SETUP.INF" file to: automatically change the default directory, to specify printer and display options you want installed, to define Program Information Files (PIFs) for DOS applications, and to define programs you want to run automatically at the end of Setup (such as "Swapfile"). In a network environment, for example, a customized "SETUP.INF" file can be stored on a server and, when users run setup, the customized Windows environment will be installed.

Windows installation takes place in two modes, a DOS mode and a Windows mode. The DOS mode installs the different pieces of Windows and then loads Windows in "real mode." In the Windows mode; printers, applications, and PIFs are installed, and the Program Manager groups are built. If real mode Windows does not appear during the installation, check for an incompatible display adaptor. When real mode Windows appears, you know that DOS has successfully installed all the right pieces of Windows.

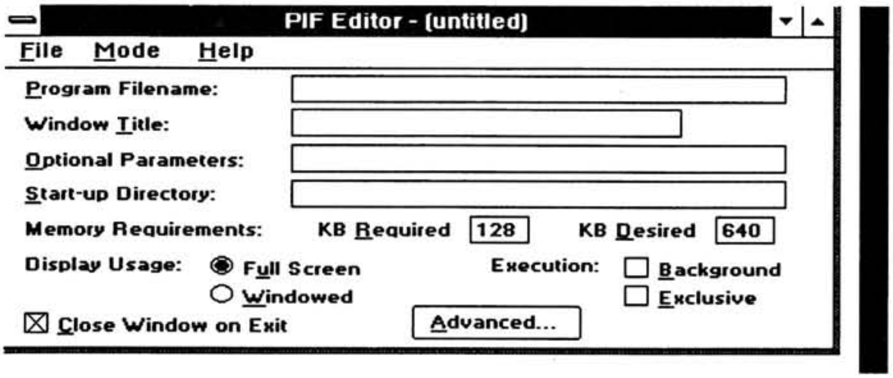
Setup is also used to maintain Windows when changing a printer or display driver, or if your group files become damaged. Using the "/p" parameter (Setup/p), Windows will rebuild the Main, Accessories and Games groups. And be sure to make a backup copy of your "SETUP.INF" file if you add an OEM driver. Some OEM's, such as Tseng Labs, Renaissance and Logitech, will copy their own "SETUP.INF" file over the Windows "SETUP.INF" file when they install their drivers. Jerry Pournelle, writing about setup programs and their problems in the January, 1992 issue of BYTE Magazine, says about important files: "Do not save them on the same machine. I mean that literally. Save them on floppy disks, or use LapLink to connect to a different machine entirely and then disconnect the wires between machines. I wouldn't even leave the computer connected to a network." Jerry lost much work when he updated one of his programs to a new version.

Managing the GUI Migration

Changing to a graphical-based user interface requires changes in both hardware and software. Microsoft's answer to GUI hardware problems is: "Spend more money." Windows is a resource hog. You need a faster, more powerful CPU, a larger hard drive, a faster display and lots and lots more RAM. You can successfully run Windows in an 80286 computer if you add extended memory and a disk caching program. (Hard drives in some older 80286 computers tend to be slow.) An 80286 with 5 MB RAM will outperform an 80386 SX with 1 MB RAM.

Moving your DOS programs to the Windows environment, and getting them to work smoothly, involves tweaking your program information files (PIFs). Running your DOS programs in real or standard mode, you can task switch (alternate between programs—those in the background are suspended in operation), and run your programs full screen. But in 80386 enhanced mode, you can multitask virtual machines (programs in the background continue running), and run programs either full screen or in a window.

PIF settings determine how Windows runs a DOS application. There are three main sources of PIFs: Windows provides PIFs for many popular DOS applications; vendors provide PIFs; and, using the Windows PIF editor, you can create your own customized PIF. The PIF editor menu is straightforward. But a few of the options are confusing and need explanation. Working in Enhanced mode, you have two menus: the main menu and the advanced menu. In the main menu you are faced with the option: "Display Usage." (Figure 4). This option determines whether your program will run full screen or in a window. Running your program full screen improves performance. You can also toggle between full screen and window mode by typing "ALT + Enter."


PIF Editor - (untitled)

File Mode Help

Program Filename:

Window Title:

Optional Parameters:

Start-up Directory:

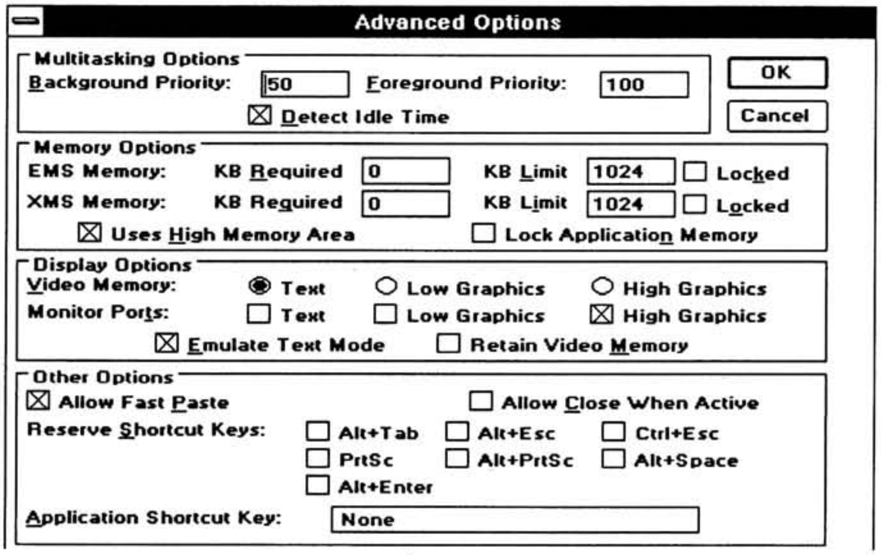
Memory Requirements: KB Required 128 KB Desired 640

Display Usage: ● Full Screen ○ Windowed

Execution: □ Background □ Exclusive

☒ Close Window on Exit

Figure 4


Advanced Options

Multitasking Options

Background Priority: 50 Foreground Priority: 100

☒ Detect Idle Time

Memory Options

EMS Memory: KB Required 0 KB Limit 1024 □ Locked

XMS Memory: KB Required 0 KB Limit 1024 □ Locked

☒ Uses High Memory Area □ Lock Application Memory

Display Options

Video Memory: ● Text ○ Low Graphics ○ High Graphics

Monitor Ports: □ Text □ Low Graphics ☒ High Graphics

☒ Emulate Text Mode □ Retain Video Memory

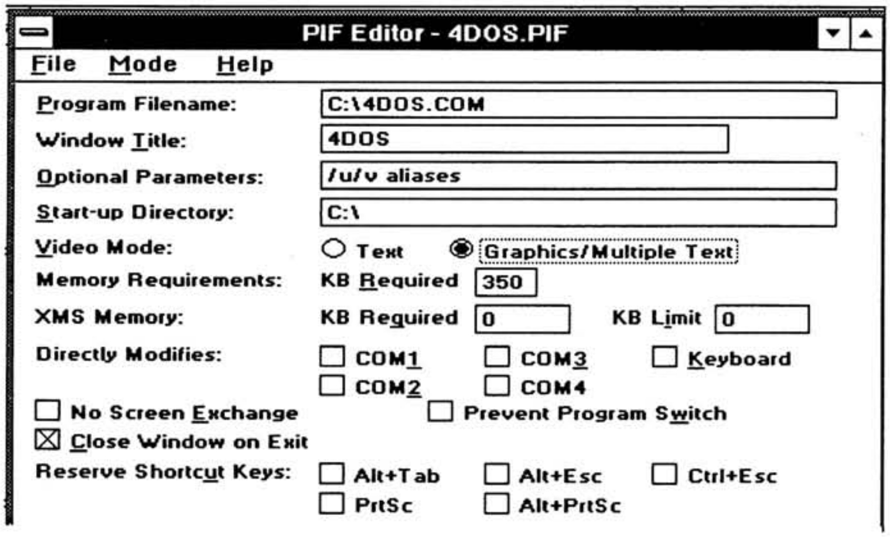
Other Options

☒ Allow Fast Paste □ Allow Close When Active

Reserve Shortcut Keys: □ Alt+Tab □ Alt+Esc □ Ctrl+Esc □ PrtSc □ Alt+PrtSc □ Alt+Space □ Alt+Enter

Application Shortcut Key: None

Figure 5


PIF Editor - 4DOS.PIF

File Mode Help

Program Filename: C:\4DOS.COM

Window Title: 4DOS

Optional Parameters: /u/v aliases

Start-up Directory: C:\

Video Mode: ○ Text ● Graphics/Multiple Text

Memory Requirements: KB Required 350

XMS Memory: KB Required 0 KB Limit 0

Directly Modifies: □ COM1 □ COM3 □ Keyboard □ COM2 □ COM4

□ No Screen Exchange □ Prevent Program Switch

☒ Close Window on Exit

Reserve Shortcut Keys: □ Alt+Tab □ Alt+Esc □ Ctrl+Esc □ PrtSc □ Alt+PrtSc

Figure 6

In the "Advanced Menu," (Figure 5), you have the additional options: Video Memory, Detect Idle Time, and Monitor Ports. Video Memory: This option saves memory for restoring the video state and for copying screen contents onto the clipboard. The "Text Mode" option uses the least memory, and you should try using that one first. Detect Idle Time: Clearing this option will speed up your program, but it also slows the rest of your system. Monitor Ports: Generally not necessary to check unless your application interacts directly with hardware ports. In standard mode, you have only a relatively uncomplicated "Main Menu" to deal with. (Figure 6).

Troubleshooting Windows

This part of the workshop covered, in rapid fire order, nine areas: Setup, Real mode, Standard mode, Enhanced mode, Networks, Printing, Displays, UAEs, and Applications.

Setup

"Smart" users make wrong decisions. Problems installing Windows can be caused by skipping modifications to Config.sys and Autoexec.bat files. Be careful that all important files get loaded.

If your installation hangs after "Setup" asks for Disk #2, you likely have a display driver problem. Try running "Setup/i." The "/i" parameter prevents autodetection of hardware.

TSR or device drivers dated 1987 or older can cause problems. Also, make sure you don't have a TSR that reduces your RAM below the 384K required to install and run real mode Windows.

Real Mode

If problems develop using real mode after a successful setup, you may have changed a display adapter, or your available RAM may have dropped below 376K. Also be sure your path statement still points to the proper subdirectory.

Standard Mode

If Windows runs in real mode but not in standard mode, check your XMS memory manager. If HIMEM.SYS is your memory manager, be sure you have the latest version and check that it is either in the root directory or a directory that can be found on bootup. Also make sure you have at least 256K conventional RAM and 192K extended RAM. Sometimes a RAMdrive will steal your available extended memory when you are not looking.

80386 Enhanced Mode

Windows, in enhanced mode, will look for upper memory blocks (UMBs) to use. This can cause conflicts with competing hardware that has not been detected. You can prevent Windows from using UMBs (it will use conventional memory instead) by adding a line to the [386enh] section of your "System.ini" file: "Emmexclude=A000-EFFF." If that solves your problem, you can experiment with smaller "exclude" statements. For instance, if you suspect a video conflict, use the line: "Emmexclude=C000-C7FF." Hardware adapter cards such as networks and scanners may use other areas of upper memory. A little detective work may be needed to find out which memory areas are being used.

Network Issues

Random UAEs or printing problems may be from incorrect shell versions — shells must be updated. Interrupt conflicts, such as IRQ 2 settings, also cause problems with Windows and should be avoided. And some networks cause problems by using the memory area D000-DFFF. Excluding this area from Windows will solve those conflicts.

Printing Errors

On networks, print jobs may timeout when sending to the network printer. You can adjust the necessary parameters by using the "Control Panel." And if your network was installed after installing Windows, you may need to run Setup again to properly configure your network. In stand-alone computers, if you can print from DOS but not from Windows, check the "SET TEMP" statement in your Autoexec.bat file. The "Temp" subdirectory is used by the Print Manager for print spooling.

Display Errors

Be sure you have the latest version of the Windows display driver that supports your display hardware. If you have an ATI Wonder Card and get a rolling screen, turn switch #8 to "off." That prevents a nonmaskable interrupt problem.

UAEs

Dealing with UAEs is a major frustration — like fixing a leak in corroded plumbing. First, be sure you have the maintenance release, Windows 3.0a. Then check the dates of your applications. An older application may need to be updated. Try removing all TSRs and putting them back one at a time — a suspect TSR may need to be eliminated. Check your version of DOS. Sometimes a proper OEM version of DOS (V.3.10 or higher) is what is needed. And increasing the environment size also may help.

For programmers, calling the Microsoft Product Support Services BBS, (206) 637-9009, and downloading the file DrWatson.exe, will give you a diagnostic program. Have Dr. Watson installed when running Windows and, if a UAE appears, Dr. Watson will provide a technical log of data you can use in tracking the error's source.

Applications

Problems with DOS applications may require that you allocate more memory using your PIF file. And mouse problems can be avoided by using the latest mouse driver and only loading it, by using a batch file, in each DOS virtual machine.

Optimization

If you have enough RAM, you can improve performance by increasing the size of your disk cache. A disk cache larger than 2 MB, however, will not offer significant performance improvement. RAMDrives are not as efficient a use of memory as a disk cache, like SmartDrive. If you have enough RAM, however, and you have already allocated 2 MB to a disk cache, then using additional memory for a RAMDrive will improve performance. And, of course, regular defragmenting of your hard disk is essential.

Windows Resource Kit

Microsoft provides a collection of utilities and reference materials, the "Windows Resource Kit," which should have been part of the original Windows package. For $20.00 ($14.50 plus $5.50 shipping), you get a technical reference manual that is, finally, a real manual, and you get a series of utility programs including "Learn Windows." Learn Windows, a tutorial program, leads you step by step through each of Windows' features. Most people learn Windows by playing Solitaire. The Learn Windows program is a big improvement.

Another utility program that I like is the screen saver, "FISH," a clever shareware program. Instead of just giving you a blank screen, like other screen saver programs, FISH provides a variety of fish peacefully swimming back and forth across your screen. You can order the Windows Resource Kit, or sign up for a Windows Technical Workshop, by calling (800)642-7676.

The graphical user interface is here to stay. Whether Microsoft Windows will be the answer, or another system will — like IBM's OS/2 — only the future will tell. Meanwhile, Microsoft has announced Windows Version 3.1, which they promise: ". . . will offer significant improvements in stability, ease of use, and performance." I only hope the new Windows is as good as the free lunch.

Continued from Page 15

```
      for (i=0; i < 717; i++)
         c = getc(data);

      cardno = 0;
      while (!feof(data)) {
         fread(&card[cardno], sizeof(RECORD), 1, data);
         fread(filler, sizeof(filler), 1, data);
         putEOS(card[cardno]);
         cardno++;
      }

      qsort(card, cardno, sizeof(RECORD), strcmp);
      PrintAddr(stdprn, card, cardno, skip);
      fclose(data);
   }
```

Parsing the DOS Command Line:

A QuickBasic 4.5 Routine

John Day
59, rue Sauer
77500 Chelles, FRANCE

Background

One of the useful features introduced near the beginning of Basic compilers for DOS was access to the command line. If you start your program with, say,

```
C>progname file1,,file3 /H /Kp
```

you can use a system variable COMMAND$ in your Basic program which will be loaded by Basic with

```
FILE1,,FILE3 /H /KP
```

This is handy if you want to start your program quickly with a few values input from the keyboard, and downright essential if you use batch files to run your programs. But... you get the command line all in a lump, and it's up to you to split off the different values. After having written twenty code sequences in twenty different programs to extract two or more fields from the command line, I started wondering whether there wasn't some standard way to do it. Here's the result: a Basic subroutine that parses the command line and puts the fields in consecutive entries in a string array. Once this has been done, your program can scan down the array and use the values directly. It was rewritten and simplified from a first version which has been in production for six months now; I added the double quote "" which is parsed as a single quote ", and which only took four extra lines.

Syntax is more or less that used by DOS commands, which is different from that used by the batch processor and seems more useful. The standard separator is the comma ",", and multiple commas ",," define nul entries. The space " " is an allowed separator, but any combination of spaces and one comma will only define one value. These separators are not included in the resulting array. Switches are marked by a stroke "/", which is a leading separator and which is copied to the array. Values within quotes "like this" are copied to the array without parsing (but Basic does shift them to capitals first); double quotes within quotes are copied as a single quote. The command line at the beginning of the article would give array entries:

```
FILE1          0
               1  [nul]
FILE3          2
/H             3
/KP            4
```

If you want to try the subroutine to get the feel of it, the header code will print several entries from your command line, within arrows >LIKE THIS< so you can see spaces (which you'll only get within quotes). To use the subroutine with one of your programs, move the DEFINT instruction into the subroutine and drop the test lines.

The subroutine picks up the size of the string array when it is called, and stops parsing if it would run out of bounds. All you have to do is define the array, and call the subroutine with the array descriptor:

```
CALL ParseCom StringArray$()
```

When the subroutine returns control, the values from the command line will be loaded into consecutive entries in StringArray$. You can include the subroutine in your program code, or compile it separately and link it with your program:

```
C>link progname parsecom;
```

or compile it and include it in a user library. The code was written for MicroSoft QuickBasic 4.5, runs under PDS 7.0 Basic, and should run under 4.0. For 3.0, you should replace

```
      DIM Quote AS STRING*1
      Quote = CHR$(34)
```

with

```
      Quote$ = CHR$(34)
```

and change "Quote" to "Quote$" throughout. The rest *should* work, although I haven't access to a QuickBasic 3.0 compiler to check. If you're still at version 2, you'll have to rewrite the whole thing, unless you decide to upgrade — if you decide to do this, you'll get all sorts of useful things like CASE constructs, DO loops, EXIT FOR and EXIT DO statements and so on.

The Program

Variable "State" defines what's going on. "0" means that the routine has finished a value without finding a comma, and "1" means the value was closed by a comma. This lets the program handle a mix of commas and spaces without putting in superfluous nuls; spaces are ignored, and commas are only logged if another comma was found previously. "2" means parsing a value, and any separator will close the array entry. State "3" is used to parse within quotes, and disables all delimiters except the closing quote. The final parameter on the line won't usually be delimited by anything, so an afterthought puts it into the array.

```
      DEFINT A-Z
      DECLARE SUB ParseCom (Parms$())
      DIM Parms$(3) ' any size you want
      ParseCom Parms$()
      FOR I = 0 TO 3' same as DIM
      PRINT ">"Parms$(I)"<"           ' echo back for testing
      NEXT I
      END
'
'
================================================================
'  ParseCom
'     Return command line parameters in PARMS$
'     Separators: space, comma separate two fields
'        / separates two fields, is included
'        "..." delimits a field
'
================================================================
      SUB ParseCom (Parms$())
      I = LBOUND(Parms$)
      DIM Quote AS STRING*1
      Quote = CHR$(34)
      FOR Char = 1 TO LEN(COMMAND$)
         SELECT CASE MID$(COMMAND$,Char,1)
            CASE " "
               IF State = 3 THEN    ' don't parse within quotes
                  Parm$ = Parm$ + " "
               ELSEIF State > 1 THEN ' take space as separator
                  Parms$(I) = Parm$
                  Parm$ = ""
                  State = 1
                  I = I + 1
               ENDIF
            CASE ","
            IF State = 3 THEN          ' don't parse within quotes
               Parm$ = Parm$ + ","
            ELSEIF State = 1 THEN ' ignore comma following other
               State = 0               '  delimiter
            ELSE
               Parms$(I) = Parm$
               Parm$ = ""
               State = 0
               I = I + 1
            ENDIF
         CASE "/"
               IF State = 3 THEN    ' don't parse within quotes
                  Parm$ = Parm$ + "/"
               ELSEIF Parm$ = "" THEN
                  Parm$ = "/"      ' leading stroke: include
                  State = 2
               ELSE
                  Parms$(I) = Parm$       ' trailing stroke: log
                  Parm$ = "/"             '  previous
                  State = 2
                  I = I + 1
               ENDIF
            CASE Quote
               IF State = 3 THEN
                  IF Char < LEN(COMMAND$) AND_
                     MID$(COMMAND$,Char+1,1) = Quote THEN
                     Parm$ = Parm$ + Quote
                     Char = Char + 1 ' skip double quote
                  ELSE
                     Parms$(I) = Parm$    ' closing quote
                     Parm$ = ""
                     State = 1
                     I = I + 1
                  ENDIF
               ELSEIF State = 2 THEN  ' opening quote as delimiter
                  Parms$(I) = Parm$
                  Parm$ = ""
                  State = 3
                  I = I + 1
               ELSE  ' opening quote
                  State = 3
               ENDIF
            CASE ELSE   ' character to include
               Parm$ = Parm$ + MID$(COMMAND$,Char,1)
               IF State < 3 THEN
                  State = 2
               ENDIF
         END SELECT
         IF I > UBOUND(Parms$) THEN   ' abandon if array is full
            EXIT FOR
         END IF
      NEXT Char
      IF Parm$ <> "" AND I <= UBOUND(Parms$) THEN
         Parms$(I) = Parm$          ' end of line without delimiter
      END IF
      END SUB
```

Continued from Page 5

to be removed. A set of three screws located along the bottom rear, a set of two screws located on each side about half way back from the front of the computer, and the set of three screws located on the front lip of the computer. All three sets are Phillip head screws.

7. After removing these 8 screws, carefully grasp the computer and turn it upside down making sure that the top part of the case does not separate from the bottom half.
8. Now that the computer is setting right side up with the keyboard end facing you, you should be able to separate the two halves of the case. You want to lift the upper portion of the case about a quarter of an inch above the lower half and lay the top part of the case behind the lower part of the case with the face down (Open the case as though there was a hinge located along the back of the case). When completely open the two halves should be back to back. It is necessary to lay the two halves this way because the cable between the motherboard and the display determine the maximum distance at which the two halves can be placed.

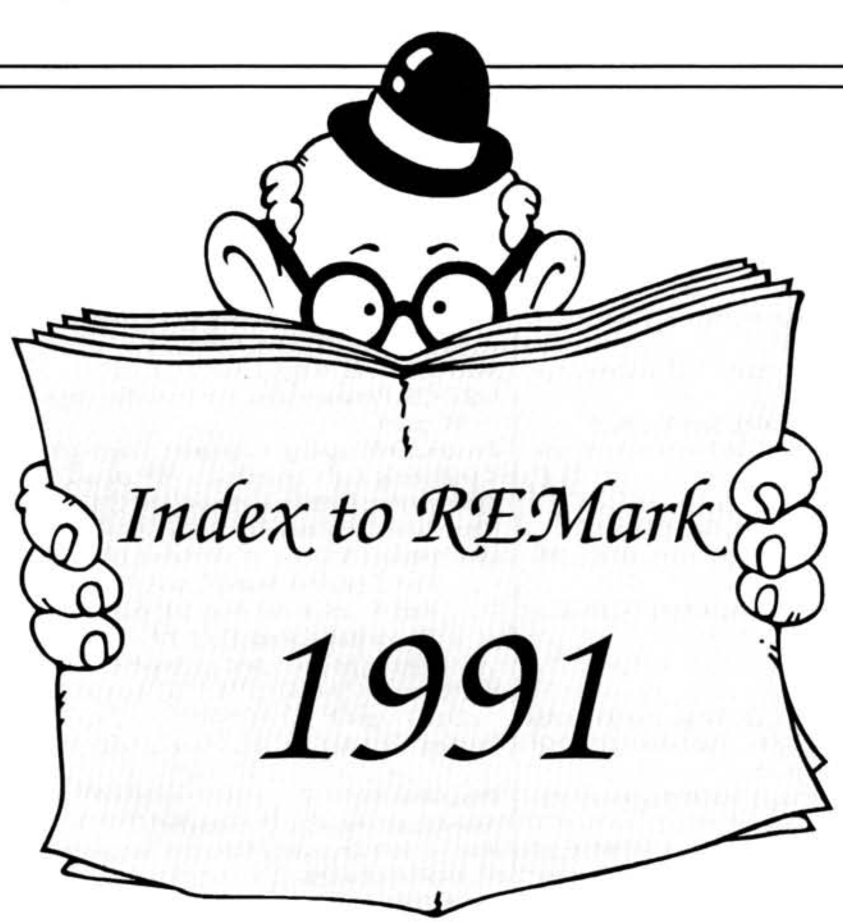
Congratulations! You should now be staring at the inside of your computer. You might be surprised at what you find inside. Over time dirt and sometimes tumble weed filters its way through the keyboard into the case. The third time I opened my case I found what appeared to be a cobweb located in the slot designed to hold a modem.

The connector for the display cable is located on the top rear of the motherboard just to the left of center. Even if the cable appears tight it is best to pull the cable off and then re-attach it. Sometimes a layer of oxide develops between the contacts. On my system a clamp has been attached to the connector so that the cable will not work its way free. Rather than removing the connector completely I carefully jiggle the wires going into the connector to make sure that they are seated properly. I have successfully fixed three computers using this method and saved an unknown amount in repair costs and down time.

Before putting the system back together you might want to clean it. That is the reason for the Q-Tips and double-sided tape. Place a strip of tape on the cotton end of the swab and run the Q-Tip through the machine. Most hair and tumbleweed, alias dust bunnies, will be removed from your system using this technique.

Now that your cable has been reseated and the machine is clean you can begin to reassemble your system. First, carefully flip the top half of the case over the bottom half as though there was a hinge on the back half of the unit. Make sure that the cable is placed properly in the case and does not become tangled on any protruding parts. This will restrict the motion of the LCD. To assemble the laptop simply follow the step-by-step instructions, provided above, in reverse order.

If you were successful the monitor should be as good as new. If the monitor is still not behaving normally you might want to try the process again since you might not have seated the cable just right. At one time I had to repeat the process twice before the cable made good contact with the pins on the motherboard. If the problem still persists you will have to make a trip to the service center.



Index to REMark 1991

Jan Axelson
2209 Winnebago Street
Madison, WI 53704

REGULAR COLUMNS & FEATURES

Buggin' Zug (letters)

SupersPort 8 & enhanced keyboard, Z-170 expansion, Shore ZUG update, Jan 91, p.4
Hard drive for portables, problem with AutoCAD Rel. 11, laptop displays, laptop battery care, Z-100 screen problem, Prodigy in color on Z-158, Nov 91, p.5

On the Leading Edge (W. Adney)

Computer model numbers, Jan 91, p.19
Windows 3.0, static electricity and humidity, LaserJet III, your HP LaserJet handbook, Feb 91, p.35
LaserJet III printing with Windows 3.0, Imager, backup alternatives, backup strategies, Mar 91, p.11
Physical memory, expanded vs. extended memory, memory managers, QRAM, Manifest, QEMM386, DESQview, DESQview386, May 91, p.9
Computek Systems, Quattro Pro 3.0, Software Bridge, Outside In, Jul 91, p.45
More on memory: UMBs and HMA, system configuration, Lotus 1-2-3 upgrade, speaking of upgrades, Aug 91, p.21
Setting up QEMM-386 on a Z-386, Computek Systems, Oct 91, p.19

Port-O-Call: COM1 (L. White)
(excerpts from and information about the ZUG BBS)

Windows 3.0 Problems, Jan 91, p.39
Additions and Changes to ZUG BBS, Feb 91, p.9
Bargain Centre, Mar 91, p.33
HUG-386-25 upgrade, Apr 91, p.5
MinisPorts, May 91, p.5
9600 Baud modems, detecting viruses, protecting files, Jul 91, p.31
Desktop publishing, diagnostic LEDs, IDE controllers, QEMM with an ESDI drive, TurbosPort and VGA, 386/16 upgrade to 20 MHz, Sep 91, p.43

Powering Up, Volume 2 (W. Adney)

How to use ANSI.SYS, Jan 91, p.34
Powering up the PROMPT command, Feb 91, p.17

HARDWARE

How-To

Desktop Publishing: Improving Your Image (hand scanners and digitizers), (M. Haverstock), Jan 91, p.5
EaZy PC, All Around, (L. Vesei), Oct 91, p.15
Expanded Memory, Adding to Your PC (Everex 3000 Delux AT Memory Board), (D. Jerome), Nov 91, p.31
Expanded Memory: What Is It All About?, (J. Eichenberger), Oct 91, p.27
Floppy Drive, Adding a 2.88Mb 3.5" to Your Computer, (R. Siebers), Aug 91, p.33
Fonts on the Fly, (A. Neibauer), Jul 91, p.41
Hard Disk Survival Kit, (H. Fale), Sep 91, p.18
Hard Drive, External Portable, for Your Laptop or PC-Compatible (ex-PORT hard drive by Lightek), (H. Fale), Oct 91, p.25
HUG-386-25 upgrade, (L. White), Apr 91, p.5
I Hate Progress! (expanding an H-151), (R. Kehlmeier), Jan 91, p.9
Keyboard, Using a Full Size, on a Laptop Computer (handy utility), (P, Swayne), Nov 91, p.21
Laser Printer Font Tools, Getting Started with, (A. Neibauer), Sep 91, p.39
Memory Mystery, Clearing Up the, (P. Swayne), Dec 91, p.23
Mice and Pens, Getting Started with, (A. Neibauer), Mar 91, p.43
MinisPorts, Care and Feeding of, (M. Schutz), Jul 91, p.5
Modem: A Data Transfer Solution (R. Brenner)
  Part 2 (innovative designs), Feb 91, p.11
  Part 3 (data compression), Apr 91, p.39
  Part 4 (standards), Sep 91, p.45
  Part 5 (protocols), Oct 91, p.45
Monitor, Choosing a Computer, (D. Veit), Oct 91, p.7
Planes, Hotels, and Lizards (using a SupersPort on the road), (S. Kristan), May 91, p.43

Printers, Dis an Dat About, (L. Vesei), Jul 91, p.19
RAM, Converting Z-386/16 from 1Mb to 4Mb, (H. Fale), Jun 91, p.35
Small Computer System Interface, Son of (SCSI-2), (N. Baker), Dec 91, p.37
Small Computer System Interface (SCSI), (N. Baker), Oct 91, p.9
Stay Cool, Dude (H/Z150 upgrades and additions), (J. Bazhaw), May 91, p.23
386SL Microprocessor, Introducing, (J. Ross), Sep 91, p.5
Two Drive B's, (G. Elwood), Feb 91, p.21
286LP 8 to 12 Megahertz Upgrade Made Easy, (G. Ludden), Apr 91, p.45
Upgrades, Heath/Zenith Useful PC, (H. Fale), Sep 91, p.21
Video Board, Install a High-Performance Z-649 into Your Computer, (D. Veit), Dec 91, p.5
Video Primer, PC, (D. Veit), Aug 91, p.5
Virus Protection and Recovery, (A. Neibauer), Sep 91, p.35
Z-386/16, Memory Transplant for (replacing a hard disk), (M. Wolfson), Jun 91, p.29
Zap: No More Data (effects of electrostatic discharge), (R. Miller), Apr 91, p.26

Reviews

Alps Allegro Printer Series: A User's View, (T. Bing), Jan 91, p.46
Desktop Publishing, Budget, Is Here (Key Publisher and HP DeskJet 500), (P. Swayne), Dec 91, p.31
Desktop Publishing: Improving Your Image (hand scanners and digitizers), (M. Haverstock), Jan 91, p.5
Fear of Frying (camping with a MinisPort), (N. Baker), Jul 91, p.29
Hard Drive, External Portable, for your Laptop or PC-Compatible (ex-PORT hard drive by Lightek), (H. Fale), Oct 91, p.25
Hewlett-Packard LaserJet IIP, (W. Wills), Oct 91, p.35
MastersPort 386SL, (E. Hicks), Jun 91, p.5
Motherboard, Western Imaging's Z-33 "ZUNI" Motherboard, (J. Fonte), Oct 91, p.5
Planes, Hotels, and Lizards (using a SupersPort on the road), (S. Kristan), May 91, p.43
Video Board, Install a High-Performance Z-649 into Your Computer, (D. Veit), Dec 91, p.5
Z-486/25E, An Introduction to, (D. Sadler), Jul 91, p.17

SOFTWARE

How-to-Use Applications and Operating Systems

Batch Files: What, How, Why, Who, and When, (J. Day), Sep 91, p.9
By Your Command, Chapter 1 (PCWatchWord macros), (R. O'Connor), Apr 91, p.21
CALENDAR.CAL (using a spreadsheet as a calendar), (S. Vagts), Feb 91, p.5
Database Publishing with dBASE IV and a LaserJet Printer, (A. Neibauer), Mar 91, p.17
dBASE III (D, Cool)
  Part 8 (custom reports), Feb 91, p.25
  Part 10 (getting the bugs out), Feb 91, p.29
  Part 11 (advanced techniques), Mar 91, p.29
  Part 12 (LOAD and CALL commands), Apr 91, p.37
DOS, Breaking the Churlish Bounds of (D. Lind)
  Part 1 (parent and child processes), Mar 91, p.35
  Part 2 (overlays), Apr 91, p.13
  Part 3 (expanded memory), May 91, p.13
  Part 4 (extended memory), Jun 91, p.13
Enable Revisited (G. Elwood)
  Part 1 (starting up), May 91, p.35
  Part 2 (word processor), Jun 91, p.37
  Part 3 (spreadsheet), Jul 91, p.9
  Part 4 (basic graphics), Feb 91, p.41
  Part 5 (advanced graphics), Mar 91, p.21
  Part 6 (database—a beginning), Aug 91, p.17
Fonts on the Fly, (A. Neibauer), Jul 91, p.41
Hard Disk Survival Kit, (H. Fale), Sep 91, p.18
Hard Drive Backup, Minimum Effort, (R. Siebers), Mar 91, p.47
HEPCAT patch, (P. Swayne), Dec 91, p.25
Instant Recall (1.2), Getting Started with, (A. Neibauer), Apr 91, p.29
Laser Printer Font Tools, Getting Started with, (A. Neibauer), Sep 91, p.39
LaserJet Soft Fonts, Adding to Wordperfect 5.X with Bitstream's Facelift, (W. Nabor), Feb 91, p.31
MENU.BAT, (S. Vagts), Apr 91, p.42
MS-DOS, Programming the Function Keys in, (P. Carboni), May 91, p.39
Personal Information Managers, Getting Started with (PrimeTime Personal and Current), (A. Neibauer), Jan 91, p.11
Quattro Pro, Getting to Know, (R. O'Connor), Jan 91, p.41
Virus Protection and Recovery, (A. Neibauer), Sep 91, p.35
Wordperfect 5.0 and Its Wonders (S. Bracket)
  Part 3 (styles), Mar 91, p.7
  Part 4 (columns), May 91, p.27
  Part 5 (tabs), Aug 91, p.29
  Part 6 (headers and footers), Oct 91, p.33
Wordperfect, Shrinking V5.1 to Fit on a MinisPort ZL-2, (J. Klinke), Oct 91, p.43
Writing, First Aid for, (M. Haverstock), Nov 91, p.33

Programming

Assembly Language (P. Swayne)
  Part 9 (inputting from the keyboard), Jan 91, p.29
  Part 10 (disk I/O), Aug 91, p.11
Batch Files: What, How, Why, Who, and When, (J. Day), Sep 91, p.9
Binary Conversion by Division: Why It Works, (G. Hoellerich), Dec 91, p.21
By Your Command, Chapter 1 (PCWatchWord macros), (R. O'Connor), Apr 91, p.21
C++, Introduction to (L. Wilson)
  Second Installment (output, variables, loops, true and false), Jan 91, p.25
  Third Installment (tools, operators, loops), Feb 91, p.45
  Fourth Installment (OOP, functions), Mar 91, p.25
  Fifth Installment (maintaining programs, more functions, directives), Apr 91, p.33
  Sixth Installment (loops, control statements, conditional operators), May 91, p.29
  Seventh Installment (arrays, strings), Jun 91, p.10
  Eighth Installment (pointers), Jul 91, p.33
  Ninth Installment (more on pointers), Aug 91, p.45
  Tenth Installment (structures, classes, constructors), Sep 91, p.31
  Eleventh Installment (designing and implementing an assembler), Dec 91, p.9
C, Learning by Computer (T. Bing)
  Part 1 (Master C), Jul 91, p.37
  Part 2 (Learn C Now), Aug 91, p.26
C: Bonehead Plays In (pointers for programmers), (T. Bing), Nov 91, p.45
Do You Want to Program (G. Hoellerich)
  Part 1 (BASIC), Jan 91, p.17
  Part 2 (BASIC syntax), May 91, p.7
  Part 3 (data files, BASIC compilers), Jun 91, p.33
DOS, Breaking the Churlish Bounds of (D. Lind)
  Part 1 (parent and child processes), Mar 91, p.35
  Part 2 (overlays), Apr 91, p.13
  Part 3 (expanded memory), May 91, p.13
  Part 4 (extended memory), Jun 91, p.13
FORTRAN, Harnessing (I/O routines), (D. Shoemaker), Jun 91, p.27

- Getting the Most from Your Computer (Pascal) (J. Lewis)
  - Part 1 (basics), Jun 91, p.46
  - Part 2 (programming a window, calculator), Jul 91, p.25
  - Part 3 (constructing a unit), Aug 91, p.41 (see Nov '91, p.12, for Listing 2)
  - Part 4 (calendar utility), Sep 91, p.26
  - Part 5 (Turbo Debugger), Oct 91, p.38
  - Part 6 (file-viewing utility), Nov 91, p.13
  - Part 7 (data base programs), Dec 91, p.13
- HEPCAT patch, (P. Swayne), Dec 91, p.25
- I Can Do It (D. Warnick)
  - Part 1 (programming basics), Nov 91, p.19
  - Part 2 (do math, output data, loop, structured programming), Dec 91, p.18
- Keyboard, Using a Full Size, on a Laptop Computer (handy utility), (P, Swayne), Nov 91, p.21
- Melodies for Your Programs, (R. Moon), Aug 91, p.36
- MENU.BAT, (S. Vagts), Apr 91, p.42
- MS-DOS, Programming the Function Keys in, (P. Carboni), May 91, p.39
- Programmer's Craft: Tools of the Trade, (T. Bing), Apr 91, p.10
- Windows 3, Developing Applications Programs for, (H. Ogg), Dec 91, p.26

Reviews

- C, Learning by Computer (T. Bing)
  - Part 1 (Master C), Jul 91, p.37
  - Part 2 (Learn C Now), Aug 91, p.26
- Central Point Backup, A Review of, (R. O'Connor), Jul 91, p.13
- Desktop Publishing, Budget, Is Here (Key Publisher and HP DeskJet 500), (P. Swayne), Dec 91, p.31
- Electronic Clavier (New ZUG Product), Feb 91, p.16
- Frugal PC (Let Shareware Fill Your Hard Disk Drive), (H. Ogg), Jun 91, p.42
- Harvard GeoGraphics, (G. Elwood), Nov 91, p.7
- Harvard Graphics 3.0, (G. Elwood), Nov 91, p.25
- Harvard Graphics, New and Improved, (E. Zimmerman, Jr.), Apr 91, p.7
- LANtastic, (J. Babcock), Dec 91, p.34
- Laser Printer Font Tools, Getting Started with, (A. Neibauer), Sep 91, p.39
- LetterPerfect: Less Filling, Writes Great, (M. Haverstock), Jun 91, p.7
- Quattro Pro, Getting to Know, (R. O'Connor), Jan 91, p.41
- Rightwriter 4.0, Polishing Your Writing Style with, (T. Bing), Nov 91, p.37
- Utility (PC Tools, DR, RN, DIRMATCH, Snipper, 4DOS, QEMM-386), (S. Shapiro), Nov 91, p.41
- What Is It, REALLY (E. Mosher)
  - Part 1 (ZUG software for PC, H/Z100), Jun 91, p.24
  - Part 2 (ZUG software for H/Z100), Jul 91, p.22
- What Price a Man's Toys? (A Look at Windows 3.0, QEMM, and DESQview), (S. Shapiro), Feb 91, p.14
- Windows 3, Developing Applications Programs for, (H. Ogg), Dec 91, p.26
- Word Processor, Search for the Perfect (TEXTRA), (L. Vesei), May 91, p.33
- WordPerfect Version 5.1, Impressive New Features, (E. Wiggins), Mar 91, p.5
- WordStar Past and Present, (W. Pitts), Dec 91, p.44
- Writing, First Aid for, (M. Haverstock), Nov 91, p.33

MISCELLANEOUS

- Index to REMark: 1990, (J. Axelson), May 91, p.46
- Subscribers, Information for Concerned, (L. Cobb), Feb 91, p.8
- Subscribers, Information for New and Confused, (L. Cobb), Jan 91, p.28
- What Is It, REALLY (E. Mosher)
  - Part 1 (ZUG software for PC, H/Z100), Jun 91, p.24
  - Part 2 (ZUG software for H/Z100), Jul 91, p.22
- Zlink-COM1 BBS (general information), Mar 91, p.4

I entered the information in this index into PC-File Version 5.0, then imported it into my word processor and created the text file you see here.

If you prefer an electronic index, you can download REMARK91.EXE from the ZUG BBS. REMARK91.EXE is a self-extracting file that contains the above information in the form of a PC-File-compatible database. Using REMARK91 and PC-File, you can quickly search for occurrences of any topic or combination of topics you wish.

Note: To use the disk-based index, you must have PC-File Version 5.0 or compatible database software. ✻

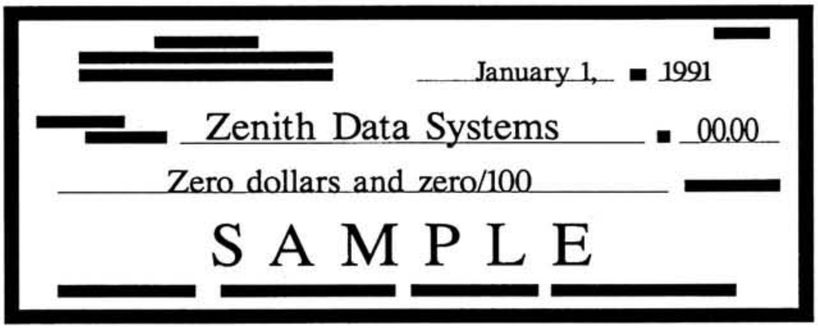
Continued from Page 28

you use MS-DOS 5 or DR DOS 6 you would use FDISK and have only your primary partition bootable.

If you use PART, you also need to FORMAT /S your partition(s). FDISK does a FORMAT automatically. You can now restore your software.

If you can't PART or FDISK you need to PREP. See your ZDS MS-DOS book for details (sorry, there's no DR-DOS command for this, and it only works with ZDS MS-DOS). Once PREP is complete you can PART (FDISK) it into two 60M partitions as above. ✻

All Checks must be made out to Zenith Data Systems



January 1, ■ 1991

Zenith Data Systems ■ 00.00

Zero dollars and zero/100

SAMPLE

Classified Ads

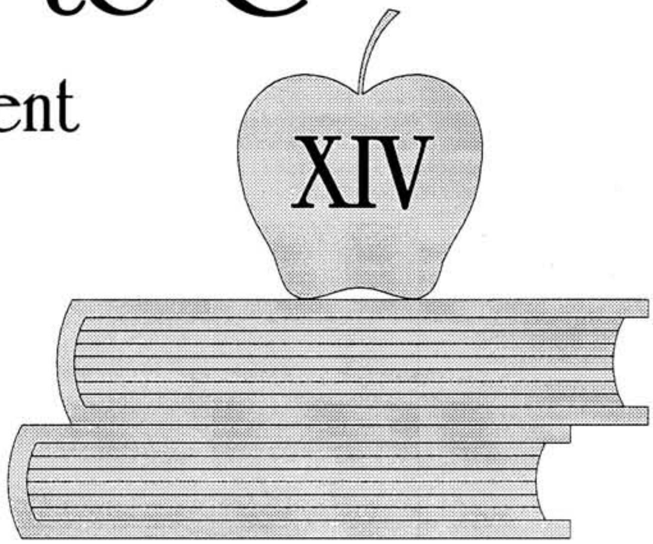
ZENITH DATA SYSTEMS 386/16MHz, 4 meg. memory, coprocessor, 1.2/1.44 FFD's, 80 meg. HDD, Z-549 video, cache, keyboard, manuals. $1100

Z-515 4 Meg. memory, $400

W. Serp, 4027 Harvey Avenue, Western Springs, IL 60558. (708) 246-7856

Introduction to C++

Fourteenth Installment



Lynwood H. Wilson
2160 James Canyon
Boulder, CO 80302

A String Class

This month I am going to present a complete class which will serve as an example of the further extension of the C++ language, and which you may find useful in your programs. It is a String class, to make the use of character strings easier and clearer. (I have adopted the convention that class names begin with capitol letters. It's nice to be able to tell immediately what you are dealing with. In this article a string is an array of chars with a 0 at the end, just like always, and a String is an instance of the new class we are writing.)

The immediate inspiration for the String class was a desire to be able to compare two strings by writing:

```
    if(str1 == str2)
```

instead of

```
    if(!strcmp(str1, str2))
```

If the truth be known, it was brought on by omitting the exclamation point in the above statement. It wasn't the first time I'd forgotten that particular exclamation point. For some reason I've had a problem remembering that strcmp() returns false (0) when the strings are the same. Now I won't have to.

In spite of what I have said in previous installments about the importance of specifications and the design process (or maybe because of it) I really enjoy a small problem like this one which I can just hack out on the screen. I did give some prior thought to goals and methods, and decided to write as simple and obvious a class as I could. After all, the reason was to make string handling simpler and more straightforward.

There are plenty of design choices here. I chose to stick to (what I see as) the original C philosophy of a fast, terse and simple language.

Data

First I thought of ways to represent the data, the string, within the class. I considered for almost three seconds before I decided to represent it as a character pointer which points to a null terminated string of characters. I've been representing strings that way for years and I can't see any reason to change.

In the class declaration this is a character pointer, rather than a character array. That way we can allocate memory to hold the actual string at run time when we know how long the string is. This will give us a substantial memory saving over defining character arrays big enough to hold the largest possible string.

Note that if the contents of the String are changed the String class may have to allocate a larger block of memory to hold it. For that matter, if the new String is shorter than the old, the String class will allocate a smaller amount of memory. In either case it will deallocate the old memory so it can be reused. In my experience most strings do not change size after they are filled, but if you write a program in which they do you could speed it up at a very slight cost in memory by modifying the code below so that it does not adjust the size of the memory block for a String when it gets smaller.

The second question is whether to include any more data. I considered adding an integer variable to hold the length of the string, but decided against it. It's a good idea not to maintain the same data in two different forms or places if you can avoid it. If we don't store a number for the length of the string but instead count the characters in the string each time we need to know the length, we will waste a bit of time but we will eliminate any chance of the number being wrong. And we can always change our minds and add the length variable later.

Creators

Now that we have the data settled, we should next consider creator functions. The first one I wrote was the creator without arguments, which creates an empty String. This supports the old C habit of defining variables at the head of functions and initializing them later. In C++ we tend to define variables later when we can initialize them at the same time, but we shouldn't try to enforce it by not providing a constructor with no arguments. There are other reasons, too. One is that a constructor with no arguments allows you to define an array of Strings without initializing all of them right then and there.

So what does this creator function need to do? The only thing I could think of is to create a null string and set the char pointer to point to it. If there is no string the pointer should not be left pointing at some random place in memory. Also, we want to be compatible with old code as much as is convenient. It is a common practice in C and C++ to put a 0 in the first byte of a character array if you need to show that it has no string in it, or perhaps that it has not been initialized. Then you can see if there is a string with a simple

```
    if(s[0])
```

which is true unless the first char of the string s is 0.

Here is the declaration of the constructor which takes no arguments.

```
String(void) {str = new char; *str = 0;}
```

Note that the declaration includes the body of the function as well. This is good practice if the function is small and simple.

It is equivalent to an inline declaration.

Next we need a constructor that takes a pointer to a string as an argument. This will allow us to use the same initialization syntax that we use with strings. This constructor will find the length of the string, allocate memory to hold it, make the String char pointer point to the newly allocated memory, and then copy the string to the memory pointed to by the String char pointer. Here is the declaration.

```
String(const char *s);
```

And here is the function definition.

```
String::String(const char *s)
{
   str = new char[strlen(s) + 1];
   assert(str);
   strcpy(str, s);
}
```

Note that if the length of the string is 0 no memory is allocated and the pointer in the String is set to NULL. Note also that the source string is declared as a const. This is usual in the declaration of pointer arguments to things which are not changed in the function. There are two reasons. If you accidentally change the thing pointed to by such an argument in the function, you would get an error message from the compiler, and you can call such a function with an actual argument which is a constant.

The assert() function will send a message, including the file and line number, to stderr and abort the program if its argument is false. In this function that will occur if str is 0, which would indicate that the memory allocation failed. This is primarily intended for your use during debugging. If there is a chance that assert will be called during normal program operation, you should provide a more informative message for the user or handle the problem yourself by deallocating some other memory.

The last constructor we will need takes a String as an argument, so we can initialize a new String with an existing String. Here is its declaration:

```
      String(const String &s);
```

and its definition.

```
String::String(const String &s)
{
   str = new char[strlen(s.str) + 1];
   assert(str);
   strcpy(str, s.str);
}
```

This is very similar to the constructor which takes a string. When copying the data, this one copies from s.str, the pointer in the source String. Note that this function is called as a member of a new String which is taking a copy of an existing String, and the constructor has access to the private data in both the old and the new Strings because it is a String member function.

Note also that the formal argument, which is filled by the source String passed to the constructor, is declared in the function as a reference. This has no effect on the logic of the function, it would work the same if the formal argument in the function was a simple String, but in that case the program would have to create a new String object in the function to receive the value of the actual argument the String passed to it. In this case, with the formal argument declared to be a reference, no copy will be created. Once again, the incoming argument is declared to be a const.

I almost put in another constructor which would take an integer as argument and create an empty String with enough memory to store a string of that length. If you like the idea, it would make a good simple exercise. If you don't, let's go on to the destructor.

Destructor

We need an explicit destructor because the default destructor will not deallocate the memory which was allocated in the constructors, but that doesn't amount to much work. Here is the declaration which also contains the function body.

```
      ~String(void) {delete str;}
```

Member Functions

The Assignment Operator

The next thing to consider is what member functions to include. This includes the question of what to do about the other things someone might want to do with a string which we don't include as member functions. I decided to try to ensure that everything you could do with an old style string you would also be able to do with a String. This simplifies the question of member functions fıom "What will we want to do with strings?" to "What would we like to be able to do a different way?". We do not need to provide a complete suite of string functions, only the new and different ones. We can fall back on the existing code and habits for everything else. I think this is very desirable. The great success of C++ is partly because of the smooth and relatively easy transition from C. Let's not tamper with success.

So what do we want to be able to do differently? Let's overload the assignment operator, so we can copy the value of one String to another with a simple equal sign instead of using strcpy(). Have I mentioned previously that operators (like =) can be overloaded just like functions? This is one of many features of C++ which allow you to make the data types that you create as natural to the language and easy to use in programs as the types (like int) that are part of the language.

Here's the declaration.

```
String& operator=(const String &s);
```

The String& on the left indicates that the expression evaluates to a reference to a String. Remember that an ordinary assignment expression evaluates to the value (and datatype) being assigned. That's why we can chain them. The statement x = y = 7; gives the value 7 to x because the expression y = 7 evaluates to 7. We want to preserve this in our overload of the operator.

The word operator shows the compiler that the equal sign which follows is an operator being overloaded. And the (const String &s) is the declaration of the formal argument which will be given the value of the expression on the right side of the equal sign. Once again it is declared to be a reference so no new copy of the String will be created.

In a sense it's like a function named operator= with the word operator to indicate that we are overloading a pre-existing operator. Like the existing operator our new one will take two arguments, one on each side of the =. Since this is a member function in the class String it expects the expression on the left side of the operator to be a String.

Here is the code for the assignment.

```
String& String::operator=
                         (const String &s)
{
   int len = strlen(s.str);
   if(len != strlen(str)) {
      delete str;
      str = new char[len + 1];
      assert(str != 0);
   }
   strcpy(str, s.str);
return(*this);
}
```

The thing on the left side of the equal sign is assumed to be a String, because the function is a member of the String class and the thing on the left side of the equal sign is the object of type String that the function belongs to. Otherwise this particular overloaded version of the assignment operator will not be used.

In the first line of the function body len is assigned the value returned by strlen(s.str). This function call sends the char pointer from the String s to the function strlen(). The function we are writing, the overloaded operatoı=, has access to the private char pointer inside s because our function is a member of the String class.

The second time we call strlen() we send it str, which is the char pointer from the String on the left side of the equal sign, the current String object.

Note that the function returns *this. The pointer called this is a pointer to the particular object of type String under which the function is invoked. Such a pointer is created automatically for each object which is created, and can be used by any code which is a member of the object. For example, in this code fragment,

```
      String greeting = "hi", extra_greeting;
      extra_greeting = greeting;
```

the assignment operator is a member of the particular String object called extra_greeting because it is on the left side of the equal sign. Therefore, the pointer this inside the assignment code above points to the String extra_greeting. Therefore, when we return *this we are returning the String extra_greeting. But since we declared the return value of the operator to be of type &String, a reference to a String, the program need not waste time creating a new copy of extra_greeting but merely returns a reference to it.

Now that we can assign a String to another String, what about assigning an old style string to a String? Certainly a useful thing to be able to do. It turns out that we already have that capability. The constructor which makes a String from an ordinary string can be used as a type conversion. As always, when the compiler sees an operator or function being used with datatypes it doesn't recognize as legal arguments for the operator or function, it tries to convert the data into legal types. When the compiler sees us assigning an old style string to a String, it will use the constructor and create a temporary String from the string and use that to make the assignment. Then the temporary is destroyed.

This is a good and useful trick to know about, but it has the disadvantage that it requires the time and memory to create the temporary String object. To save that time and memory we will provide an explicit version of the assignment operator to assign a string to a String. Here is its declaration.

```
     String& operator=(const char *s);
```

And here is the code.

```
String& String::operator=(const char *s)
{
   int len = strlen(s);
   if(len != strlen(str)) {
      delete str;
      str = new char[len + 1];
      assert(str != 0);
   }
   strcpy(str, s.str);
   return(*this);
}
```

It's no surprise that this is very similar to the previous code.

The Append Operators

I have overloaded the += operator to append one String to the end of another, and I have also overloaded the + operator to return a String consisting of one String appended to another, without changing either of the original strings. I am not certain that these deserve to exist, in the light of what I said earlier about the clean simple language, but once I got started I had trouble stopping. In any case the code is similar to that above so I will not discuss them. See the complete code which accompanies this article.

Type Conversion

Earlier, I promised to make a String behave like a string in all cases not covered by the member functions. This can be accomplished by writing a data type conversion which will allow a String to behave just like a character pointer, which is the normal representation of a string. Here's how.

```
     operator char *() {return(str);}
```

If this syntax doesn't seem very obvious, try thinking of it as an overload of the type cast operator.

The result is to define an automatic data type conversion which will come into play if a String is used in a situation which calls for a char pointer. In this situation the char pointer from the String object is made available to the code which was expecting the char pointer. For example:

```
     String greeting("Hi there");
     puts(greeting);
```

This fragment will print the words "Hi there" to the screen. The function puts() expects a char pointer and so the String greeting obediently turns into one. Here is a further example:

```
     greeting[1] = 'o';
     puts(greeting);
```

This bit will print "Ho there" to the screen. In the same fashion, anything you normally do with a string will work with a String. And the same things that didn't work with strings will also fail with Strings. To continue with the example:

```
char *new_greeting = "Hello big boy";
for(int i = 0; new_greeting[i]; i++)
          greeting[i] = new_greeting[i];
```

This line will write outside the memory allocated for the String greeting, just as it would in the case of an ordinary string.

We escape a lot of complication by allowing the conversion of a String to a char * to the string itself. This allows all of the language operators and library functions and any functions in the program to deal with Strings without our making any particular provision for them. This is a good deal. The cost is that any code which has access to a String through this conversion can put junk in it, write a non-string (with no null to terminate it), or write outside the memory allocated for the string. These dangers seems less dire when we consider that we have been getting along fine with all of them (and more) in C for years.

```
friend int operator==(const String &s1, const String &s2);
friend int operator==(const String &s1, const char *s2);
friend int operator==(const char *s1, const String &s2);
```

Figure 1

```
     int operator==(const String &s1, const String &s2)
     {
        for(int i = 0; s1.str[i] == s2.str[i]; i++)
           if(s1.str[i] == '\0')
              return(1);
           return(0);
     }
```

Figure 2

The Equality Operator

The last member function on the list for the String class is the test for equality, which is actually the one which started it all. Of course we can write

```
     String s1, s2;
     ...
     if(!strcmp(s1, s2))
```

as we always have and the automatic conversion of a String to a char pointer will make it work, but we want to be able to use the double equals instead.

We know how to overload operators, but this is a bit different because we would like to be able to use the double equals to compare two Strings or a String and a string, no matter which side of the equal sign the string is on. When we overload an operator as a member function and the operator takes two arguments (like the assignment operator above) the argument on the left side must be an instance of the class. This would prevent us from overloading the double equals to handle a string on the left and a String on the right. The solution is to make these functions friends rather that members. As friends of the String class they have access to the private data in a String object but they do not share the restriction of needing a String type on the left side. I could have written the two versions of the equality function which do have the String on the left as members and the other as a friend, but I made all three friends for regularity. Figure 1 shows the declarations. And Figure 2 shows the code for one of them.

This operator returns 1 if the two Strings are identical and 0 if they are not. If you need to know which is greater for determining alphabetical order you can fall back on strcmp().

The other two functions are very similar to this one. Note that the function does not begin with String:: because it is not a member of the String class.

That's the end of the String class in its current form. I will continue to use it in my programs and modify it as experience indicates. If I come up with anything interesting I will pass it along in a future article.

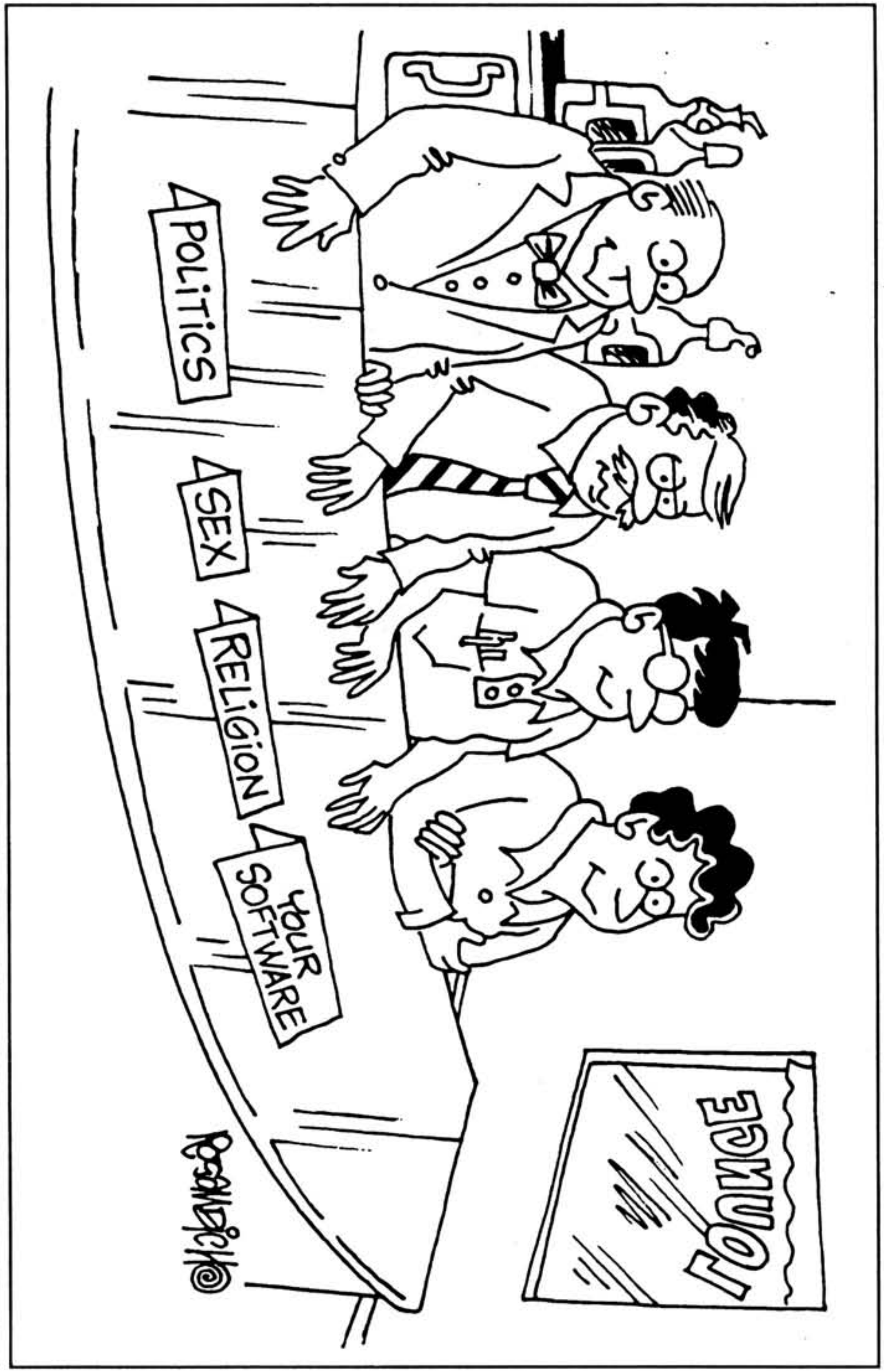
There was a good article in Dr. Dobbs Journal for October, 1991 about a proposed standard String class. You can get the code from the DDJ forum on Compuserve. Their String class is much bigger and more complex than ours, and makes an interesting study of another way to do things.

The more C++ books I read the more I appreciate the C++ Primer by Stanley B. Lippman. I use it a lot, both to study and as a reference.

```
STR.HPP ***************************************************************

/* STR.HPP                 Lynwood H. Wilson
   11-15-91
*/

#ifndef STR
#define STR


class String {
   char *str;
public:
   String(void) {str = new char; *str = 0;} // make an empty one, null string
   String(const String &s);         // make one from a String
   String(const char *s); // make one from a string
   String& operator=(const String &s);      // assignment operator from a String
   String& operator=(const char *s);        // assignment operator from a string
   String& operator+=(const String &s);     // append one to another
   String& operator+=(const char *s);       // append a string to one
   String operator+(const String &s)const;  // add one to another
   String operator+(const char *s)const;    // add one to a string
   operator char *() {return(str);}         // conversion of a String to a string
   ~String(void) {delete str;}      // say goodnight
   friend int operator==(const String &s1, const String &s2); // are two = ?
   friend int operator==(const String &s1, const char *s2);  // one = string?
   friend int operator==(const char *s1, const String &s2);  // string = one?
};


#endif


STR.CPP ***************************************************************

/* STR.CPP                        Lynwood H. Wilson
   11-21-91
*/

#define TEST                              // to compile test program, below

#include <assert.h>
#include <iostream.h>
#include <string.h>
#include <stdio.h>

#include "str.hpp"


String::String(const String &s)
{
   str = new char[strlen(s.str) + 1];
   assert(str);
   strcpy(str, s.str);
}


String::String(const char *s)
{
   str = new char[strlen(s) + 1];
   assert(str);
   strcpy(str, s);
}


String& String::operator=(const String &s)
{
   int len = strlen(s.str);
   if(len != strlen(str)) {
      delete str;
      str = new char[len + 1];
      assert(str != 0);
   }
   strcpy(str, s.str);
   return(*this);
}


String& String::operator=(const char *s)
{
   int len = strlen(s);
   if(len != strlen(str)) {
      delete str;
      str = new char[len + 1];
      assert(str != 0);
   }
   strcpy(str, s);
   return(*this);
}


String& String::operator+=(const String &s)
{
   char *p = new char[strlen(str) + strlen(s.str) + 1];
   assert(p != 0);
   strcpy(p, str);
   strcat(p, s.str);
   delete str;
   str = p;
   return(*this);
}


String& String::operator+=(const char *s)
{
   char *p = new char[strlen(str) + strlen(s) + 1];
   assert(p != 0);
   strcpy(p, str);
   strcat(p, s);
   delete str;
   str = p;
   return(*this);
}


String String::operator+(const String &s)const
{
   String r(*this);

   r += s;
   return(r);
}


String String::operator+(const char *s)const
{
   String r(*this);

   r += s;
   return(r);
}
```

```
int operator==(const String &s1, const String &s2)
{
   for(int i = 0; s1.str[i] == s2.str[i]; i++)
      if(s1.str[i] == '\0')
         return(1);
      return(0);
}


int operator==(const String &s1, const char *s2)
{
   for(int i = 0; s1.str[i] == s2[i]; i++)
      if(s1.str[i] == '\0')
         return(1);
      return(0);
}


int operator==(const char *s1, const String &s2)
{
   for(int i = 0; s1[i] == s2.str[i]; i++)
      if(s1[i] == '\0')
         return(1);
      return(0);
}


#ifdef TEST

main(void)
{
   char *s;
   String s1, s2, s3(" 'n dogs");
   String s4(s3);

   if(!s1[0])
      cout << "first !s1" << endl;

   s2 = "cats";
   s1 = s2;

   if(s1[0])
      cout << "s1 again" << endl;

   cout << "s3 length = " << strlen(s3) << endl;
   cout << s1 << endl;
   s2 += s4;
   cout << s2 << endl;
   s2[3] = 'z';
   cout << s2 << endl;
   cout << "s2 == s3 is " << int(s2 == s4) << endl;
   cout << "s4 == s3 is " << int(s4 == s3) << endl;
   cout << "s1 == cats is " << int(s1 == "cats") << endl;
   cout << "s1 == dogs is " << int(s1 == "dogs") << endl;
   cout << "s2 == s2 is " << int(s2 == s2) << endl;

   s1 = "";
   if(s1)
   cout << "s1 again" << endl;

   s1 = s2 + " 'n cats";

   s1 = s1 + s4;
   cout << s1;
}

#endif
```

✱

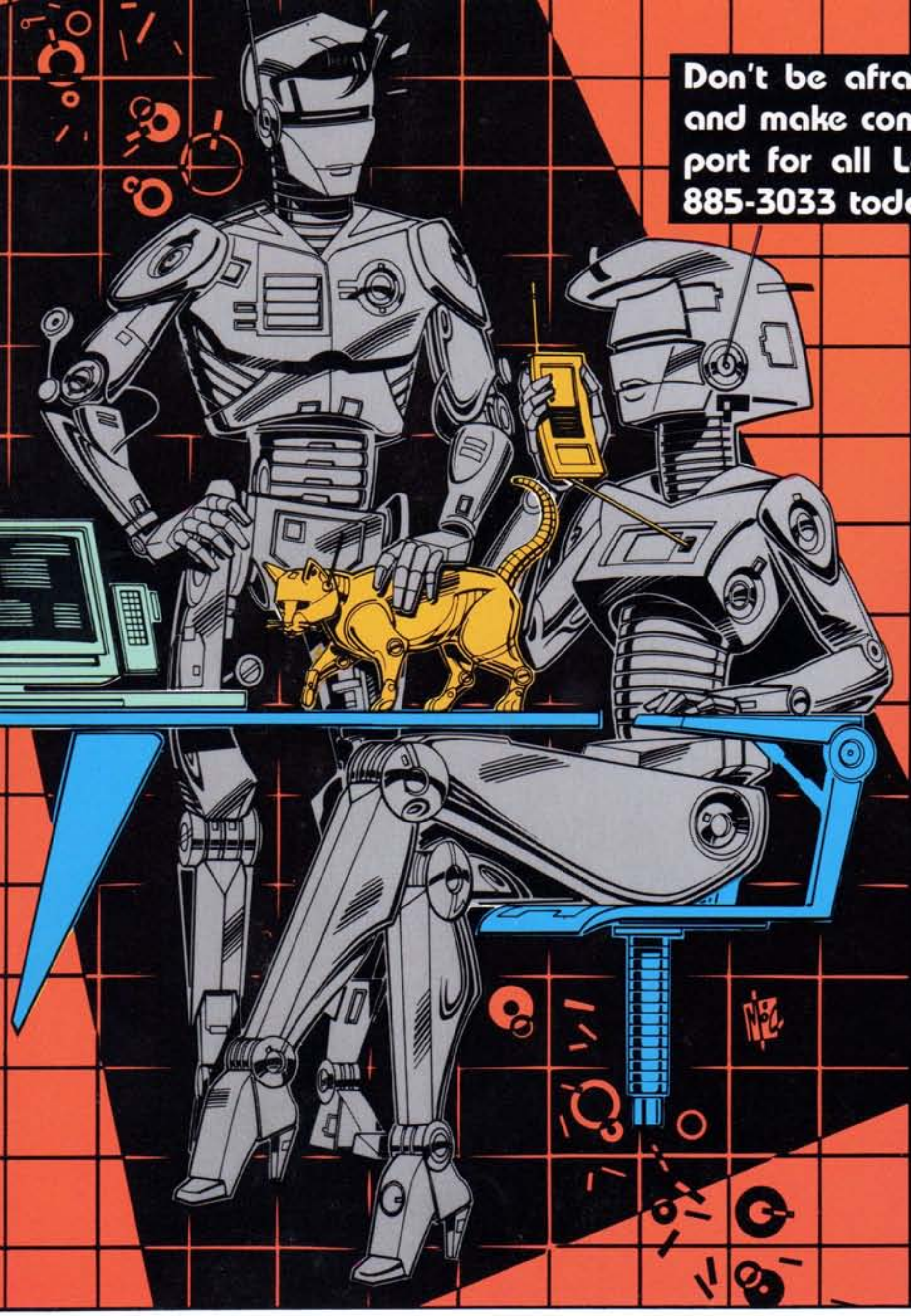


HADES-II

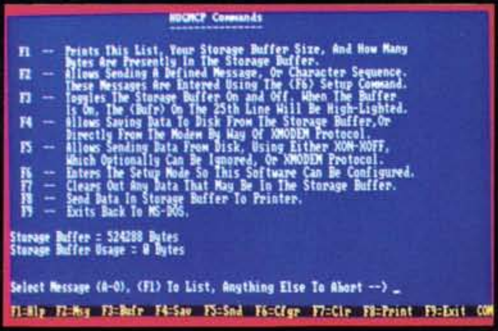
It's HOTTER than ever! Jam-packed with new features, HADES II still remains the easiest-to-use disk editor ever! Just look at some of the features:

- 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

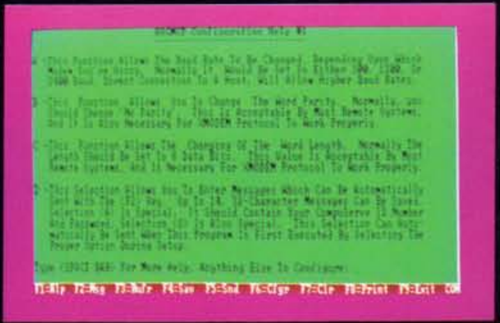
HADES II is still only $40, and original HADES owners can upgrade their distribution disk for only $15. Call HUG today at: (616) 982-3463.



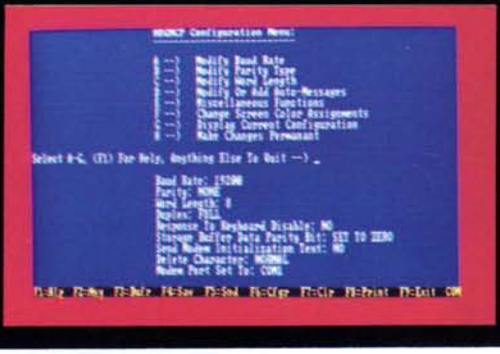
Don't be afraid to communicate! Get HUGMCP and make contact the easy way. Now with support for all Laptops, order HUG Part number 885-3033 today.



```
                    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 15, 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
```

ZENITH
data systems
Groupe Bull

$2.50
P/N 885-2147