

# REMark

February 1991



The Official Zenith Data Systems Computer Users Magazine

## Not Just a Plain Brown Wrapper!

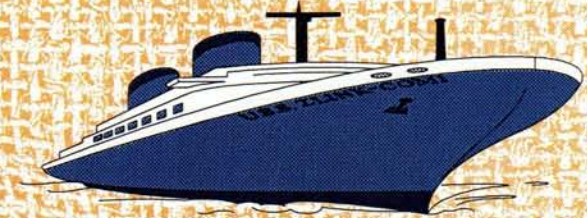
Concerned Subscribers

Page 8



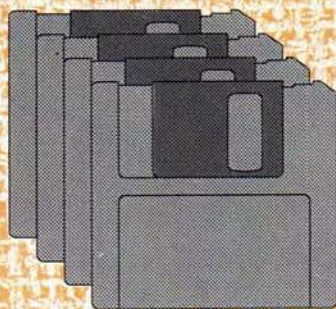
Port-O-Call: COM1

Page 9



New ZUG Software

Page 16





A decorative border of stylized orange and yellow flames surrounds the central text.

# 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.

# REMark

February 1991



The Official Zenith Data Systems Computer Users Magazine

## Resources

Software Price List	2
New Products	16
Classified Ads	28

Reader Service No.		Page No.
250	B.U.D. Unlimited	13
104	F.B.E. Research Co., Inc.	20
136	Lindley Systems	44
117	Payload	4
153	Surplus Trading Corp.	30
149	W.S. Electronics	10



## ACCOUNTING & TAX

Not sure if you need the expensive 'Chinese Flower 1-2-3', or 'Spanish Numeral Four' spreadsheet programs? Then find out for only \$20! "CheapCalc" will do double precision addition, subtraction, multiplication, division, power, SUM, and roots (using fractional powers). CheapCalc has many other functions too numerous to mention (just like the expensive spreads)! CheapCalc is available for all Heath/Zenith computers and operating systems. For more information, check out page 58 of the Software Catalog Update #1, or call HUG and order your copy today.

## PC Compatibles

Powering Up - Volume 2	17
<i>William M. Adney</i>	
dBASE III - Part 8	25
<i>D. R. Cool</i>	
dBASE III - Part 10	29
<i>D. R. Cool</i>	
Adding Laserjet Soft Fonts to WordPerfect 5.X With Bitstream's Facelift	31
<i>William C. Nabor</i>	

## H/Z-100 & PC Compatibles

On the Leading Edge	35
<i>William M. Adney</i>	
ENABLE Revisited - Part 4	41
<i>George P. Elwood</i>	

## General

CALENDAR.CAL	5
<i>Steven Vagts</i>	
Concerned Subscribers	8
<i>Lisa Cobb</i>	
Port-O-Call: COM1	9
<i>Laura White</i>	
MODEM: A Data Transfer Solution - Part 2	11
<i>Robert C. Brenner</i>	
What Price a Man's Toys?	14
<i>Sanford Shapiro</i>	
Two Drive "B's"	21
<i>George P. Elwood</i>	
Introduction to C++ - Third Installment	45
<i>Lynwood H. Wilson</i>	



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

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

\* U.S. Funds

Limited back issues are available at \$2.50, plus 10% shipping and handling - minimum \$1.00 charge. Check 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.

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

Copyright (c) 1991, Zenith Users' Group

# Software

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM		DESCRIPTION	PRICE
		H8 - H/Z-89/90	H8 - H/Z-89/90 - H/Z-100 (Not PC)		
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 PROC	35.00
SCICALC	885-8027	HDOS		UTILITY	20.00
SMALL BUSINESS PACKAGE	885-1071-[37]	HDOS		BUSINESS	75.00
SMALL-C COMPILER	885-1134	HDOS		LANGUAGE	30.00
SOFT SECTOR SUPPORT 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
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



# 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
<b>H/Z-100 and PC Compatibles</b>					
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
<b>PC Compatibles</b>					
CARDCAT	885-6006	MSDOS		CAT SYSTEM	20.00
CHEAPCALC	885-6004	MSDOS		SPREADSHEET	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

This Software Price List contains all products available for sale. For a detailed abstract of these products, refer to the Software Catalog, Software Catalog Update #1, or previous issues of REMark.

### Now Available!

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

### LAPTOP OWNERS . . . don't feel left

out! All of ZUG's MS-DOS software is available on 3-1/2" micro-floppies too! When ordering, just add a "-80" to the 7-digit ZUG part number. For the standard 5-1/4" floppy, just add a "-37".

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

### ORDERING INFORMATION

For VISA, MasterCard, and American Express phone orders, telephone the Zenith Users' Group directly at (616) 982-3463. Have the part number(s), description(s), and quantity ready for quick processing. By mail, send your order, plus 10% postage and handling (\$1.00 minimum charge, up to a maximum of \$5.00) to: Zenith Users' Group, P.O. Box 217, Benton Harbor, MI 49023-0217. VISA, MasterCard and American Express require minimum \$10.00 order. No C.O.D.s accepted.

Questions regarding your subscription? Call Lisa Cobb at (616) 982-3463.



# SEAGATE ST-251-1 42 MEG HARD DRIVES ON SALE

## Seagate - HARD DRIVES

MODEL	CAPACITY/FORMAT/SPEED/SIZE	DRIVE ONLY	XT KIT
* ST-125	21 MEG / MFM / 40 MS / 3.5"	\$229.00	\$277.00
* ST-138	32 MEG / MFM / 40 MS / 3.5"	\$277.00	\$325.00
* ST-138-1	32 MEG / MFM / 28 MS / 3.5"	\$307.00	\$355.00
* ST-151	42 MEG / MFM / 24 MS / 3.5"	\$353.00	\$401.00
* ST-157R	49 MEG / RLL / 40 MS / 3.5"	\$286.00	\$339.00
* ST-225	21 MEG / MFM / 65 MS / 5.25"	\$199.00	\$247.00
* ST-250R	42 MEG / RLL / 70 MS / 5.25"	<b>\$248.00</b>	<b>\$288.00</b>
* ST-251-1	42 MEG / MFM / 28 MS / 5.25"	<b>\$289.00</b>	<b>\$337.00</b>
* ST-4096	80 MEG / MFM / 28 MS / 5.25" FH	\$582.00	\$631.00
* ST-238R	32 MEG / RLL / 65 MS / 5.25"	\$218.00	\$271.00
* ST-277R-1	65 MEG / RLL / 28 MS / 5.25"	\$348.00	\$401.00
* ST4144R	122 MEG / RLL / 28 MS / 5.25" FH	\$623.00	\$671.00

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

### \*\*\* ZENITH PC COMPUTER UPGRADES \*\*\*

#### SmartWatch from FBE RESEARCH

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

#### Z-150/160 MEMORY UPGRADE

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

#### Z-150 SERIES HARD DISK DRIVE KIT

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

* ST-125/Z150 Kit	21 Meg, 40 MS,	\$281.00
* ST-138/Z150 Kit	32 Meg, 40 MS,	\$329.00
* ST-251/Z150 Kit	42 Meg, 28 MS,	<b>\$341.00 SALE PRICED</b>

#### Z-148 HARD DISK DRIVE KIT

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

* ST-125/Z148 Kit	21 Meg, 40 MS,	\$352.00
* ST-138/Z148 Kit	32 Meg, 40 MS,	\$399.00
* ST-251/Z148 Kit	42 Meg, 28 MS,	<b>\$408.00 SALE PRICED</b>

#### Z-148 EXPANSION CARD

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

#### Z-150 VIDEO ELIMINATOR

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

#### 2400 BAUD MODEMS

⇒ Fully Hayes compatible 2400/1200/300 Baud with Software. Internal \$84.00 External Model \$128.00 Cable for External Modem \$8.50

⇒ MICROSOFT WINDOWS SOFTWARE version 1.04 for PC (not Z-100)  
 ⇒ FREE with any order from this ad. Just ask for it and add \$5.00 for shipping and handling. Includes 5 disks and 300+ page manual. Offer good for limited time.

#### WESTERN DIGITAL HARD DISK CONTROLLER CARDS

⇒ WD XT GEN2+ 8 BIT, MFM, DUAL HARD DRIVES, XT COMPUTERS	\$48.00
⇒ WD1004-27X 8 BIT, RLL, DUAL HARD DRIVES, XT COMPUTERS	\$53.00
⇒ WD1006V-MM2 16 BIT, DUAL HARD, DUAL FLOPPY, 1:1 AT	\$99.00
⇒ WD1006V-SR2 16 BIT, RLL, DUAL HARD, DUAL FLOPPY, 1:1 AT	\$109.00
⇒ HARD DRIVE CABLE SETS, XT.. \$3.50, AT .. \$4.50, AT + Floppy ....	\$6.50

#### PRINTERS

⇒ STAR NX-1000 II 9 PIN, 180 CPS, 45 CPS NLQ,	\$179.00
⇒ SEIKOSHA SL-90 24 PIN, 240 CPS, 80 CPS NLQ, PAPER PARKING	\$314.00
⇒ PANASONIC 1180 9PIN, 192 CPS	\$204.00
⇒ PANASONIC 1124 24 PIN, 192 CPS	4335.00
⇒ PANASONIC 1624 24 PIN, 192 CPS, WIDE CARRAGE	\$422.00
⇒ PANASONIC 1695 9 PIN, 288 CPS, WIDE CARRAGE	\$458.00
⇒ PRINTER CABLE	\$12.00

#### FLOPPY DISK DRIVES

⇒ MITSUBISHI MF501	5.25" 48 TPI DS/DD 320K/360K	\$ 68.00
⇒ MITSUBISHI MF504	5.25" High Density 360K/1.2 MEG	\$ 81.00
⇒ MITSUBISHI M-353	3.5" in 5.25" frame 720K	\$ 84.00
⇒ MITSUBISHI M-355	3.5" in 5.25" frame 1.44 MEG	\$ 94.00
⇒ TOSHIBA ND352	3.5" with 5.25" frame 720K	\$ 74.00
⇒ TOSHIBA ND356	3.5" with 5.25" frame 1.44 MEG	\$ 79.00
⇒ M-355 and ND356 run on AT compatible or special controller only.		

#### PAYLOAD CUSTOM ASSEMBLED COMPUTERS

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

#### VIDEO MONITORS

⇒ ZCM-1492	ZENITH Color Flat Screen VGA	\$ 679.00
⇒ MA2565	SAMSUNG Amber TTL 720x350	\$ 89.00
⇒ CW4656	SAMSUNG Color RGB 640x200	\$ 223.00
⇒ CM4592	SAMSUNG Color EGA 640x350	\$ 339.00
⇒ CJ4681	SAMSUNG VGA 720x400	\$ 360.00
⇒ CVB4581	SAMSUNG Multi-sync VGA 1024x768	\$ 429.00
⇒ CM1440	SEIKO VGA 1024x768 .25 dot	\$ 569.00

#### VIDEO CARDS

⇒ BASIC VGA	PARADISE AUTOSWITCH 640x480	\$ 99.00
⇒ VGAPLUS	PARADISE AUTOSWITCH 800x600	\$ 146.00
⇒ VGAPLUS16	PARADISE AUTO 16 BIT 800x600	\$ 177.00
⇒ VGA 1024-256k	PARADISE AUTO 1024x768	\$ 194.00
⇒ VGA 1024-512k	PARADISE AUTO 1024x768	\$ 247.00

#### MEMORY CHIPS, ETC.

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

⇒ 41256 256x1 80 ns.....\$2.95	SIP 1Mx9 80 ns .....	\$68.00
⇒ 41256 256x1 100 ns .....	SIM 1Mx9 80 ns .....	\$69.00
⇒ 41256 256x1 120 ns .....	SIM 256x9 80 ns .....	\$24.00
⇒ 1Mbit 1Mx1 80 ns .....	V-20 replaces 8088 CPU	\$14.00

#### Z-100 SERIES COMPUTER UPGRADES

##### High Density 1.2 Meg Drives

⇒ External floppy drive set-up complete with drive, power supply, case and cables. Ready to connect to your 8" floppy controller. Single Drive Unit \$207.00  
 ⇒ Dual Drive Unit \$299.00 Bare drive and cable for internal mount \$115.00

##### ZMF100A by FBE Research

⇒ A modification package which allows 256K chips to be used on the old-style motherboard (part number 85-2653-1) to reach 768K. Simple assembly with no soldering or trace cutting. Compatible with Easy PC and Gemini Emulator. Order 27 256x1 RAM chips to complete this kit. ZMF100A.....\$60.00

⇒ Please Mail, Phone or FAX your order. All hardware carries the manufacturers warranty plus the PAYLOAD 90 day guarantee. No surcharge on credit card orders. COD shipments on request. Add \$5.00 to all prepaid orders for handling and shipping in the Continental USA, we pay the balance. Actual shipping costs for foreign, overseas and net billing orders. We accept purchase orders from schools, government and approved accounts. Mail or phone your order for prompt friendly service. Texas residents please add 8.25% state sales tax.



15718 SYLVAN LAKE \* HOUSTON TX 77062

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

**PAYLOAD \* PAYLOAD \* PAYLOAD \* PAYLOAD**



# CALENDAR.CAL

**Steven W. Vagts**  
**2409 Riddick Road**  
**Elizabeth City, NJ 27909**

Been frustrated by the calendar functions of standalone desktop utilities? I have been.

The calendar function is usually quite nice for appointments, but I need more of a weekly or monthly planner. As a Coast Guard Air Station Operations Officer, I needed a calendar that showed, at a glance, deployed aircraft - date of return or relief by another aircraft - or personnel trips with dates of return. Such a planner could be used for any type of scheduling need.

I recently wrote to one shareware program developer and vented my frustrations on the poor souls. As usual, their program had a nice month screen and a week screen, but the only way to add events was through a daily appointment screen, forcing you to use hour slots. I explained that I wanted a calendar planner that permitted me to do what I would on a paper calendar - run a line through several days and place above it, "vacation", or, "trip." Other lines over some other days might say, "deployed aircraft", or, "CO on leave." A computer display should show two or more weeks at a time. It should also be capable of printing hard copy.

I would think it would be relatively easy to develop a "Planner" function to their program that modified the week screen such that instead of showing the hourly appointments, it had "event" lines. Each of these lines would represent an aircraft, class, person, or other event and would continue over all the days that the asset was gone. Where the events don't overlap, the same line could be used again for a different event.

To enter the information, I would envision opening the window for the day the trip starts, in this case Jan 3rd. Move the cursor down to any line that's vacant for the entire period - say we chose the 4th. In response to a question asking the end

date, we say 1/6. Next, we're asked for a short phrase to enter, and we type "HC-130 to Florida". It has to fit inside the width of the period (the highlighted area). Upon a "RETURN" the period is highlighted with the appropriate message. We would then do the same for the second trip. We should also have the capability to delete the trip or change the dates (probably by deleting first, then entering a new one).

Well, maybe it's too much to ask. It sure was a slick package, except for that.

Then, one morning as I lay awake waiting for the alarm to tell me its the start of a new day (as always, my most productive time), it hit me - a spreadsheet could do most of what I wanted. In fact, I already used something similar, but not nearly so sophisticated, for tracking pilot qualifications. It would be a simple matter to develop a calendar, showing as much as two weeks at a time, and have the ability to scroll anywhere in the year! And the spreadsheet provides considerably more flexibility in designing the calendar to suit your needs!

## Spreadsheet Preliminaries

The spreadsheet that follows uses "Peachcalc", integrated as a portion of "Peachtext 5000", from Peachtree Software. It is very similar to the original "Supercalc" program from Sorcim Corporation that ran on the H/Z-89.

As you can see in Spreadsheet 1, "Peachcalc" uses numbered rows and lettered columns. It uses direct addressing - 'B6' refers directly to that cell. Other spreadsheets use relative addressing, numbering both rows and columns, and referring to a cell by jumping backward or forward a certain number of rows and/or columns - 'R[-2]C[+3]' refers to a cell up the page two rows and the third column to the right from the present cell.

"Peachcalc" defines a block of cells as

the upper left cell and the lower right cell separated by a colon, e.g., A3:H6 defines a block columns A thru H wide and rows 3 thru 6 deep. I will use ranges quite a bit in copying blocks of cells, so read up on how your program accomplishes this important capability.

Another difference you may notice between spreadsheet programs is that some will not permit text greater than the individual cell width. "Peachcalc" will display text of any length as long as succeeding cells are blank. For example, in Spreadsheet 1, the entire text in cell 'B1' is "VAGTS PLANNING CALENDAR". It will display in full as long as 'C1' is left blank. If something were placed in 'C1', the full text of 'B1' would not be shown, but would be truncated to "VAGTS PLAN".

This last function will be extremely helpful in our calendar, where an event will span several days. The other programs will force portions of a statement to be placed in individual cells (each representing a day).

Watch for other differences as we go along. "Peachcalc" assumes everything is a formula, unless there is a double quote, ", as the first character. Therefore, the actual text in 'B1' is "VAGTS PLANNING CALENDAR".

Use the Copy, (/C), or the Repeat, (/R), commands when able, but be alert to formula changes which may require correction. Cells are automatically updated when copied, but we will be using some cells, such as E1 and M1, as constants. Reference to these cells will change when the Copy or Repeat commands are used. Other spreadsheet programs will treat these cells differently.

## The Planner Spreadsheet

Looking at Spreadsheet 1, I'm sure you're saying, "That looks too cramped to be use-



	B	C	D	E	F	G	H
1	VAGTS PLANNING CALENDAR 1990						
2							
3		Jan- A7:I60		Feb- I7:Q60		Mar- Q7:Y60	
4		Apr- A63:I116		May- 163:Q116		Jun- Q63:Y116	
5		Jul- A119:I172		Aug- I119:Q172		Sep- Q119:Y172	
6		Oct- A175:I228		Nov- I175:Q228		Dec- Q175:Y228	
7							
8							
9	1990 JANUARY PRINT- A7:I60						
10	SUNDAY	Monday	Tuesday	Wednesday	Thursday	Friday	SATURDAY
11							
12	31	1	2	3	4	5	6
13							
14							
15							
16	7	8	9	10	11	12	13
17							
18							
19							
20	14	15	16	17	18	19	20
21							
22							
23							
24	21	22	23	24	25	26	27
25							
26							
27							
28	28	29	30	31	1	2	3
29							
30							
31							
32	4	5	6	7	8	9	10
33							
34							
35							
36	JANUARY NOTES:						
37							
38							
39							
40							
41	Spreadsheet 1.						

full!" I removed excess rows to reduce space. When you build the spreadsheet, you are to insert three more rows of space in each week. The actual space taken by each month is provided in the table of rows 3 through 6; nearly a full page when printed. If you find this unsatisfactory because you need wider cells or more event lines per day, it's simple to adjust the spreadsheet accordingly, but printing will be more difficult - a month will cover more than one page.

Our spreadsheet today uses a matrix of three months across by four months down, in the order shown in rows 3 thru 6 of Spreadsheet 1. This table also serves as an index to the other months of our calendar. Another arrangement of months could easily be accommodated. I prefer this arrangement as more conventional, at least in my mind. However, because of the way "Peachcalc" performs its calculations, you must recalculate 9 times for a calendar day change made in January to be reflected into December. This is no big deal, because you only need to change the dates when you make a new year's calendar. If you put the calendar in a reversed matrix, such that February was under January, March under February, etc., then you would only need to recalculate twice, bringing the results at the end of April up to May, and the results at the end of August up to September.

I also prefer this matrix because of the way the screen is updated when you scroll vertically with "Peachcalc". It is much easier to move through the spreadsheet horizontally, than vertically. I think the program requires the full screen to be dis-

played before proceeding down another row. The slowness of the Z-181 screen doesn't help, either. I don't have the problem with other programs, but it's a real pain with this one.

Finally, due to program differences, I'll give the general "Peachcalc" command to use rather than the specific order of key entry when we build our spreadsheet. You should be somewhat familiar with your program's basic commands before you start. Well, let's get on with it!

### Building the Spreadsheet

Using a clean worksheet, adjust all the column widths to 10 using the format, /F(Global) command. Then, change columns A, I, Q, and Y to 1.

Enter your spreadsheet name in cell B1. Place 1990 in cell E1. All months will refer to this cell and will therefore change when you change this entry.

Again using the format, (/F), command, but this time using the E (entry) level, format the range C3:C6 to TR (text right). Do the same for the E3:E6 and G3:G6 ranges. Enter the text shown in rows 3 thru 6 of Spreadsheet 1 into their appropriate cells.

Enter E1 in cell B8. Enter " JANUARY in cell E8.

Format, /F, cell G8 to TR (text right), then enter "PRINT- Enter " A7:I60 into H8. This is the range to use when you print January's spreadsheet.

Go to cell A8 and enter "|. Now using the repeat (/R) command, replicate this over the range, A9:A60. This forms the left margin for the month. Using the copy (/C) command, copy from range A8:A60 to cell I8. This places a dashed line vertically down

the right side and also serves to separate months.

Back to cell B9, enter a single apostrophe and the equal sign (=). This places a double dashed line horizontally, ending where it intercepts the vertical line in cell I9. You may have to enter something different in your spreadsheet. Check your manual.

In row 10, enter the days of the week in their appropriate cells. Except for SUNDAY, I started the cell with a vertical dashed line to show the start of each cell. For example, in cell C10, enter "| Monday. I capitalized SATURDAY and SUNDAY.

In cell B11, enter '-. As with the equal sign placed in cell B9, the row should fill with a dashed horizontal line.

In row 12, enter 31 in B, 1 in C, and the formula, C12+1, in D. Using the repeat (/R) command, replicate D12 into E12:H12. The formulas should update automatically.

In cell B18, (remember, I deleted 3 rows from each week of Spreadsheet 1.) enter '-. This places another dashed horizontal line across the screen.

In cell B19, place H12+1. In cell C19, place B19+1. Using the repeat (/R) command again, replicate C19 to D19:H19.

Now using the copy (/C) command, copy from B18:H19 to B25. All formulas will update automatically.

Similarly, copy from B25:H26 to B32.

In cell B39, enter '-.

Now it gets complicated. I wanted the spreadsheet to update itself whenever I changed the year and the first few days in January. The spreadsheet must then test for the 31st day in January so it can start February in the proper place.

In cell B40, enter IF(H33=31,1,H33+1). We check if H33 equals 31. If it does, then cell B40 is 1; if not, then B40 is H33+1.

In cell C40, enter IF(B40=31,1,B40+1). Using the repeat (/R) command, replicate C40 into D40:H40.

Copy (/C) from range B39:H40 to B46. The formulas should update.

Now you might be asking yourself, "Why is he making a month six weeks long?" If you look at a paper calendar, there are going to be times when a month starts on a Friday or Saturday, then runs into a sixth week. Also, you may have events running into the next month that you want printed out as part of the preceding month.

Finish the month by entering '- in cell B53, "JANUARY NOTES: in B54, and '= in cell B60. The notes block will come in handy for noting birthdays, anniversaries, holidays, etc., that, if placed into their actual respective date, would not be moved with the date when you started a new year. Some newer spreadsheets might be able to move text from another block, but none of mine are capable. With the information in the notes area, it can easily be manually placed in the respective day blocks if desired.

That finishes January. Debug by trying to



	J	K	L	M	N	O	P
1:	LEAP YEAR (NO=0, YES=1): 0						
2:							
3:							
4:							
5:							
6:							
7:							
8:	1990 FEBRUARY PRINT- I7:Q60						
9:	-----						
10:	SUNDAY:	Monday:	Tuesday:	Wednesday:	Thursday:	Friday:	SATURDAY:
11:							
12:	28	29	30	31	1	2	3
13:							
14:							
15:							
16:	4	5	6	7	8	9	10
17:							
18:							
19:							
20:	11	12	13	14	15	16	17
21:							
22:							
23:							
24:	18	19	20	21	22	23	24
25:							
26:							
27:							
28:	25	26	27	28	1	2	3
29:							
30:							
31:							
32:	4	5	6	7	8	9	10
33:							
34:							
35:	-----						
36:	FEBRUARY NOTES:						
37:							
38:							
39:							
40:							
41:	Spreadsheet 2.						

change the day of the week for the 1st and check the results on the rest of the month.

To begin February, copy (/C) command, from range B7:I60 to cell J7.

February has its own quirks, see Spreadsheet 2. First, we need to be able to handle Leap Years! Actually, that's easier than it sounds. Leap Years are found by dividing the calendar year by 4. If it divides evenly, then it's a leap year.

In cell J1, enter " LEAP YEAR (NO=0, YES=1):".

In cell M1, enter IF(INT(E1/4)=E1/4,1,0). If the integer of the quotient equals the normal result, it's a leap year and a one is placed in M1. If they aren't equal, a zero is placed in M1. Format, /F command, the entry to place the answer in the left side of the cell.

Put E1 in cell J8, " FEBRUARY in M8, and " I7:Q60 in P8.

The first week of data in February must be changed. In cell J12, enter IF(B40>25,B40,B47). This will determine where the end of the previous month falls and will start the next month somewhere in the first week.

In cell K12, enter IF(J12=31,1,J12+1). Using the Repeat (/R) command, replicate cell K12 to L12:P12. These check which day of the week is the 31st.

The end of the month needs to be changed also - February normally only has 28 days, and we have to account for the leap year.

Edit, /E command, cell J40 to IF(P33=28+M1,1,P33+1). Remember M1? Ok, continue by changing K40 to IF(J40=28+M1,1,J40+1). Similarly, change the rest of the last two weeks of February

the same way.

Edit (/E) cell J54 to "FEBRUARY NOTES: On to March! Copy (/C) block B7:I60 to R7.

Put E1 in cell R8, " MARCH in U8, and " Q7:Y60 in X8.

Again, we edit the first week. Cell R12 contains IF(J40>(22+M1),J40,J47). Cell S12 contains IF(R12=28+M1,1,R12+1). Similarly, edit the remainder of the first week of March the same way.

Like January, March also has 31 days. Therefore, when we copied January to March, the formulas in the last two weeks should also have changed properly. Check cell R40 contains IF(X33=31,1,X33+1) and cell S40 contains IF(R40=31,1,R40+1). Similarly, check the other cells.

Copy the first three months to the second three months by copying block A7:Y60 to A63.

Edit row 64 such that the year refers to E1, the months reflect the proper months, and the print range is the same as that given in the table at the start of the spreadsheet.

For April, edit the following cells:

B68 = IF(R40>25,R40,R47)

C68 = IF(B68=31,1,B68+1)

D68 = IF(C68=31,1,C68+1)

Similarly, do E68 thru H68.

For May, edit the following cells:

J68 = IF(B96>24,B96,B103)

K68 = IF(J68=30,1,J68+1)

Similarly, do L68 thru P68.

For June, edit the following cells:

R68 = IF(J96>25,J96,J103)

S68 = IF(R68=31,1,R68+1)

Similarly, do T68 thru X68.

By now, you may have seen some similarities and differences between the months.

For example, the values within the formulas used in J12, R12, B68, J68, and R68 depend upon the number of days in the preceding month: 25 if 31 days, 24 if 30 days, and 22+M1 if 28 days (+1 for a leap year).

Change the last two weeks of these months, also. The values in the formulas are the number of days in the month. For April, which has 30 days, edit the following cells:

B96 = IF(H89=30,1,H89+1)

C96 = IF(B96=30,1,B96+1)

Similarly, do D96 thru H96.

B103 = IF(H96=30,1,H96+1)

C103 = IF(B103=30,1,B103+1)

Similarly, do D103 thru H103.

For May, which has 31 days, edit the following cells:

J96 = IF(P89=31,1,P89+1)

K96 = IF(J96=31,1,J96+1)

Similarly, do L96 thru P96.

J103 = IF(P96=31,1,P96+1)

K103 = IF(J103=31,1,J103+1)

Similarly, do L103 thru P103.

For June, edit the cells in a similar manner. Since June has 30 days, use 30 in the formulas.

Correct the cells labeling the notes area of each month.

We're half way through, and the rest is all downhill!

Copy the first six months to the last six months by copying block, A7:Y116 to cell A119. As we did to the previous three months, correct the Year cells, month names, and print range cells. Due to the length of this article already, I leave the formula corrections to the reader. The changes are straight forward and there are no new pitfalls.

Following completion of all the changes, check your 1990 calendar against a regular calendar. The days of all the months should be correct and fall on their proper days of the week. If they are not correct, try a recalculation or two (shift-I in "Peachcalc") and check for changes. Remember, it takes nine recalculations for a change made in January to be reflected in December.

If there is still an error, recheck your formulas in the first month having the error.

### Using the Spreadsheet

Using the spreadsheet is simple, at least in "Peachcalc." Before doing anything else, establish a generic Calendar, perhaps under the file name "CALENDAR.CAL". At the start of the new year, open this new file, change the year, in cell E1, and the first few days in January; the spreadsheet takes care of changing the rest of the year. Perform 9 recalculations as we discussed above. Finally, save the spreadsheet under a new file name, perhaps as "YEAR90.CAL" or "CALEND90.CAL", leaving the "Father" file unchanged for future use.

I suggest using the lesser than (<) and

greater than (>) symbols to delineate the beginning and ending of a period in excess of a day. For example, show a trip between the 3rd and 7th by entering "<Trip to Florida, in the first applicable cell (for the 3rd), then skip to the cell representing the 7th, and simply insert ">. Placement of the greater/less than signs in the first and last appropriate blocks might indicate how late in the day you depart or arrive, i.e., the further right you place the symbol, the later in the day you depart or return.

For something like school starting - that won't end again within the month - you might place 1 or 2 hyphens after the less than symbol (i.e., <-School Starts) or, for school closing, perhaps "School ends->. For the second example, insert the statement in the cell before, so the end of the statement falls under the right day.

As I mentioned earlier, holidays, birthdays, anniversaries, and other single day events must be inserted after the basic calendar has been made, since they will be under the incorrect day if we change the calendar after they are entered. They can be placed in the monthly notes spaces of

your generic calendar (CALENDAR.CAL) where they won't get lost, then entered manually in their correct day when you have established your new file name, if you wish. For example, enter the following in the appropriate cells:

The question mark signifies that the actual day differs each year. Also, these entries reflect the actual holidays, not the days they are observed, which also differ each year. Enter birthdays, anniversaries, and other annual special events in a similar manner. Save the finished file under your generic file name.

- |      |   |  |
|------|---|--|
| B59  | = | "1) New Year's Day   |
| J58  | = | "2) Groundhog's Day, 12) Lincoln's BD, 14) St. Valentine's Day   |
| J59  | = | "22) Washington's BD   |
| R59  | = | "17) St. Patrick's Day, 20) Spring Begins, ?) Easter             |
| B115 | = | "1) All Fools' Day, 25) Secretaries' Day, ?) Easter              |
| J115 | = | "2nd Sun) Mothers' Day, 30) Memorial Day                         |
| R115 | = | "14) Flag Day, 3rd Sun) Fathers' Day, 21) Summer Begins          |
| B171 | = | "4) Independence Day   |
| R171 | = | "1st Mon) Labor Day, 2nd Sun) Grandparents' Day, 23) Fall Begins |
| B227 | = | "12) Columbus Day, 31) Halloween                                 |
| J227 | = | "1st Tue) Election Day, 11) Veterans' Day, 4th Thu) Thanksgiving |
| R227 | = | "21) Winter Begins, 25) Christmas                                |



To save space, I don't use any markers with single day events, such as "B&G ANNIV, or "DAVE'S BD, and I only use a hyphen to call attention to appointment times, such as "-0930 DOC (doctor's appt), or "-1000 DNT (Dentist appt).

This challenging project is another example of using versatile spreadsheet programs. I hope you enjoy the project and it works as well for you. If you have any changes or enhancements you feel would increase the capabilities of this spreadsheet, please don't hesitate to write.

## Concerned Subscribers

Lisa Cobb  
REMark Staff



By now, I'm sure you're well aware of the difficulties we've encountered at HUG. If not, let me explain..

Zenith Computer Group was bought by Groupe Bull of Paris. This merger occurred in 1990. Through the transition, we have had many adjustments and interruptions in the daily activity of the REMark magazine and our group.

One change that we have incurred, is that HUG is no longer affiliated with Heath. We are now a department within Zenith Data Systems, with no official word yet of a name change for the group.

The changes within the company will

account for the disappearance of the ID card. With Heath changing its focus towards Home Automation and Educational products, there will be less emphasis on computers. Therefore, Heath will no longer extend discounts to computer purchases for HUG members.

Along with all the other changes, the Postmaster has notified us of a zip code change. REMark's address is Zenith Users' Group, PO Box 217, Benton Harbor, MI. 49023-0217.

Now, some of the delays that we've encountered, are due mainly to staff shortages within our group. Right now, we are two staff members short, putting

us well behind on the last few issues.

The December 1990 issue was sent at the end of January, the January 1991 was sent the first week in March. The February was mailed in the last weeks of March and the expected date of March is still a couple of weeks away.

To the best of our knowledge, REMark magazine will continue to be on the leading edge of computer user information. We ask your understanding and patience while we try to regain our momentum.

Thanks for your continuing support.



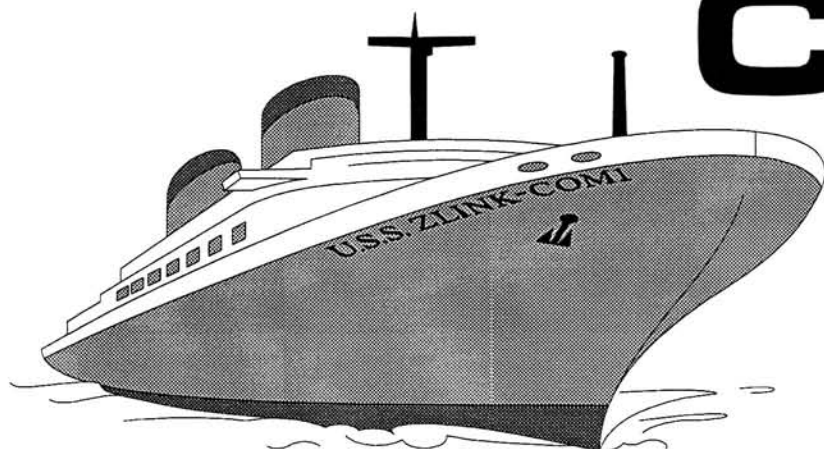


---

---

# Port-O-Call:

# COM1



Laura White  
759 Polfus  
Benton Harbor, MI 49022

In spring of 1990, REMark subscribers were urged to get a modem if they didn't have one because the bulletin board was receiving a facelift. Well the time has come once again for you to put your modem to use if you haven't already. COM1, the chosen name for the current Bulletin Board has several new arrivals — the latest: 16 phone lines, FidoNet, EchoMail, and increased speed just to name a few added attractions!

Originally, five or six years ago, the bulletin board was designed to support one telephone line with a little more than a 100 message data base and an upload/download file system. After some years, the Bargain Centre came into being and before too long, the HUG bulletin board was infamous; HUG was becoming a larger group and in need of more telephone lines. After last spring's change from old reliable HUGPBBS (Heath User's Group Personal Bulletin Board System), to the new and improved TBBS (The Bread Board System), changes have been happening regularly: lines have become more available and plentiful, information has become more abundant, and efficiency is becoming a way of life for the board.

In the past few months, COM1 users have had the opportunity to experience: the Conference Area, Information and Bulletins, and System and Configuration Functions. These are all new and in addition to the previously mentioned Message System, Files Database, and the ever popular Bargain Centre. The *Conference Area* allows you to communicate directly to other users who are currently on the system. This may be done on an individual basis or in groups; you may also find out just who else is on the system at that time. Under the *Information and Bulletins* Section, you can

get information on Registration for the HUG Bulletin Board, as well as subscription information. Bulletins from the SYSOP can also be found under this heading. The *System and Configuration* heading puts you in contact with information regarding space usage on the hard drive, who is currently on-line, user statistics and, terminal configurations.

The most recent additions, FidoNet and EchoMail have opened new doors in the world of telecomputing! FidoNet allows TBBS users to network into other bulletin boards that are also on the FidoNet system. This means that communication is now possible on a network to network basis. In addition, FidoNet possesses search capabilities; huggies can now search, for any reason, by city, county, state, region, etc. An example of how this can be useful is, a huggie noticed a message a few weeks back from someone who wanted to purchase a hard drive. The owner of the hard drive can only remember that the guy was from Florida and wants to search for his potential sale. Rather than searching through the entire board, he simply uses FidoNet to search by State, thus cutting his time drastically. With this recent addition to the family, the number of resources available through the bulletin board has increased a hundredfold!

A second change is EchoMail. This allows users to send a random message out to other systems, for example, a Z-100, with ECHO mail, is now able to talk to an Omega system. Stated Jim Buszkiewicz, COM-1 System Operator, "At present, interest in responding lies entirely in the hands of the receiver." In the future, Buszkiewicz is interested in starting a ZDS EchoMail system so that ZDS users can all give and receive information more effi-

ciently without necessarily needing to be a fellow huggie.

A third change lies in the on-line Gaming area of the bulletin board. We have all played games against the computer. Now, you can actually play against another live opponent and there's a very real possibility that this opponent may be across the country! If you are tied into the system and are playing a game of "Starquest", there is a good chance there is someone else in or near your sector! It adds a whole new dimension to computer games. (Oh, be careful where you leave yourself — you can be destroyed while you are gone!)

There is also a Medallion section. This is a section most huggies won't use. It is a network between authorized ZDS dealers and their customers. You are allowed to send a message privately to a dealer and your return answer would also be private. This limited access allows dealers to hold their different discussions, yet allows huggies to tap into a very valuable resource.

Finally, the bulletin board now has 16 phone lines hooked up — all of which are 2400 baud. Now, you can make contact more quickly for those great deals available through the bargain center. Just how quick you ask? With 16 lines, you would experience only a 10% degradation at 8 MHz, and ZUG is running with 33 MHz machines. So now, you can feel confident that you will be receiving speedy, efficient service when you dial up the bulletin board. There is yet another advantage to you with all these additional phone lines. For example, in the past, when ZUG would have a sale, the system would have to be shut down while the sale items were entered. Now with 16 lines, software is freed up tremendously, making multi-tasking a regular part of the bulletin board's smooth

running system.

When asked what the plans were for the future, Jim Buszkiewicz replied, "We are only limited by our imagination." By April 1991, The Bread Board System, will have a new version available. ZUG intends to purchase it giving the system even more power than it already has. With this new version, there will be 64 phone lines which should be more than enough. Presently the bulletin board's 16 lines aren't being used to their full

capacity except during a "pieces-parts" sale. Once Zuggies begin to use the new additions to the system, they will realize that the ZUG bulletin board has enormous power and potential and there will be a real need for the full 64 lines.

Finally, an area that zuggies will be interested in is Zenith Data Systems EchoMail which is in the planning stages. You can also look forward to the bulletin board having U.S. Robotics: high speed modems that can handle up to 9600 baud.

With all these changes, you might be asking yourself just how much is all this going to cost? Well there's good news! Membership in Zenith Users' Group is still \$22.95 per year for new members and \$19.95 for renewals, with a 20% discount to those who renew their membership on-line. So if you haven't already, there's more reasons than ever to get yourself a modem and dial up (616) 982-3956; chances are, you won't be disappointed. ✨

All Checks must be made out to  
Zenith Data Systems

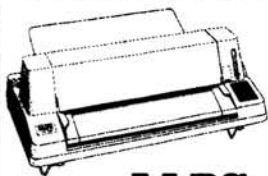
_____	January 1,	1991
_____	Zenith Data Systems	00.00
Zero dollars and zero/100		
S A M P L E		

# W S Electronics

(513) 376-4348 \*\*\*\* Since 1975 \*\*\*\* (513) 427-0287

1106 State Route 380, Xenia, Ohio 45385

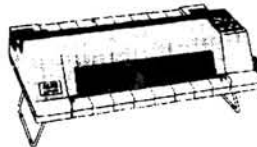
## YOU AND YOUR BIG IDEAS.



- \* 300 cps draft speed
- \* Wide carriage
- \* 24 pin printing
- \* Front panel controls

**ALPS** Allegro 500XT™

## THE ALPS ASP1600 PRINTER. COSTS VERY LITTLE, JAMS NOT AT ALL.



- Auto tear bar prevents paper waste. With the flick of a button, your fanfold paper advances to the perforation, then automatically returns to top-of-form position.
- Rugged 9-pin head delivers crisp output at 192 cps in draft mode, 38 cps in letter quality.
- Compact 9-lb. body makes for easy portability.
- Printer stand is built-in.
- Prints labels easily.
- Full Epson FX-85 compatibility.

**ALPS**  
AMERICA

Built by popular demand.

## SPECIAL

Z-415 1.5 Meg Ram for Z-248	\$100.00
Z-505 1 Meg Ram for Z-386	\$200.00
Z-315 EMS Kit for Z-159	\$ 10.00
Z-417 H. D. Controller Z-248	\$100.00
Z-317 H. D. Controller Z-150	\$ 75.00

Quantities limited to stock on hand

Attention: Federal Government Offices  
We stock ALL ALPS Printer Models and  
we stock ALPS PARTS and RIBBIONS for all  
ALPS models including your P2000's and ASP  
1000's

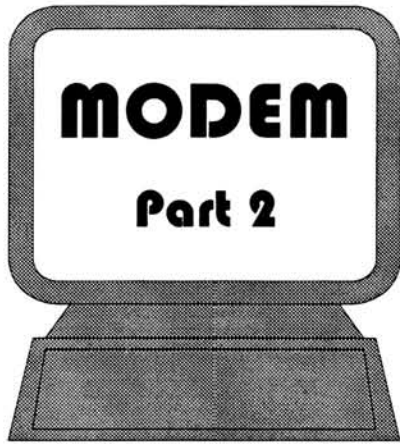
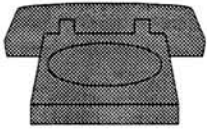
\*\*\*\*Government Discounts Offered\*\*\*\*

We are looking for good dealers.  
ALPS Authorized Distributor and Service Center.



# MODEM

## Part 2



## A Data Transfer Solution

**Robert C. Brenner**  
**Brenner Info Group**  
**9282 Samantha Court**  
**San Diego, CA 92129**  
**Copyright (c) 1991 Robert C. Brenner**

The previous article described various modulation and data transfer techniques (AM, FSK, PSK, DPSK, QM, QAM). System performance is limited by the 3100 Hz bandwidth restriction (300 to 3400 Hz), by the 2400 baud theoretical maximum transfer rate, and by inherent defects in the dial-up telephone system. This article discusses the inherent defects and how innovative designs are getting around these problems.

### Factors Affecting Data Transfer

With modem communications encompassing a worldwide net of interconnecting data paths, any transmission path defect can affect good data transfer. Problems occur because:

- transmission paths vary in quality
- paths can be different each time a link-up occurs
- signals tend to become distributed along the path
- signals get weaker as the path gets longer
- compressed signals can lose phase information
- older copper wires can introduce echoes, amplitude jitter and data loss
- telephone wiring in old offices and homes can introduce crosstalk, noise and intermittent distortions
- fiber optic cables can affect signal phase information
- microwave links can introduce amplitude, frequency, phase, and echo problems
- transoceanic cables can introduce amplitude, phase and frequency distortion
- satellites introduce additional delay (600 ms) and can cause phase and amplitude distortion
- crosstalk can occur between two different adjacent signal paths
- near and far echoes can confuse correct signal detection
- mechanical switches and nearby lightning can introduce impulse noises causing bursts of errors in a data stream

- electronic circuitry can introduce non-linear noise distortion
- different bell ring voltages can introduce pitch (phase) interference
- analog equipment generates more white noise (hiss) than digital equipment
- picking up an extension phone during transmission can reduce signal amplitude
- dialing on an extension phone during transmission can disconnect the link

Figure 1 shows the effects of passing a data signal through a poor quality telephone line. With so many things affecting data transfer, it's amazing the system works. Potential problems are addressed through better transmission paths, better modem design, and innovative ways to package and transfer information.

We shield telephone wires, install electronic switching stations, and replace copper wire with fiber optics to improve the transmission paths. There's little we can do for pathways outside our country so im-

provements in modem design and special algorithms do the rest.

As described in the last article, modem design has steadily improved. More data is passed within the 3100 Hz bandwidth limitation by phase and amplitude shifting the signals. At 2400 baud, QAM modems can transmit 9600 bps successful one way. But there isn't enough bandwidth to allow simultaneous (full duplex) bi-directional communication between two 9600 bps modems without major interference. Therefore, early 9600 bps modems simulated full duplex by ping-ponging signals. The originating and answering modems alternatively transmitted and received. This half-duplex technique worked fine on clean lines, but was inefficient.

### Echo Cancelling

To achieve high speed full-duplex data transfer, each modem sends while cancelling echoes of its own transmission. Reflections of the original signal, caused

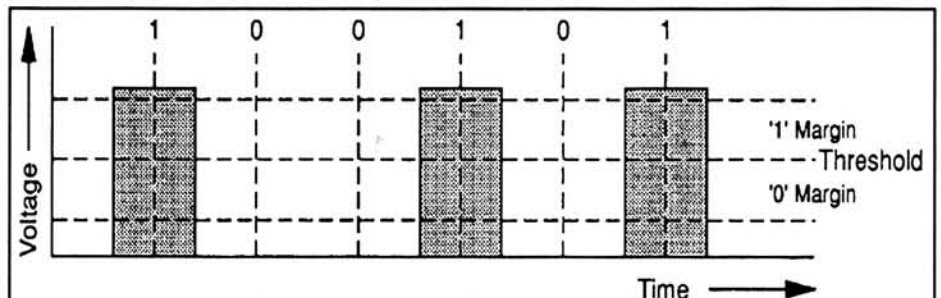


Figure 1A. Data at time of transmission.

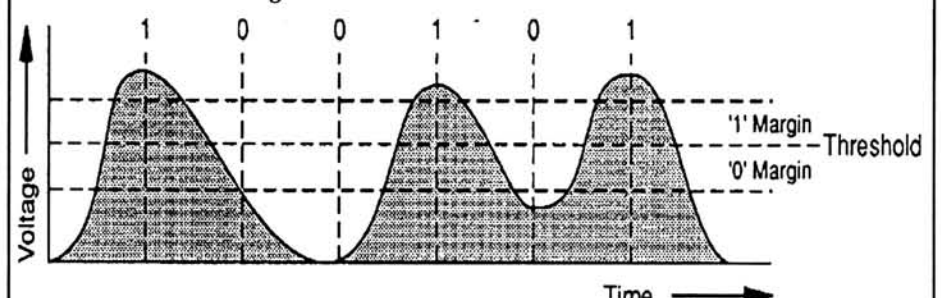


Figure 1B. Data at receiving end.

by impedance mismatches from telephone equipment in the transmission path and delays from satellites, scramble and distort the original data. In voice communication, this sounds like crosstalk.

The simultaneous signals from two 9600 bps modems co-mingle causing each modem to detect a composite of sending and receiving tones and echoes. By internally cancelling the echo of its own sending signal, only the primary tones from the other modem and some transmission path echoes remain. The digital signal processing circuitry that performs echo cancelling generates a delayed copy of the transmitted signal and then compares it with an incoming signal. It filters out its own signal echoes and some line echoes making all but the signal from the far end modem too weak to affect reception. Echo cancelling algorithms also accommodate a 600 ms time delay for satellite links in the path.

Poor phone lines rob the modem signal of strength and introduce noise distortion. As worldwide communication becomes common, our need to transfer information accurately gets more difficult. Shielding wires to prevent crosstalk from nearby signals helps, but other defects remain. Fiber optics also helps, but not all of our globe uses fiber optic media. Much of our planet continues to share information over pairs of thin, barely-shielded wire.

Dedicated leased lines can be conditioned (rated C or D) to control the frequency response and channel characteristics, but two-wire dial-up telephone lines are considered unconditioned so modem electronics must compensate for poor line quality.

### Error Rate

Undesired signals called "noise" introduce transmission path interference that directly affects line quality. The ratio between a modem's desired signal level and the background noise is called its signal-to-noise-ratio (SNR). SNR is measured in decibels (dB). A 6 dB change represents a doubling (or halving) of signal strength. Most 300 baud modems run fine with an SNR 6 dB. The 1200 baud modems need 12 dB SNR, and 2400 baud modems need 21 dB. Dial-up phone lines have a theoretical maximum SNR of 15 dB so many of our long distance connections don't meet requirements. One AT&T study concluded that 25 percent of long distance calls are made over poorly conditioned lines; 15 percent are over terrible lines, and the remaining 10 percent are attempted over hopeless lines. To achieve a successful data transfer, modems must implement sophisticated error-correction circuitry or the transmission rate must be reduced.

Routing is the most frequent factor affecting errors in dial-up modems. Each time you dial a number it's likely that the transmission path will be different. This

problem doesn't exist on leased lines because the same communication path is always used.

Since we can't correct most line quality defects (without disconnecting and redialing onto a cleaner telephone line), modems incorporate software and hardware to compensate for noise distortions. High speed signals have increased the risk for errors. As many as three QAM characters can be affected by a single bit error. To minimize this, the signal is sometimes scrambled before transfer and then descrambled at the receiving end. If an error is detected, the descrambler resynchronizes before continuing to receive.

A "voice-grade" telephone channel meets the Bell Tariff 3002 standard specifications with less than 1 bit error in 100,000 bits transferred and with received white noise less than 3 dB above the carrier detect threshold.

### Equalization

As transfer rates increase and phase amplitude modulation becomes common, modem design becomes increasingly complex. The number of unique phase shifts that can be accommodated by a modem has a finite limit because the telephone system itself causes noise (chop) that makes subtle phase shifts undetectable. A technique called "equalization" effectively corrects linear telephone line distortions and compensates for line inconsistencies. Equalizers act like filters to distort the gain in a direction that will compensate for expected or detected noise interference.

A compromise equalizer with adjustable set points can be accurately adjusted for local conditions or transmission line variations on a leased line but works unreliably on a dial-up line because it can't automatically adjust for changes in different transmission paths.

In the early 1960s, Bell Labs pioneered an automatic adaptive equalizer for PSK modems that dynamically compensates for line distortion. Built-in circuitry establishes imaginary decision boundaries that bisect the quadrants of the phase samples of each signal element. By comparing the sampled signal with the decision boundaries, the equalizer determines which quadrant the signal is in. Signals occurring near boundary limits cause the decision device to compute an error vector that minimizes future errors. All high speed modems now use automatic adaptive equalizers. Most medium speed modems have used only compromise equalization due to cost and complexity. Adaptive equalization substantially reduces error rates (even on poor lines) by compensating for line irregularities when sudden disturbances, such as sharp impulses, project signals across decision boundaries.

### Error Detection

Phone lines are noisy, and a particularly bad connection can result in numerous re-transmissions before blocks of data can be successfully exchanged. Both hardware and software methods are used to sense and correct transmission errors during data transfer.

The cyclic redundancy check (CRC) was developed to detect errors that occur during data transfers. The information being sent is passed through a mathematical polynomial circuit producing a unique CRC value that is transmitted with the data. Upon receipt, the receiving device calculates a new CRC value on the data received. It then compares its calculated CRC with the transmitted CRC. If they match, the data is accepted. If not, an error has been detected, and the sending device is requested to retransmit.

Other error detection solutions include sending a parity bit with each character or calculating a checksum on a block of data and sending it with the data.

Hardware error control is better than software control because hardware doesn't require action from you or a communications program. It automatically determines and uses the highest level of error control available in both sending and receiving units.

### Trellis Code Modulation

As transfer speed increases, the grouping of defined signal points in QAM can be increasingly affected by telephone line noise and other distortions.

To minimize uncertainty, "trellis code modulation" (TCM) is implemented to reduce errors in most 9600 bps modems. TCM adds redundancy without affecting the information transmission rate by forming a constellation with more symbols than necessary to represent all possible bit combinations. Not all transitions are legitimate. If the receiver detects a symbol that falls between the points on the constellation, it uses its knowledge of previous symbols to rule out illegal symbol transitions and chooses the closest legitimate symbol. This spreads the decode information among several symbols resulting in a doubling of the SNR.

Four-bit-per-symbol QAM designs modulate five bits instead of four onto each phase change. The receiving modem checks the extra redundancy bit against a complex code to determine symbol accuracy.

Another TCM design divides seven-bit ASCII characters into groups of five and two bits each. The two-bit group is passed through a convolutional encoder that generates an additional redundant bit dependent on successive signal points. Eight bits are thus transmitted during each baud period resulting in a signal constellation of 256 possible points. Each of these



256 points is assigned to one of eight classes so the decoding algorithm recognizes the greatest possible separation of successively transmitted signal points.

At the receiving end, TCM implements two-stage detection. The first stage selects the closest signal point in each class as the most likely alternative (coarse recognition). The distance between the received point and the eight possible positions is measured and recorded. Then the dependencies between the candidates and the eight possible points are calculated using a Viterbi decoder algorithm that considers past and present signals. It examines 16 successively received candidate points before making a decision on a currently received signal. By looking backward for help in determining the present, the Viterbi decoder automatically makes an educated guess and selects the most likely points out of the signal sequence being received. It's smart enough to usually make the right choice.

A third TCM algorithm uses four concatenated two-dimensional signal constellations that introduce dependencies in successive eight-dimensional signal points. This reduces the number of points in each constellation from 256 to just 160. By eliminating the performance penalty caused when the number of signals in a constellation are increased to handle coded and uncoded modulation techniques, this trellis trap significantly improves noise immunity. This "backward looking" historical

## BAUD RATE vs. TRANSFER SPEED

How does baud rate relate to character transfer speed? Recall that an ASCII character is an 8-bit code. There are about 5.5 characters in an average word (U.S. English). Therefore a 300 baud modem will transfer about 37.5 characters per second ( $300/8 = 37.5$ ) or 6.8 words per second ( $37.5/5.5 = 6.8$ ). A 10,000 word document file would take 1470 seconds (24.5 minutes) to send. If we increase the transfer rate to 1200 baud, we send about 150 characters (27 words) each second. This will decrease our 10,000 word data transfer time to just over six minutes. At 2400 baud, we'll send 300 characters each second and further decrease our transfer time to about 184 seconds (just over 3 minutes). The difference between 24.5 and 3 minutes is significant at long distance telephone rates. A further increase to 9600 bps using QAM will let us send the same document file in just over 45 seconds! And if we implement data compression, we can handle it in just 22.5 seconds! Enough said?

DATA AMOUNT	1,200 bps	2,400 bps	9,600 bps	9,600 With 2:1 Data Compr.
10,000 word file	6.1 min.	3.01 min	45 sec.	22.5 sec.
20 pages of text	10.4 min.	5.0 min.	1.0 min.	0.5 min.
360 KB file	1 hour	31 min.	8 min.	4 min.
1 MB file	3 hours	1.4 hrs.	22 min.	11 min.
4 MB file	12 hours	5.6 hrs.	89 min.	46.5 min

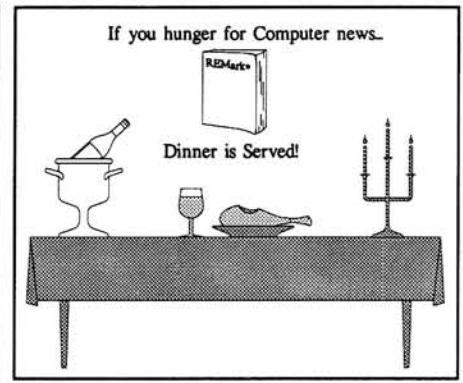
technique yields a 64-state, eight-dimensional TCM in which small increases in circuit complexity result in significant increases in system performance. This design incorporates an error-probability function that automatically adjusts the transmission rate if the line conditions deteriorate or improve. This provides maximum error-free data transfer at the optimal rate for line

conditions.

Trellis code modulation is first line of error reduction defense for high speed modems. In the next article, you'll learn how data compression techniques such as Huffman and Lempel-Ziv coding have lifted us into the next era of high speed data transfer. ✨



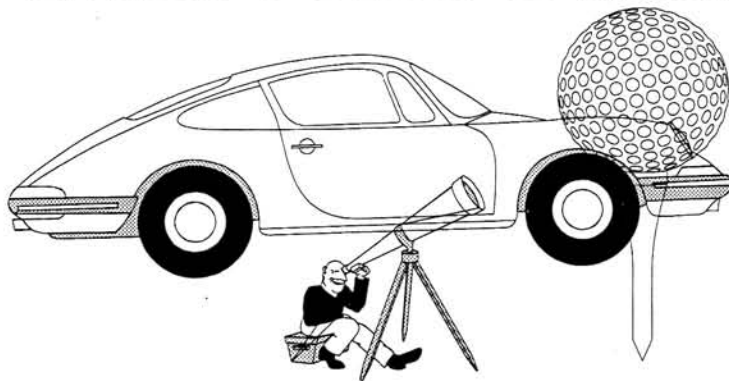
"PLEASE DAD— ONE MORE MATH PROBLEM... OKAY?"



**RAPID INDEX:** A computer program that lists titles of magazine articles by subject or in alphabetical order.  
 CPM or  PC/MS-DOS 2.1 or later.  
**DRIVE:**  
**COMPUTER:**  5.25"DD...\$21.00  
 Z-100 Models  5.25"HD...\$23.00  
 IBM Compatible  3.5"DD...\$23.00  
 Magazines currently available:  
 Elect. Service & Tech. (1984 - 89)  
 REMark computer magazine (1985-89)  
 Calif. residents add 6.75% sales tax.  
 Send check or money order (US funds) only to:  
**B.U.D. Unlimited,**  
**P.O. Box 503, Elverta, CA 95626.**  
 REMark is a trademark of Heath. Zenith User's Group.

Reader Service #250

# What Price a Man's Toys?



## A Look at Windows 3.0 QEMM and DESQview

Sanford Shapiro  
654 Gravilla St.  
La Jolla, CA. 92037

Some years back, when I was 30 something (in another lifetime, I think) I had a sports car — a red Alfa Romeo. I loved driving that Alfa, but I hated paying for the upkeep. If it wasn't the electrical system that needed fixing, it was a head gasket that leaked — or a clutch that had worn out — or something. There was always something. The repair man could have been a member of my family, I visited him so often. Driving a "sensible" car now is dull, but my budget likes me better.

So what happens to all the money that I save? I now use it to buy computer hardware that I don't need, to run new programs that I won't use. My curiosity makes me vulnerable to every upgrade offer. When Microsoft offered me (as an Excel user) Windows 3.0 for \$50.00, I had to try it — though I knew I would have to buy more memory for my computer.

Windows intrigues me and frustrates me. Like a girl with a pretty face, it turns many heads. But, you wonder, is there any substance beneath the surface? William S. Hall, in the September 11, 1990 issue of PC Magazine, called Windows 3.0: "The new standard." He said about Windows: "The time to move is now." The question is: "Time for who?" It is not the time for me — at least not yet. Is it the time for you? Perhaps if I share my experiences with increasing computer speed, with adding more memory and with using Windows, you can judge what might be useful and cost effective for you.

The computer magazines write about Windows for the "power user." I am not a power user, though I do like power. In fact, I love power. But the programs I use most are WordStar, for writing and PC-File, for a data base. I don't do desktop publishing or computer aided design or even large spreadsheets. Most computers in use today are XT's and AT's, and the most commonly used programs are word processing programs. What does a WordStar user like me have to gain by increasing speed, by adding more memory and by using Windows? You might say that all I

really need is a 4.7 MHz XT computer. Not true. Improving your equipment pays dividends, but not always in the ways that you might expect.

### Computer Speed

Accelerate your computer's speed and improve performance. But why bother if most of your work is word processing? The experts will tell you that the computer will spend most of its time waiting for you to type something on the keyboard. I installed an Intel Inboard 386/PC in my Zenith 151, an XT compatible computer. This 16 MHz, 80386 CPU, combined with a disk cache program running in 256K of extended memory has revolutionized my writing. Andrew Schulman, writing in Byte Magazine, says that using the 386 as a fast XT "... is like buying a top-of-the-line sound system and then using it to play only old 45s or 78s." He is right, of course, but after listening to your old records on a fine sound system, I doubt that you will want to go back to your old record player.

With the faster CPU, all my programs load more quickly, in fact, instantaneously. I can easily exit from one program, load another program, exit that program when finished and reload the first program. Windows can switch between programs, but then everything slows down. I find it easier to exit from WordStar, load PC-File, look up a phone number, exit PC-File and reload WordStar than it is to switch between the two programs using Windows! Programs now run faster, too. How can a word processor go faster? Using WordStar, I now can go from the beginning to the end and back to the beginning of long documents instantly. Spell checks are also faster. Before, when doing a spelling check of a long document, it was time for a cup of coffee. Now the spelling checks go like lightning. My home accounting program, "Mini-Ledger," also impressed me with its improvement. Before, it would keep me waiting while finding and sorting all the checks for a given expense

category. Now I get immediate results, and I am spoiled. The higher speed is not wasted, even with WordStar.

### Memory

A basic fact of computer life is that 640K is the maximum amount of user memory that you can have under MS-DOS. Computer users are constantly surprised and puzzled to find that their investments in more memory cannot be used by their existing programs. If you want to have a disk cache, or a print spooler or a RAM drive, then buy more memory. Otherwise, be careful. But Windows is supposed to change all that, isn't it? Maybe. Maybe not.

There are three kinds of memory: conventional memory, expanded memory and extended memory. Conventional memory is the first one megabyte of memory. But translated into practical terms, that means 640K — the rest is reserved. Some clever programs have been designed to use areas of the reserved memory where they can swap data from other areas. If they can get the data into that reserved area, then DOS can recognize it, and then maybe your program can use it. This is called exPANded memory, and it requires special hardware and special software. It also requires programs that have been written to take advantage of this memory.

WordStar cannot use exPANded memory, nor can any of my other programs (except 4DOS, which can use 80K). I had an exPANded memory board for a while, an Intel Above Board. I used its one megabyte of memory for a large disk cache. The Above Board uses the slower, 8-bit memory, and I sold it when I got the Inboard 386/PC with its 32-bit high speed memory. ExTENDED memory is that memory above one megabyte, also known as "high memory." This memory is of no use to anyone except those with 80386 computers using certain programs. Again, some clever programs can convert this exTENDED memory into exPANded mem-



ory so that it can be swapped into a window in the reserved area and then be used by some programs. Windows supposedly can use exTENDED memory as though it were conventional memory.

## Windows

What does Windows have to offer if you don't have a fast 80386 machine? What does it offer if you do? Tom Hart, President of the San Diego HUG, says: "Computer users downloading large files over a modem need to get to their word processors without having to wait around. They also need to easily transfer data directly from one program into another program." This is the promise of Windows, but Windows doesn't do this very well for me. Another program, DESQview 386, I found does do it!

My work with Windows so far has been in "real" mode. The present version of Windows 3.0 does not recognize the 80386 CPU in my Inboard 386/PC. A special version of Windows 3.0 for the Inboard 386/PC is scheduled for release in the next few months. I may try it. I can always add it to my collection of unused programs. Meanwhile, I tried Windows using the one megabyte of RAM on my Inboard, and I bought another 2 megabytes of memory and tried Windows again. I had reservations about buying the 2 megabyte piggyback board. It was expensive, and I wasn't sure if the extra memory could be used productively. Still, it was another toy I felt that I should have. How did Windows make use of that extra memory? It used it for a disk cache (using the Smartdrive utility program). With one megabyte of memory, Windows set up a small disk cache, and with 3 megabytes of memory, it set up a 1024 K size cache. That was it. A temporary subdirectory, installed on the hard drive, stored programs swapped during the switching.

All program swapping was to the hard drive. What this means is that a program's operation is suspended while it is saved to the hard drive. Switching back to an earlier swapped program resumes that program's operation while the current program gets suspended. There was no limit to the number of programs I could load, as long as there was empty space on the hard drive. The switching from program to program, however, was SLOW, SLOW, SLOW — even with the large disk cache. I found it faster to avoid Windows, to load one program at a time, and merely exit one program, load another one, exit the second program when done and reload the first one.

I tried using a RAM drive instead of the hard drive for program swapping. I reset the disk cache to 256 K size and installed a 1.5 meg. RAM drive in the remaining extended memory. I then set the temporary files area for the RAM drive instead of the hard drive. Now the switching of pro-

grams was instantaneous, but I found myself limited to two programs. Trying to load a third program gave a message that the drive was full, and everything stopped. With Windows, at least in "real" mode, you have to choose between capacity and speed. If you increase the capacity, everything slows down.

I also found that my programs didn't run as well when in Windows. For example, a feature of WordStar that I like is the ability to open a second document in a second window and to copy items from the one document into the other. Trying to do this from Windows gives an error message: "Not enough memory to open a second window." Windows steals too much conventional memory for its own uses.

What about running a communications program in the background? When I loaded my communications program, Mirror III, I couldn't switch out of it without crashing. The Windows communication program, "Terminal" should work but I couldn't try it. I had changed the IRQ number (the interrupt) on my serial comm port because of a conflict with my Kodak 6 meg. floppy disk drive. I gave the new IRQ address to Mirror III, and it works fine, but Windows is looking for my comm port at the old IRQ address and can't find it. So far, I have not found a way to give Windows the new address. What about transferring data from one program to another? Windows has a program called "Clipboard," which is not very flexible. With Clipboard, you can only capture an image of the entire screen for transferring. There is no way to select a part of the screen. I found this feature awkward to use. A much easier solution is to leave Windows and run the PC Magazine utility called "Snipper." A TSR program available on many bulletin boards, Snipper is called up with a hot-key Alt-W, and you can easily mark the part of the screen to save. Typing "P" will print the data, and typing "S" will save the data to a file (you will be prompted for a file name). If you use the "S" function, you can then open your second program and read in the saved file.

Am I sorry I bought the extra memory? No — because of DESQview 386. DESQview 386 is a combination of two programs: QEMM, a memory manager and DESQview, a window program.

## QEMM

When I bought the Inboard 386/PC, it came bundled with QEMM. I didn't have enough memory to take advantage of QEMM, and I put the program away. Now, with more memory, I took out this magical program. The first thing that QEMM did was to go into that reserved area of standard memory and add another 64K to my conventional memory. Conventional memory is pure gold — it can be used by ANY program. When I run

"Chkdsk," it now reports 704K of memory instead of 640K. QEMM saw that I didn't have EGA, and it took the half the 128K reserved for video memory that normally

Area	Size	Status
0000 - AFFF	704K	Conventional
B000 - BFFF	64K	Video
C000 - C7FF	32K	High RAM
C800 - C9FF	8K	ROM
CA00 - DFFF	88K	High RAM
E000 - EFFF	64K	Page Frame
F000 - FFFF	64K	ROM

Figure 1

would be used by EGA. (See Figure 1.)

WordStar soaks up much memory, and I am always on the ragged edge of having enough. I like to load the dictionaries and thesaurus into RAM so that they are immediately available. When I upgraded from MS-DOS 3.3+ to MS-DOS 4.0, I got an error message, not enough memory, when I tried to open a second document window in WordStar. I freed up some memory by reluctantly removing "Snipper" from my autoexec.bat file. Now with that extra 64K of memory, I not only put Snipper back, I also found that I could load WordStar's Dictionary Definitions into RAM as well. This allows me to check definitions as well as spelling and synonyms. This is the first time I have ever had enough memory to load the spell checker, thesaurus and dictionary definitions simultaneously.

The extra memory also solved a "multitasking" problem. When downloading a long file over the modem, it is frustrating waiting to use the computer. Mirror III can work in the background and free up the computer. Up to now, if I put Mirror III into the background and tried to load WordStar, I would get an error message: "Not enough memory." Now I have enough memory to load WordStar and get back to work while Mirror runs in the background. Windows also works better when run under QEMM. The increased available memory allows for more flexibility.

A second feature of QEMM is the "LOADHI" utility. LOADHI finds unused areas of the reserved memory area and allows you to load some of your TSR programs there. I used this feature to load my favorite disk caching program, VCACHE (Cache-AT), and the Logitech Mouse drivers (Mouse, Logimenu and Click). Figure 2 shows how these were loaded, and it shows that 66K of memory is still available.

## DESQview

A third feature of QEMM is that it looks at all that extra memory I purchased and makes it available to DESQview for program swapping. Swapping to a hard drive is slow. Swapping to a RAM drive is faster. And swapping directly into high memory

Area	Size	Status
C000 - C007	0K	Used (cache-at)
C008 - C00F	0K	Used (mouse)
C010 - C017	0K	Used (logimenu)
C018 - C01F	0K	Used (click)
C020 - C7FE	31K	Available
CA00 - CF6A	21K	Used (cache-at)
CF6B - D25C	11K	Used (mouse)
D25D - D590	12K	Used (logimenu)
D591 - D706	5K	Used (click)
D707 - DFFF	35K	Available

**Figure 2**

is the fastest. DESQview, also a window program, does believe that I have an 80386 CPU. It will load 4 or 5 of my programs (depending on size) and keep them all running each in a separate window. You can switch instantly from one program to another and back. When the high memory limit is reached, DESQview just starts swapping to disk. Programs swapped to disk are suspended in operation, but DESQview keeps on working. Unlike Windows 3.0, it does not crash or hang up.

I opened a DESQview window, loaded BASIC and wrote a BASIC program to

count numbers (X=X+1;Print X;etc.). I then opened a second window with WordStar and a third window with WordPerfect. I alternated between the two word processors while watching those numbers counting away in that third window. I also transferred data back and forth between the two word processors using DESQview's TRANSFER feature. With this feature, you mark the exact data you want to transfer, switch to your other program, point to where you want the data to go and then start the transfer. Data transferred seamlessly back and forth between WordStar and WordPerfect.

An exception to the program switching was my communications program. I could not switch out of MIRROR III without crashing. Fortunately I solved that problem by running Mirror separately and using its own ability to work in the background.

DESQview may not have Window's pretty face, but its beauty is soul deep. Even Jerry Pournelle, in the Byte Magazine IBM Special Edition for 1990 said: "I like Window's 3.0 well enough, but I find I'm more productive when I'm running DESQview." For me, both Windows and

DESQview are still toys. They are fun to play with, but they do not increase my productivity. QEMM, however, is a wonder. I cannot imagine ever giving that program up. No one with a 386 should be without it.

Products discussed:

QEMM-386  
DESQview  
Quarterdeck Office Systems  
150 Pico Blvd.  
Santa Monica, CA 90405  
(213)392-9851

Windows 3.0  
Microsoft  
21919 20th Avenue  
Bothell, WA 98041  
(206)454-2030

Vcache  
Golden Bow Systems  
P.O. Box 3039  
San Diego, CA 92103  
(619)483-0901



# New Products

P/N 885-6016

**Electronic Clavier ..... \$20.00**

**Introduction:** Electronic Clavier is a program that turns your keyboard into a keyboard — a dual manual polyphonic 7 octave music keyboard, that is. Electronic Clavier lets you play music on your computer, and it can record what you play and play it back later. If you spent all of the money you were saving up for a piano on your computer system — well, Electronic Clavier is not really as good as a piano, but if you can play a piano you should be able to make music with it.

**Requirements:** Electronic Clavier requires a PC-compatible computer (including all Zenith Data Systems models that run MS-DOS except the Z-100 series), MS-DOS version 2 or higher, and at least 64k of free program memory. Electronic Clavier plays music through the existing speaker in your computer, and requires no extra sound equipment.

**Program Author:** Patrick Swayne, ZUG Software Engineer.

**Program Content:** The Electronic Clavier disk contains these files:

**CLAVIER.DOC** — Instructions for using Electronic Clavier

**CLAVIER.COM** — The Electronic Clavier program. CLAVIER displays a chart on the screen that shows how the keyboard keys are "mapped" to piano keys. It is designed to be used with a keyboard having a Zenith Data Systems or Northgate style layout. The lower "manual" runs from the left shift key to the right shift key, including all keys between, with some of the keys in the row above used for the "black" keys. The upper manual runs from the Q key to the enter key, with the row above used for the black keys. The two manuals overlap a few notes, and together they cover 2 3/4 octaves. The Alt and Ctrl keys are used to lower or raise the starting note by one octave for each key press, and the full range of the instrument 7 3/4 octaves. CLAVIER plays polyphonic notes (two or more notes played at the same time) by playing the notes arpeggio style (in rapid

succession).

CLAVIER can store what you play in your computer's memory as you play it, and you can play back your stored music. It stores in "real time", which means that the song is played back at exactly the same tempo (and with exactly the same mistakes!) as when you played it. It can also save stored music to a disk file, and load files that have been saved.

**CHOPSTIK.CLV** — The piano song "chopsticks" as played by me (Pat Swayne). Keep in mind that I'm not a piano player as you listen to it.

**CLAVIER.ASM** — The assembly source code for CLAVIER.

**Call for Songs**

If you get Electronic Clavier and create some decent sounding song files, please send them in. If we collect some good stuff, we will distribute it some way (perhaps on the COM1 bulletin board).





# Powering Up

## Volume 2

William M. Adney  
P.O. Box 531655  
Grand Prairie, TX 75053-1655  
Copyright (C) 1990 by William M. Adney. All rights reserved.

## Powering Up the PROMPT Command

In several articles within this series, I have mentioned the PROMPT command and the various kinds of tasks you can perform with it. In some cases, I have presented an example of a PROMPT command to perform a specific task, such as displaying the drive and the subdirectory, but that is only one of the possibilities. The PROMPT command has so many features which are generally easy to use that it justifies an entire article. For example, you can change the colors on the CRT, and you can even set up some function keys to perform simple tasks, such as the DIR command or any other command you choose as illustrated in the last article.

As discussed in the last article, many of these features require that ANSI.SYS be installed first, so be sure that you have installed that device driver before you try any of the fancy things discussed here. If you find that something described here does not work on your system, be sure you check that ANSI.SYS is installed properly because the last three examples require it. Also, some versions of MS-DOS included a device driver that did not fully implement some of the usual ANSI.SYS features, especially version 3.10 (on the Z-100, too). You will need to have installed at least version 3.20 or later for some of these features to work.

Even if you have not worked with this kind of thing before, I encourage you to experiment a little to see how this command works. All you need to do is make sure you enter the examples exactly as shown, and once you see how the command works, then try your own variations and ideas. You can't break anything in the computer, even though it is possible to get some really unusual (or even weird) displays. With that said, let's begin by looking at how the PROMPT command is generally used.

### The PROMPT Command

The PROMPT command is normally used to change the form of the DOS command prompt display which is usually in the form

of "A>" or "C>". In this basic example, the drive designator is A or C followed by the "greater than" separator (>). If you want to change that to have a colon (:) instead, you can enter the command PROMPT \$N: to change it. To change back to the original (default) prompt, just enter the PROMPT command by itself. No matter how you change the actual display with the command, the PROMPT command by itself will always return you to the standard DOS prompt.

One nice feature of this command is that it is commonly referred to as a built-in, system (resident) or internal command. Those three terms mean that it is contained (i.e., part of) in the DOS Command Interpreter (COMMAND.COM). You can execute any built-in command at any time, regardless of the current drive indicated by the command prompt.

### The General Form of the PROMPT Command

The PROMPT command has two general

forms that we have already discussed as shown in Figure 1.

As previously discussed, the PROMPT command by itself simply returns you to the standard DOS prompt. When the command is followed by special prompt parameters (prompt-params) as shown in Figure 1, you can easily change the display to suit your individual needs.

Although the prompt-params generally may be entered in upper or lower case, all other characters are interpreted EXACTLY as typed. This is critical when you are working with the ANSI functions that are discussed later in this article. For example, you can clear the screen with an Escape [2], but Escape [2j] (note the lower case letter j) will not clear the screen because it is not recognized. In this article, I have used the upper case for the prompt-params in all examples because it makes the examples more readable and emphasizes the fact that some of the ANSI escape function characters must be in lower case.

All other characters not shown in Figure

**PROMPT** (Return to default prompt)  
**PROMPT prompt-params** (Enter parameters)

#### prompt-params

#### Description

\$B	Bar ( ) symbol
\$D	Date display for current system date
\$E	Escape character (equivalent to 27 decimal or 1B hex)
\$G	Greater than (>) symbol
\$H	BACKSPACE (erase previous character)
\$L	Less than (<) symbol
\$N	Current (default) drive letter.
\$P	Current (default) drive letter and directory name
\$Q	Equal sign displayed within prompt
\$S	Space displayed as first character (leading space)
\$T	Time display for current system time
\$V	Version number of the DOS (like the VER command)
\$ <u>c</u>	Null character <u>c</u> not listed above
\$	Underline character for a CRLF (same as ENTER) sequence
\$=	Equal sign to set values for prompt

**Note:** All prompt-params may also be entered in lower case except for the last two symbols.

Figure 1. PROMPT Command.

1 may also be used in the command, except that they perform no function. These characters, such as \$A, are useful in some situations and are called null characters. We will take a look at one use of the null characters in a later example.

One other restriction in the use of the PROMPT command (or any DOS command) is that it can be no longer than 128 characters. PROMPT has 6 letters which must be followed by a space for a total of 7 characters. In addition, all commands are followed by a CTRL-M character which is generated when you press the ENTER key. That is a total of 8 characters required for the command which leaves a maximum of 120 characters which can be used for the prompt parameters themselves. Similarly, the COPY command allows a total of 122 characters for file names and other parameters.

Since that presents a display problem when the command line is longer than 80 characters, you may need to use a "line feed" sequence on your terminal to extend it to the next line without sending it to the system. All you have to do is enter CTRL-J or CTRL-ENTER. When you have finished entering the complete command line, just press ENTER as usual to send the command to the system.

### An Easy Way to Test

One of the easiest ways to experiment with a this kind of command is to set it up in a batch file. That makes it easy to edit a complicated command, and you can save various versions of the command until you find one that you really like. I usually choose a simple file name, like T.BAT (or TEST.BAT), and later versions can be called something TEST1.BAT, TEST2.BAT, and so on.

If you decide to use the batch file trick, remember that the batch file must contain standard ASCII characters. You must use an editor that generates a standard ASCII file such as MS-DOS EDLIN, the "Text Out" feature in Word Perfect or the non-document mode of WordStar. Many word processors, such as WordPerfect, do not normally generate standard ASCII files — they add special characters which are used by the word processing software. EDLIN is an easy way to add and edit small files because it is supplied as part of all standard DOS operating systems.

### Current Drive and Directory

Using DOS subdirectories is quite convenient and required when the number of files exceeds the capacity of the root directory on any disk. When used properly, subdirectories allow you to organize various programs and data, but it is easy to forget which subdirectory you are in. The easy way around that problem, which has been discussed before in this series, is to use a PROMPT command. My personal preference is:

PROMPT \$P \$Q\$Q\$G

The \$P displays the current drive letter and directory in the form of d:\directory. The \$Q parameter represents an equal sign (=). And the \$G displays the greater than symbol (>). Note the space between the \$P and the remaining prompt parameters \$Q\$Q\$G. That space is inserted in the prompt to separate the drive and directory from the remainder of the prompt. This particular form of the command will display the command prompt in a form shown in Figure 2.

```
C:\ --> (ROOT directory)
C:\BATCH ==> (\BATCH subdirectory)
C:\DOS ==> (\DOS subdirectory)
```

Figure 2. Example PROMPT Display.

If you do not like the two equal signs before the greater than symbol, simply eliminate both of the \$Q parameters. Note again that I have shown a space between the drive/directory parameter (\$P) and the first equal sign (\$Q) so that I will have a space between the directory name and the first equal sign. This example demonstrates the importance of spaces as well as providing useful information on where you are.

### Adding to the PROMPT Command

You may find it useful to display the current system time and date, so we will add them to our PROMPT command. In addition, we will add some text (called a literal) to the prompt, which may be particularly useful if you are setting up a system for someone else. Consider the following command line:

```
PROMPT Time is$T on $D$$_Enter Command for
Drive $N$$_$Q$Q$G
```

Notice that I have intentionally omitted some spaces in this command line. I have purposely left them out on the time and date commands to show what the resulting display will be. You can improve the readability of the information by adding spaces before and after both equal signs in the command. If you use the command as shown, you will see a display similar to Figure 3.

```
Time is08:11:42.37 on Mon 11-5-90
Enter Command for Drive C
(Blank line)
==>_
```

Figure 3  
Adding Time and Date Plus Literals

Note the double "\$\$\_" (Carriage Return, Line Feed — CRLF) sequence following the \$N parameter. The first CRLF ends the line "Enter Command for Drive \$N". The second CRLF produces the blank line between the literal and the "==" prompt indicator. You can use this idea to improve the readability of your prompts.

### Using the Version Parameter

If you have a number of bootable floppy disks that were created by using the FOR-

MAT with the /S (system) option, you may need to know which version of the operating system you are using. This is especially true if you have received an update to MS-DOS and have not updated all disks with the new version. Adding the version parameter to the prompt will always inform you which version of the DOS that you are using. It is an easy way to display the information which is similar to the VER command.

Let's try another trick with that. Consider the following command line:

```
PROMPT $A.....Version is:
$_$_$V$_$_$P $Q$Q$G
```

In this example, I have used a valid null character, \$A, followed by a number of periods. The eight periods (.) are used to indicate spaces (i.e., SPACE BAR input) for clarity and should be omitted when you enter the command. I could have also used the \$\$ (space) parameter to accomplish the same thing. The purpose of the spaces is to right justify the literal "Version is:" on the screen. The null command is useful (and in fact required) in this case since we do not have any other value to input immediately following the PROMPT command. If we simply entered spaces, without the null character, the command would only skip the spaces until it found the first character or command parameter. Interestingly enough, the \$\$ is actually a null command, which is why it works just like any other null command, such as the \$A that I've used in the example. Figure 4 shows the display results.

```
Version is:
(Blank line)
MS-DOS Version 4.01
BIOS Version 4.00.02
(Blank line)
C:\directory ==>
```

Figure 4  
Using the Version (\$V) Parameter

With a few exceptions, we have covered the most common of the parameters that can be used with the PROMPT command. The parameters for the bar symbol (\$B) and the less than symbol (\$L) are used just like the \$Q (equal sign) and \$G (greater than) symbols that we have already used. You may find the \$H (backspace) parameter useful if you want to "erase" the hundredths of a second that are not much use in a time display with the \$T parameter.

For those of you waiting to find out how to do magic things with the PROMPT command, like changing display colors, I have saved that for last.

### Adding to CONFIG.SYS

Before you can use any of the ANSI functions on your computer, you must add a command line to the CONFIG.SYS file. This same example was used in the last article, but I am including it again because it is essential to using the ANSI functions.



Figure 5 shows an example of installing ANSI.SYS in CONFIG.SYS.

In this example, the ANSI.SYS device driver is located in the \DOS subdirectory

```
FILES=25
BUFFERS=30
DEVICE=C:\DOS\ANSI.SYS
```

**Figure 5. Implementing ANSI.SYS.**

on drive C, and you will have to change the path if your system is different. The FILES= and BUFFERS= commands shown in this example are included only for a complete CONFIG.SYS file listing, and the values required for your system may be different. Remember that you must reboot the system after adding a new command to CONFIG.SYS so that it will be installed. Adding this information to CONFIG.SYS should not interfere with any of your existing software, but if it does, all you have to do is rename the file (e.g. CONFIG.XXX) and reboot your system. Also remember that CONFIG.SYS is an ASCII file that must be edited (or created) with an appropriate editor. EDLIN is one way to edit or create an ASCII file. You can also use WordStar's non-document mode or the "Text Out" function in Word Perfect.

### Using ANSI Functions

A complete list of the ANSI functions was included in the last article. However, you must substitute the \$E prompt parameter for the ESC in order to use those functions with the PROMPT command. To illustrate how to use some of these ANSI functions with the PROMPT command, this article includes specific examples of selected functions that I think you will find interesting and informative.

Please be sure that you are using the appropriate value for all escape sequences. As previously mentioned, either \$E or \$e can be used as the escape character, but you must use the correct uppercase or lowercase character for the function as shown.

### Developing Custom PROMPT Commands

It is easy to develop a custom PROMPT command for your system. You need to remember that working with the PROMPT command like this is really a form of programming, which means that everything must be EXACTLY correct or the command will not work as you expected.

I have selected three examples to illustrate how to use the PROMPT command with ANSI.SYS. The first example simply clears the screen and displays the command prompt after any command is entered. The second example displays the command prompt in reverse video (i.e., black letters on a white background). And the third example displays the command prompt in reverse video and changes the rest of the screen display to white letters on a blue background. Of course you can also

combine any of the ideas in these examples with some of the PROMPT parameters presented earlier in this article for a very unique command prompt.

### Clearing the Screen

I selected this particular example because it is one of the simplest that I could imagine. When I tested this example on my system, I called it TEST1.BAT, which included the lines shown in Figure 6.

```
REM Clear the screen and set the command prompt
REM Note that ANSI.SYS must be installed in CONFIG.SYS first
PROMPT $E[2J$P $Q$Q$G
```

**Figure 6. Clearing the Screen.**

The REM command includes descriptive comments about what the PROMPT command does, and I recommend including these kinds of comments in any batch file so you will be able to figure out why you included a command a year from now. In this example, there are only two parts to the PROMPT command. The first part is the \$E[2] ANSI function which clears the screen (see the previous article for all ANSI functions), and the second part is the \$P \$Q\$Q\$G form of the command prompt that was mentioned earlier in this article. As an interesting experiment, try reversing the order of these two parts of the example; that is, try the PROMPT \$P \$Q\$Q\$G\$E[2] command. Very clever! Now the command prompt has totally disappeared and the only thing displayed on the screen is the cursor. Obviously there is a bug in the PROMPT command, right? Wrong!

There is no bug in the PROMPT command; rather, it was a bug in the logic used for this experiment. Consider what we did. In the first (and correct) example, we used the PROMPT command to first clear the screen, then display the command prompt. In the experiment, we use the PROMPT command to first display the command prompt and THEN clear the screen. Even though the command prompt was correctly set, clearing the screen after that completely wiped out what we had done. The point is that the order of these two parts of the PROMPT command is absolutely essential to its performing its expected design purpose.

I chose this example to illustrate how important it is to consider what you want to do BEFORE you enter a command to a computer. A computer is really a "dumb" machine, and it only does what humans tell it to. And if you tell it to perform a process (or program steps) in the wrong order, a computer will blindly do what you tell it without checking to see if it makes sense.

This particular example (and experiment) was chosen because it was simple and could be used to make a point. I agree that the example is not useful in practice because it clears the screen after any command is entered, which means that you won't get much of a chance to see a display

from the DIR or CHKDSK commands for example.

If you performed the experiment discussed, you are probably wondering how to get some kind of command prompt to display at this point. Entering the PROMPT command by itself resets the command prompt to the default of C> as discussed earlier in this article. Now that you see how this works, let's try some more useful examples.

### Using Reverse Video

Reverse video (sometimes called highlighting) is used to make something easier to see. Even if you have a color monitor, you will find that the DOS default is to display white characters (in the foreground) on a black background. You can reverse the foreground and background colors using an ANSI mode function (see Figure 3 in the previous article) so you see black characters on a white background. I used a batch file called TEST2.bat for this example, which includes the lines shown in Figure 7.

The PROMPT command in Figure 7 con-

```
REM Use reverse video for the C:\DIR
== prompt
PROMPT $E[7m$P $Q$Q$G$E[0m
```

**Figure 7. Using Reverse Video.**

sists of three parts: \$E[7m, \$P \$Q\$Q\$G, and \$E[0m. The first part is a Set Graphics Rendition (SGR) function listed in Figure 3 in the preceding article, where the 7 is a numeric parameter that turns on the reverse video. The second part of the command is my preferred form of the command prompt. And the third part of the command is also an SGR function, where the 0 (zero) turns all attributes, including reverse video, off. When the command executes, it turns on reverse video, sets the command prompt, and then turns off reverse video. The end result is that only the command prompt is displayed in reverse video, and notice that there are no spaces in the two SGR functions that are used in this PROMPT command.

You can also display any of the command prompts described earlier in this article in reverse video by "enclosing" it with these two SGR functions. As an experiment, you might want to try omitting the second SGR function that turns off the reverse video. Even if you don't want to actually try that, take a moment to consider what would happen. (Hint: What happens if you turn reverse video on and never turn it off?)

### Changing Screen Colors

No doubt about it, the default white characters on a black background is kind

## Quality Enhancements!

### EaZy PC Products

**EZM-128:** Expand 512K base memory to 640K. Simple, plug-in installation. \$125.00

**EZCLOCK:** Calendar/Clock. Piggy-back add-on for EZM-128. \$35.00

### No Slot Clock/Calendar

**FBE SmartWatch:** Automatic date/time on bootup. Installs under BIOS/Monitor ROM. Ten year battery. Works with all Heath/Zenith MSDOS computers. For PC's \$32.00, Z-100 \$33.00 Module: \$25.00

### H/Z-148 Expansions

**ZEX-148:** Adds one full-size and one half-size expansion card slot. \$79.95

**ZP-148:** PAL chip expands existing 640K memory to 704K. CGA/MDA only! \$19.95

### Configuration Control

**CONFIG MASTER:** Menu-select active CONFIG.SYS during bootup. Software for PC/Z-100 MSDOS. \$29.95

### H/Z-150 Items (Not for '157, '158, '159)

**VCE-150:** Eliminate video card. Install EGA or VGA card. All plug in. Includes circuit board, SRAM and RM-150. \$49.95

**ZP640 PLUS:** PAL chip to expand standard memory card to 640/704K with 2 banks of 256K RAM chips (not included). \$19.95

**ULTRA-PAL:** Three PAL chips: MR150 for 704K + 512K RAM Disk; MR150T for 640K + 512K RAM Disk; LIM150 for 640K + 512K (32 pages) of simulated v3.2 Lotus/Intel/Microsoft Expanded Memory. With software. Install on standard memory card. No soldering. Needs 45 256K RAM chips (not included) for maximum memory configuration. \$39.95

**COM3:** Change existing COM2 to COM3. Put internal MODEM at COM2. Don't lose serial port. With software. \$29.95

### H/Z-100 Modifications

**ZMF100A:** Expand "old" motherboard (p/n 181-4917 or less) using 256K RAM chips (not included). No soldering. \$65.00

**ZRAM-205:** Put 256K RAM chips on your Z-205 board. Get 256K plus 768K RAM disk. Contact us for data sheet before ordering. Without RAM chips. \$39.00

### H/Z-89 Add-Ons

**H89PIP:** Parallel printer 2 port interface card. With software. \$50.00 Cable \$24.00

**SLOT4:** Add fourth expansion slot to right-side accessory bus. \$39.95

Order by mail, FAX, telephone, or see your dealer. UPS/APO/FPO shipping included. VISA/MasterCard. WA residents add 8.1% tax. Hours: M-F 9-5 PST. We return all calls left on our answering machine!

# FBE

FBE Research Company, Inc.

P.O. Box 68234, Seattle, WA 98168

206-246-9815 Voice/FAX TouchTone Selectable

Reader Service #104

of drab and common, so let's fix that as the final example and retain the command prompt in reverse video that we just figured out. I used a file called TEST3.BAT for this example which contains the lines shown in Figure 8.

The PROMPT command shown in Figure 8 contains four parts: \$E[7m, \$P \$Q\$Q\$G, \$E[0m, and \$E[37;44m. The first three parts

```
REM Use reverse video for the C:\DIR == prompt, and
REM set a white foreground on a blue background
PROMPT $E[7m$P $Q$Q$G$E[0m$E[37;44m
```

Figure 8  
Changing Screen Colors

are identical to the command example shown in Figure 7, so they need not be discussed again. The fourth part is just another use of the SGR function (see Figure 3 in the previous article) which sets the foreground and the background colors. In this example, the 37 represents a white foreground (characters), and the 44 represents a blue background.

In this example, it is especially critical to keep each part of the command in the proper order. If you carefully examine the PROMPT command line in Figure 8, you will see that reverse video is turned on, the command prompt is set, reverse video is turned off, and the colors are set for the remainder of the screen (and are never turned off). There are several different kinds of things you can do with this example, and I'll leave that to your imagination.

### Out of Environment Space

Although it has been discussed before in this series, you may see an *Out of environment space* error message as you are working with the PROMPT command. You can see the contents of the environment space by entering the SET command. The default size of the environment space is 160 bytes in most reasonably current DOS versions, and the error message just means there is no more space available to store the PROMPT command values you have entered. Fortunately, that problem is easy to fix, and you need to add a command line to your CONFIG.SYS file that looks like:

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

In this example, the SHELL= command looks in the root directory on drive C to find the "shell" (i.e., Command Interpreter) COMMAND.COM. The /E (Environment)

switch expands the environment space to 256 bytes, and the /P (Permanent) loads this as a permanent configuration.

### The Last PROMPT

When you finally develop the PROMPT command that suits your needs, I suggest that you include it in your AUTOEXEC.BAT file so that it will automatically be set up

when you start your system. That file also must be in ASCII format as discussed earlier. I suggest that you use the following format:

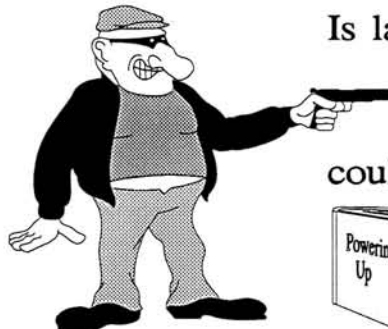
```
REM Set the command prompt to C:\DIR ==
PROMPT $N $Q$Q$G
```

One interesting problem is that you cannot use either the greater than (>) or less than (<) symbol in the REM to actually describe a command prompt because DOS apparently thinks that including either symbol means you want I/O redirection. That's the reason that the greater than (>) symbol was intentionally omitted from the above format. From what I can tell, DOS performs some strange parsing (examining) of the command line, and despite the fact that the REM command is at the beginning of the line at it should be, DOS still tries to perform I/O redirection. All examples presented in this article were developed and tested on a Z-386/16 using MS-DOS 4.0, although they should work on any compatible with these functions correctly implemented in ANSI.SYS.

### Powering Down

This concludes the *Powering Up* - Volume 2 series of articles that I hope you have found useful. By popular demand, I have been working on an outline for another *Powering Up* series of articles that I think you will find at least as useful as the first two sets.

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



Is lack of computer knowledge holding you up?

"Powering Up" could be your best protection!

"Powering Up" by Bill Adney

Order today!

ZUG (616) 982-3463



---

---

# Two Drive "B's"

**George P. Elwood**  
**1670 N. Laddie Court**  
**Beavercreek, OH 45432**

Along with the software work I do, I also have an interest in computer hardware. I have written a couple of articles for RE-Mark on hardware modifications you could make for the Z-100. Along with my trusty Z-100, I also have an H-386 so I get to beat on PC based hardware also.

This project will be of interest to those of you in the Department of Defense (DoD) with Z-248s, but can be used by anybody who wishes to add an internal tape drive to their computer.

Zenith Data Systems was the winner of the Air Force Desk Top Two contract awarded several years ago. This contract ran out in 1989. Zenith provided a special version of their Z-248 computer to many users throughout the DoD under this contract. These computers were unique in that the disk controller supported three floppy disk drives. Actually, the third disk drive was for an external tape backup unit. If you purchased this option you received a replacement back plate with a new connector, and a "Y" cable to divide the power supply. The new connector ran to the new back plate and was used to power the

external tape drive. The tape drive box contained an Irwin 20/40 Meg tape drive. This unit had two cables which were used to connect it to the Z-248. One of these cables connected to the new back plate for power and the other connected to the special disk controller 50-pin "D" connector on the back of the computer.

This system worked great and the installation did not take long. The problem occurred when some users purchased Z-248s with the tape backup at the end of the contract. Zenith Data Systems was out of the special Z-248s and replaced them with the newer commercial Z-248/12. This was a much better deal for those receiving the computers except that you could not use the tape backup units. The new disk controller cards only supported the usual two hard disks and two floppy disk drives. The office where I work as a contractor received six computers and tape backup units under this upgrade program. We now had six tape drives and no way to use them.

These new computers also came with 1.44Meg 3.5 inch floppy drives. Again, something that was not compatible with

the rest of the Z-248s and Z-100 in the office. We did have some of the external 5.25 inch, 360K floppy disk drives from the Z-100 contract which were not being used. We added these units to the Z-248s so that disks could be transferred between machines. This was no big problem as the old drives slipped right in the drive cage. The tape backup units were put aside for the time being.

After the loss of some data on a system, the question of tape backup units came up. Talking with Irwin, they suggested purchasing another disk controller card so that the tape drives could be used. I asked about doing this modification and they did not recommend it but they did send me the data cable layout. Because of the usual funding constraints, buying the extra card was not an acceptable option. I had done some work in the past on the Z-100 where I built a cable which permitted using the high capacity 5.25 inch floppy on the 8 inch drive connector. Checking the Z-100 documentation, I felt that it would be possible to "share" the "B" drive between the floppy and the tape drive. By cutting

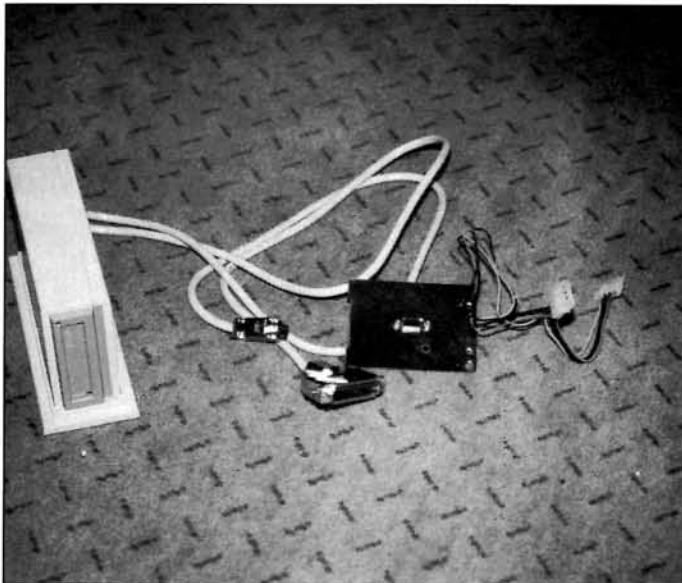


Figure 1. Basic Irwin Drive.

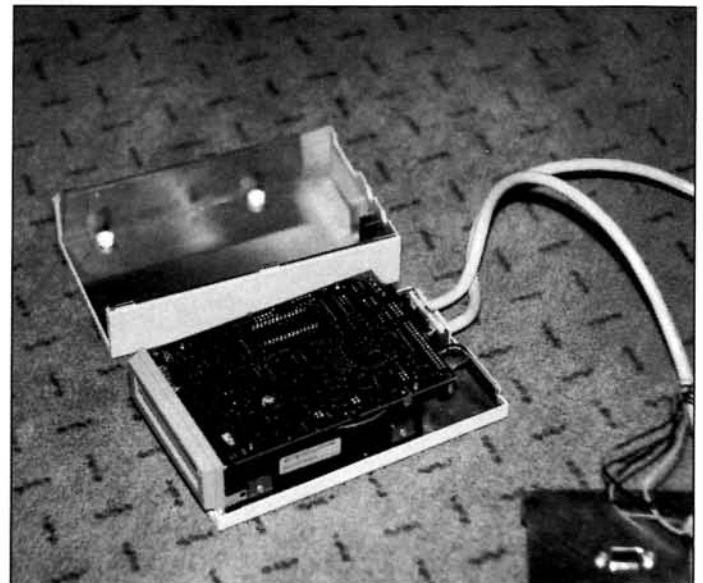


Figure 2. Basic Irwin Drive Opened Up.

the drive and motor select line on the flat ribbon cable and putting a double-pole, double-throw toggle switch in the line, either unit could be selected.

For the first system, I built a completely new cable. Because I had the 5.25" 360K floppy and the 3.5" 1.44Meg floppy, I had to install three connectors. The 360K drive needed an edge card connector while the other two drives (tape backup and 3.5 inch floppy) needed 34-pin headers. I found that you must cut and insert the twist for drive "A". I cut "B" drive select (line 12) and "B" motor select (line 16) between the 34-pin header and the first floppy connector (edge card for drive B"). I cut the same lines between the edge card connector and the first header. **Note:** You do not have to cut the ground lines associated with the

data lines. Only cut the data lines shown.

I cut a piece of six conductor flat cable about 15 inches long to run from the data cable to the toggle switch. I inserted a piece of shrink tubing on each of the lines before soldering them together. This tubing was then slipped down and heated to cover the connection, protecting it from a short. Both lines were soldered along with the two for the first connector. The two lines between the first and second connectors were again soldered together. Make sure you track the lines or this mode won't work. I use the first, third and fifth lines for drive select line 12. Line two, four and six were used for line 16, motor select. See Figure 3.

The 6-conductor line is then connected to a DPDT toggle switch. Lines one and

two go to the center points, three and four to one side, and five and six to the other side. Again, make sure that lines one, three and five are on one edge and the other lines are on the opposite side. Again, I inserted a short piece of shrink tubing on the cable before soldering the cable to the switch. This was then slipped down to cover the connections. I had an extra back plate from a PC which I used to mount the toggle switch. Make sure you insert the handle facing the same way as the top mounting bracket. If you do not have an open slot in the computer you will have to mount the switch someplace else. The large removable plate on the back would be another location. I installed the toggle switch so that with the handle up the top drive would be selected. Moving the switch handle down selected the bottom drive.

Next I took the external tape drive enclosure apart. This unit just snaps together, no screws. See Figure 2. I removed the drive from the case. The drive is a 3.5 inch unit and will not mount in the computer without a bracket. I purchased a mounting bracket for this first test unit. Mounting the unit was easy with the provided screws. You must remove the termination resistor from the tape drive and move the drive jumper to the "B" position. This is easier if you do it before mounting the unit in the bracket.

I now opened one of the Z-248/12 computer and installed the tape unit under the 5.25 inch floppy. It is easier to install the cable and power line before you install the complete cage in the computer. I installed the Irwin software (v1.14) and was ready to try the system. Because the Z-248 I use has a 40 Meg drive installed as one partition, the Irwin software will not recognize the hard drive. I had used an Irwin tape drive before with a large hard drive and knew that the following line is required in

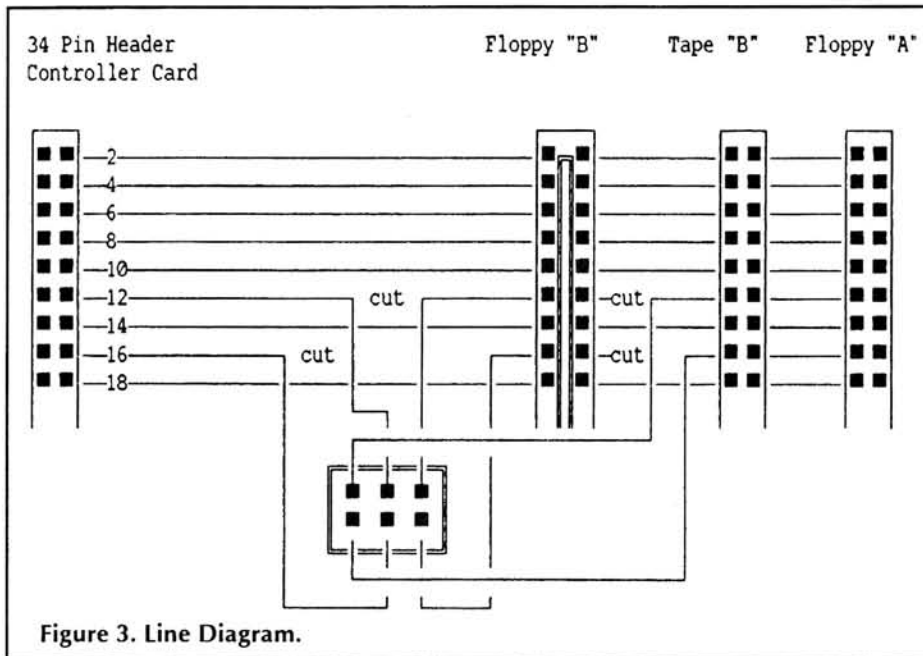


Figure 3. Line Diagram.

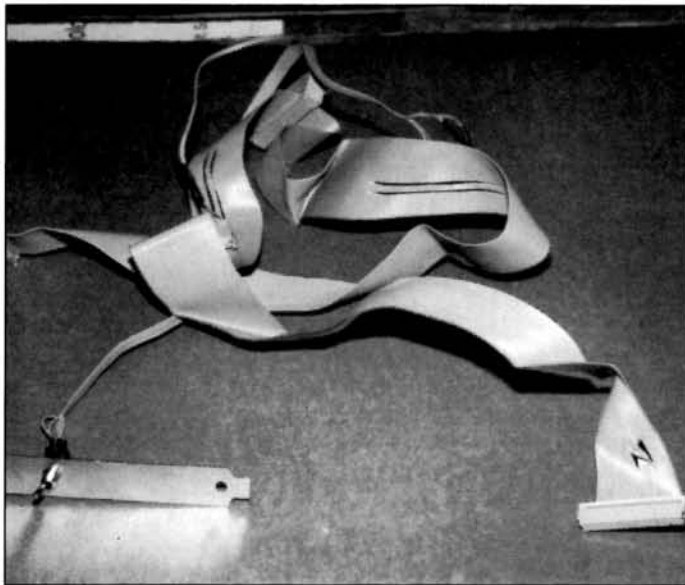


Figure 4. Completed Cable Assembly.

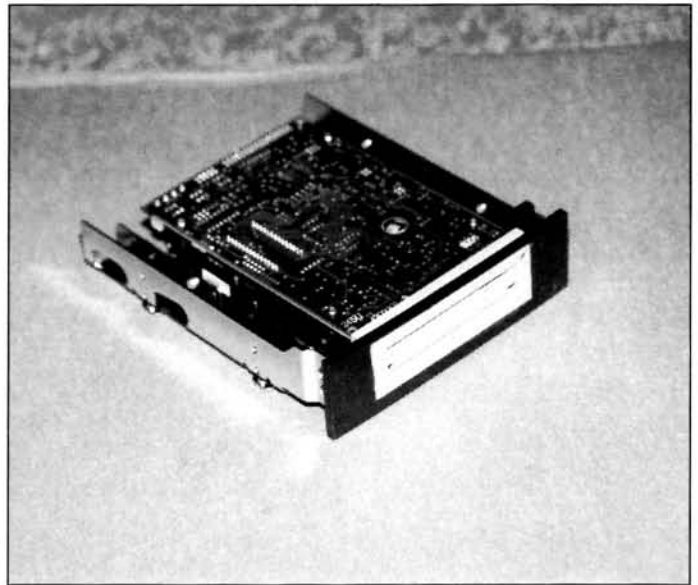


Figure 5. Tape Drive in 5.25" Bracket.



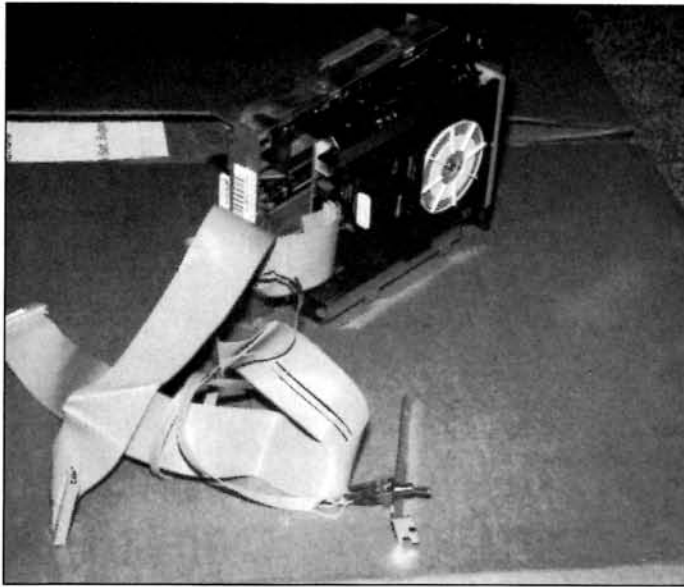


Figure 6. Cable, Floppy and Tape Assembly.

your AUTOEXEC.BAT file.

```
set ez_ddd=1
```

Once this line was installed and the system rebooted, I was able to backup the hard drive on tape by moving the toggle switch down. To use the 5.25 inch floppy the toggle switch is moved up. Since I had to buy the cable, connectors and toggle switch, the cost was slightly under \$10 plus

adaptor bracket.

I was able to find the 3.5" to 5.25" adaptor in a local electronics surplus store. These were for hard drives and had the front cover on them. I was able to cut out part of the front so that the tape drive fit through making it look good, almost like it was made for it. This drive was installed in a Z-386/25. The system would not work ini-

the cost of the adaptor bracket.

Now that I had a system that worked, I took the cable I removed from this first computer and used it for the next system. Luckily, Zenith's cable had three drive plugs on the cable. They had a 34-connector edge card, a 34-pin header, and another 34-pin header for drive "A" with the twist. This is exactly like the cable I had to build for the first system so my total cost was \$1.25 for the toggle switch, plus the cost of the

tially because of the speed of the computer. The Irwin tape drive software (v1.14) will not support the high speed of the Z-386/25. Using the MACHINE SPEED SLOW command I was able to slow the computer to 8 MHz and the tape system worked OK. This command may not be available in all versions of MS-DOS and is only for Z-386 computers. When using PC Tools Version 6.0, you do not have to reduce the speed of the computer. It will run at the higher speeds. PC Tools will only work with a 40Meg tape loaded in the tape drive. It uses a compression routine that will increase the capacity depending on the applications. (You may have to use the SETUP capability of the Zenith Data Systems computer if you don't have this command or the you don't have a Z-386.) I then installed it in the lower position in the drive chassis, and installed the chassis in the computer. I then heard a very loud noise and sometime the tape drive would not response. The problem was tracked to the mounting bracket I used. The tape motor flywheel was hitting the computer case. I moved the tape to the upper position in the cage and that fixed the problem.

This simple project permits you to use a tape drive without having to purchase another controller card. It will save you money and increase the capability of your computer. Have fun. ✨



Figure 7. Installed in Computer.

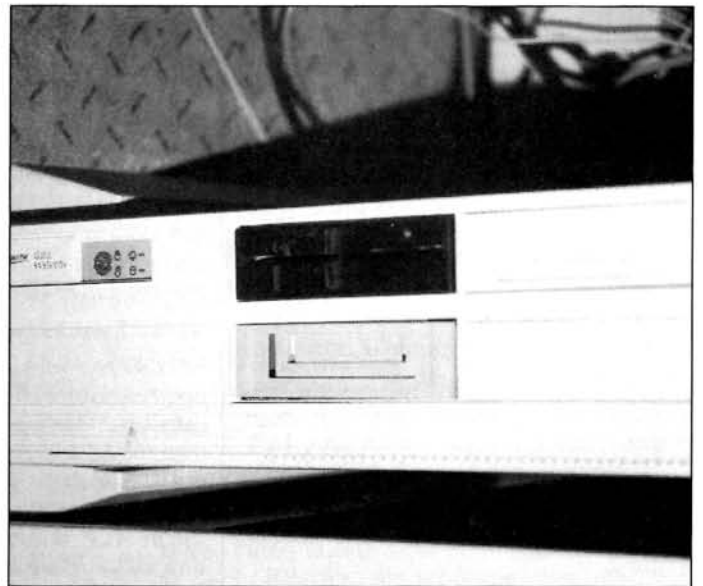
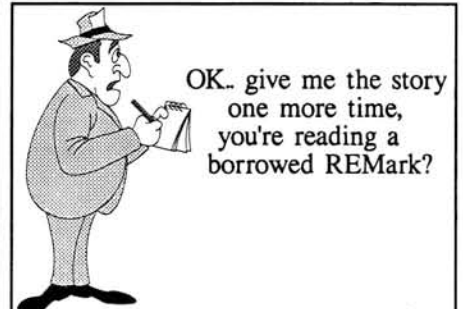


Figure 8. Front of Computer.

All Checks must be made out to  
Zenith Data Systems

_____	January 1, 1991	_____
_____	Zenith Data Systems	00.00
_____	Zero dollars and zero/100	_____
<b>S A M P L E</b>		





## The other cats get to sing along!

That's because HEPCAT runs **with** your other programs, not **over** them. HEPCAT (HUG Engineer's and Programmer's **C**Alculation Tool) is a powerful pop-up calculator for all Heath/Zenith MS-DOS and Z-DOS based computers. Unlike other pop-up calculators, HEPCAT does not stop the currently running program while it is popped up. That means that you can do calculations while your computer is busy with something else. For example:

- While Lotus (tm) is loading a huge spreadsheet, you can check your kid's math homework.
- While Dbase (tm) is sorting a large database, you can add up some grocery prices.
- While your computer is busy compiling one program, you can work on number base conversions needed for another program.

HEPCAT is safe to pop-up during just about any running program — even during disk activity. And HEPCAT has other features the other guys can't touch.

### HEPCAT gets along with everyone . . .

HEPCAT supports more video configurations than any other pop-up, and always

pops up in the current video mode, rather than forcing the screen into a text mode as other pop-ups do. It also works properly with more programs than any other pop-up. You can pop up HEPCAT over Microsoft Windows (tm) and many other programs that other pop-ups can't work with, and even over some other pop-ups.

### HEPCAT works harder . . .

HEPCAT provides a multi-function floating point calculator and a programmer's binary calculator that work together to do more than the basic four (+, -, \*, /). The floating point calculator includes the following built-in functions: powers, pi, factorial, square root, sine, arc sine, cosine, arc cosine, tangent, arc tangent, log (natural and base 10),  $e^X$  and  $10^X$ . It also includes the following conversions: degrees-radians, radians-degrees, Celsius-Fahrenheit, Fahrenheit-Celsius, centimeters-inches, inches-centimeters, meters-feet, feet-meters, kilometers-miles, miles-kilometers, grams-ounces, ounces-grams, kilograms-pounds, pounds-kilograms, milliliters-fluid ounces, fluid ounces-milliliters, liters-quarts, quarts-liters. The binary calculator works in these number bases: binary, tetral (base 4), octal, split octal, decimal, and hexadecimal; and it supports

these operations: MOD, AND, OR, XOR, SHL, SHR.

The HEPCAT floating point calculator supports 8 significant digits and can display numbers four ways: floating point, fixed point, scientific notation, and engineering notation. Numbers are handled internally in BCD format to eliminate binary round off errors in addition and subtraction.

### HEPCAT eats less . . .

HEPCAT uses less than 18k of memory — less than any other pop-up calculator that we know of. It also uses less than 14k of disk space, so you don't have to worry about where to put it on a small system. The HEPCAT window uses less screen space, too. It shows you more real information than other pop-up calculator displays, but it doesn't waste space by showing you a keypad layout. You already know what your keypad looks like! HEPCAT is easier to learn, too, with commands that make sense.

If you are tired of pop-ups that can only sing solo, give HEPCAT a try. HEPCAT is available from HUG as part no. 885-3045-37 for \$35.00. It works on any Z-100 PC, Z-200 PC, or Z-100 (not PC) system and any version of MS-DOS or Z-DOS.



# dBASE III

## Part 8

# Custom Reports

D. R. Cool  
7421 Troy Manor Road  
Dayton, OH 45424

The one area that dBASE III fails in flexibility of application is the report generator. The limitations are severe enough to warrant writing your own report-generating program. For example, with the report generator, you have little control over the format of the printed page. Location of date and page number is fixed and you cannot print more than four lines of page or column titles. An even more serious limitation is that you cannot extract data from more than one data base. In this article, I will show how to overcome these limitations using the dBASE III programming language.

A first-of-the-month report on projects completed during the previous month is required by management. This report must contain a listing of all completed projects, listing the drawing number, the revision letter, the device type and vendor part number, the project number and document date. The end of the report must contain a summary of the total number of new documents and the total number of revised documents. Also, the summary must include a break-out of total projects per engineer.

In order to provide enough data for at least a two page report, we will first add to the projects data base. If you still have the original PROJECTS data from Part 2, it should contain 24 records. Rather than add records manually, the data base can be doubled in size with a few commands. At the dot prompt, type the following commands:

```
USE PROJECTS
REPL ALL APPDATE WITH CTOD('01/10/90')
REPL ALL DOCDATE WITH CTOD('01/15/90')
COPY TO PRTEMP
```

All projects now have an approval date

of 01/10/90 and a document date of 01/15/90. Also, the entire file has been copied to a second file called PRTEMP.

Next, the records in PRTEMP should be changed to distinguish them from the original records:

```
USE PRTEMP
REPL ALL DWGNR WITH LEFT(DWGNR,5)+"89"+RIGHT(DWGNR,3)
REPL ALL PROJNR WITH LEFT(PROJNR,7)+"3"+SUBSTR(PROJNR,9,2)
```

Next, corresponding records must be created for the PROJXREF data base:

```
COPY FIELDS PROJNR TO XRTEMP
MODI STRU
(Add fields: VENDPN, character, width 12
DEVTYPE, character, width 2
REPL ALL DEVTYPE WITH "01"
REPL ALL VENDPN WITH "MADEUP")
```

Finally, we can append the temporary files to the PROJECTS and PROJXREF data bases:

```
USE PROJECTS INDEX PROJDRVR
APPEND FROM PRTEMP
REINDEX
USE PROJXREF INDEX PROJXPJN
APPEND FROM XRTEMP
REINDEX
```

We wish to make one final correction to the project records. The original data base used REV = "N" for new. This should be replaced with a blank:

```
USE PROJECT
REPL REV WITH " " FOR REV = "N"
```

Before printing the report, it will make the programming easier by first collecting all of the necessary data and storing it into a temporary data file. Remember that the project data and device type data are stored in two separate data base files and are related only by the project number. Fortunately, dBASE III provides an easy way to combine data from two separate data bases using the JOIN command. Syntax for the JOIN command is as follows:

```
JOIN WITH <alias> TO <file name> FOR
<condition> | FIELDS <field list>
```

Prior to issuing the JOIN command, all projects with an approval date greater than or equal to a specified start date and less than a specified stop date will be copied to a temporary file called

TEMPDATA. Using the following JOIN command, a file called ACTIONS is created:

```
JOIN WITH PROJXREF TO ACTIONS FOR PROJNR
= PROJXREF->PROJNR
```

The condition for the JOIN command is FOR PROJNR = PROJXREF->PROJNR

(The "arrow" consists of the dash followed by the "greater than" symbol.) What this says is - for the PROJNR in PROJECTS, join this record with the records in PROJXREF with the same PROJNR. Here is how the JOIN command works. The record pointer is set to the first record in TEMPDATA. Each record in PROJXREF is evaluated for the FOR condition (matching PROJNR). If the specified condition is true, a new record is added to ACTIONS. Since no field list is specified, each new record contains all of the fields of PROJECTS plus the fields of PROJXREF. When all records in PROJREF are scanned, the record pointer in TEMPDATA advances one record and the process is repeated. This continues until all records in TEMPDATA are processed. Obviously, this process can be very time consuming for large files.

To illustrate the difference between a report created with the dBASE III report generator and a custom program, we will first create the monthly report using the REPORT FORM command. First, we must

JOIN the project data with the device type data. At the dot prompt, type the following commands:

```
SELECT 1
USE PROJXREF
SELECT 2
USE PROJECTS
JOIN WITH PROJXREF TO ACTIONS FOR
PROJNR=PROJXREF->PROJNR
```

This creates a new file called ACTIONS which contains all of the project and device type data in one file. Now we will create the report form. First we have to tell dBASE III what data base will be used for the report to draw its data from:

```
USE ACTIONS
```

Also, we want the report to list the drawing numbers in ascending sequence, so we will create an index:

```
INDEX ON DWGNR TO ACDN
```

Now we are ready to create the report form. At the dot prompt type "CREATE REPORT PROJECTS". (The name of the report form is "PROJECTS".) The screen will display a menu followed by a page heading block. In the first line of the page heading block, type "Drawing Completion Report for January, 1990". Leave the next three lines blank. Press ENTER until you get to "Left margin" and change the default of 8 to 3. Now, press PgDn twice, bypassing the "subtotals" menu. For Field 1 contents, type "DWGNR". For Field 1 header, line 1, type "DRAWING". For line 2, type "NUMBER", and for line 3, type 13 dashes (-). Use the default values for "width", "# dec. places" and "Total". For the remaining fields, use the following:

```
Field 2 contents: "space(3) + DEVTYPE"
header, line 1: " DEVICE"
header, line 2: " TYPE"
header, line 3: "-----" (12 dashes)
```

```
Field 3 contents: "VENDPN"
header, line 1: " VENDOR PN"
header, line 2: (leave blank)
header, line 3: "-----" (17 dashes)
```

```
Field 4 contents: "space(1) + REV"
header, line 1: "REV"
header, line 2: (leave blank)
header, line 3: "-"
```

```
Field 5 contents: "PROJNR"
header, line 1: " PROJECT"
header, line 2: " NUMBER"
header, line 3: "-----" (12 dashes)
```

```
Field 6 contents: "DOCDATE"
header, line 1: "DOCUMENT"
header, line 2: " DATE"
header, line 3: "-----" (8 dashes)
```

On all fields, use the default values for width.

After completing field 6, press CTRL-END. This writes the file to disk with the extension ".FRM". We are now ready to use the report form to generate the monthly report. At the dot prompt, type

```
REPORT FORM PROJECTS NOJECT TO PRINT
```

The NOJECT option suppresses the initial

form feed, causing the report to print on the first page that comes up in the printer. Note that the report is also echoed to the screen. After issuing this command, you should have a two-page report with page numbers and the system date in the upper left corner of each page. If you want to suppress the system-generated page numbers and system date, you can include the word "PLAIN" in the REPORT command. However, if you do this, the system will ignore the number of lines per page and simply print continuously. Also, if you wanted a continuous line under the heading, the report generator allows no way to do this since it always inserts a space between columns. Finally, the report form does not calculate and print the summary information required at the end of the report.

All of these deficiencies can be corrected by writing a custom program to generate the monthly report. This REPORT program is shown in Listing 1. As usual, do not type the line numbers. The program is written as a procedure to be added to the PROJPROC procedure file. Of course, it can be written as a stand-alone program by simply deleting line 1.

Lines 6-12 get input from the user for the start and stop dates. When you run the report, use 01/01/90 as the start date and 02/01/90 as the stop date.

Line 16 copies all records for which the approval date is greater than or equal to the start date and less than the stop date. Since we made a global change of the approval date of all projects to 01/10/90, all 48 projects will be copied to the TEMPDATA file. Obviously, in a real system, only a small percentage of active projects would be completed within a given month.

Lines 26-29 set up the PROJXREF and TEMPDATA files for the JOIN command (line 30). As stated before, TEMPDATA is the active file. Each record of TEMPDATA will be compared to each record of PROJXREF. For those records which have matching project numbers, the matching records will be joined together as a single record and added to a file called ACTIONS.

Once the JOIN command completes its work, the resulting ACTIONS data base is indexed on DWGNR, since this is the desired sequence in the written report (lines 31-32).

The next task (lines 33-47) performs a formatting job. Remember that some projects have more than one corresponding device type record in the PROJXREF data base. Without modification, the completion report would look like the following:

5962-88472	01	SG117R	5962-E1201	01/15/90
5962-88472	02	LM117HVH	5962-E1201	01/15/90
5962-88472	03	LM137H	5962-E1201	01/15/90
5962-88472	04	LM137HVH	5962-E1201	01/15/90
5962-88472	05	SG117AT	5962-E1201	01/15/90

Management, however, insists that the

duplicated drawing numbers, project numbers and document dates be suppressed. They want the report to appear as in Figure 1.

This short routine replaces duplicate information with blanks. Line 33 is very important. This command closes the index. Without the ACDN index closed, every time a drawing number is replaced with a blank, the entire record sequence is rearranged and the routine will fail. Line 47 then reopens the ACDN index.

The next few lines get user input for the report heading. When you run the program, enter "January, 1990".

Lines 59 and 60 redirect output from the screen to the printer. Line 58 prevents the @...SAY commands from being echoed to the screen. Line 61 sets the left margin to 0. If the report does not appear centered, simply change this value. All @...SAY commands will be shifted accordingly.

Lines 62-63 center the heading (based on an 80-column width). The TRIM function must be used to remove trailing spaces.

The DO WHILE loop starting at line 65 addresses a typical logic problem related to multiple-page reports - when do you print a page heading and when don't you? Obviously, at the top of every page. That is obvious but the programming solution may not be so obvious. Most of the difficulty stems from not thinking of the first page as any other page.

What triggers a new page? When the line number exceeds the maximum for the report specification. Therefore, the first thing in the loop is to test for the line number. If it exceeds the maximum, do a form feed followed by a page heading. This is what line 66 does. Line 67 resets the LINE variable to 1. Since the printer has just finished printing line 50, it obviously can't go back to line 1; therefore, it must print on line 1 of the next page.

The report heading is then printed and the LINE variable incremented by 4 (line 72). The record data is then printed (lines 74-79), LINE incremented (line 80) and the record pointer advanced. Line 64 insures that the first page gets a heading. (Any value greater than 50 will work.) After the main body of the report is completed, the report summary is printed. This includes a breakout of new documents versus revisions and a summary of projects by engineer.

If you started with the original data base and expanded it as indicated earlier, the report summary should look like Figure 2. Note that the totals are not right justified. Unfortunately, dBASE III has no single function that will right-adjust numeric variables. You either must create a routine to

do this or use the unconverted numeric. If



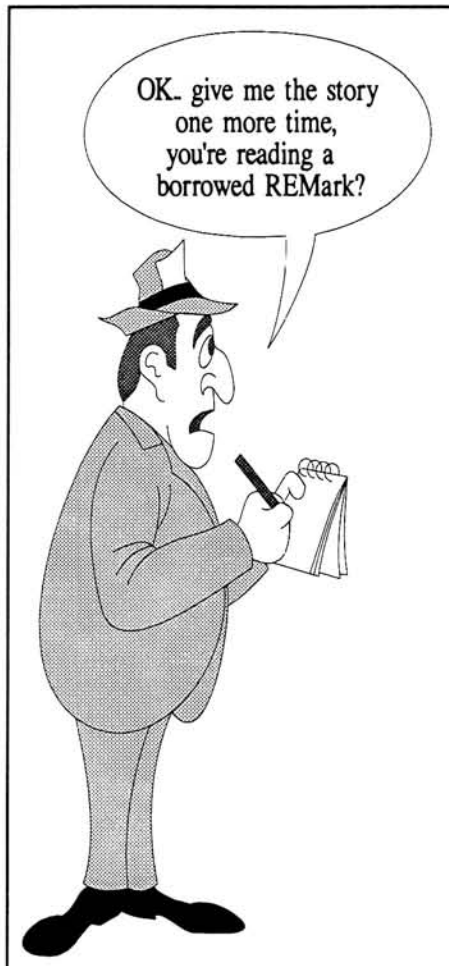
5962-88472	01	SG117R	5962-E1201	01/15/90
	02	LM117HVH		
	03	LM137H		
	04	LM137HVH		
	05	SG117AT		

Figure 1

you take the latter approach, then you must account for the total width of the numeric. Also, you cannot include a character string and a numeric variable in the same @...SAY command. As an example, if you wanted the total new document count to print one space after the string "Total new documents:", you would have to use two separate commands:

```
@ LINE, 9 say "Total new documents:"
@ LINE,22 say NEW_DOCS
```

To finish the report process, line 93 sends a form feed to the computer and lines 94-96 redirect output back to the screen. This is a very simple report-generating program. A more professional report may require a different page heading for the first page than subsequent pages, page numbering, subtitles for different groupings and other enhancements including different font styles for different parts of the report. None of these requirements would be beyond the capabilities of dBASE III. In my next article I will discuss programs that provide on-line help, access to DOS and backup procedures.



### Listing 1

```
1 procedure REPORT

2 set safety off
3 set talk off
4 clear
5 @ 4,28 say "MONTHLY REPORT"

    * Get start and stop dates
6 set intensity on
7 store ctod(' / / ') to START_DT
8 store ctod(' / / ') to STOP_DT
9 @ 10,10 say "Enter start date: " get START_DT
10 read
11 @ 11,10 say " Enter stop date: " get STOP_DT
12 read

    * Copy completed projects to temporary file
13 @ 10, 0 clear
14 @ 10,10 say "Copying completed projects to temporary file ..."
15 use PROJECTS
16 copy to TEMPDATA for APPDATE >= START_DT .and. APPDATE < STOP_DT

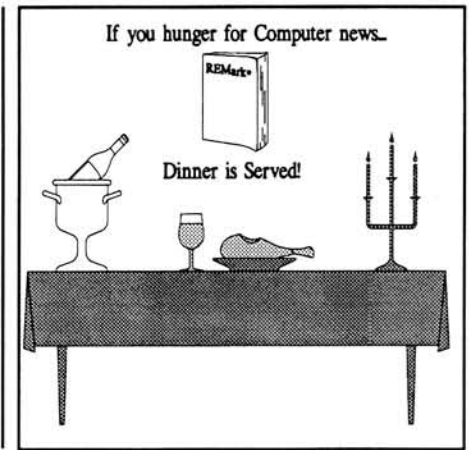
    * Calculate totals
17 use TEMPDATA
18 count for REV = " " to NEW_DOCS
19 count for REV <> " " to REV_DOCS
20 count for ENGINEER = "ABC" to ABC_TOT
21 count for ENGINEER = "JAD" to JAD_TOT
22 count for ENGINEER = "JZB" to JZB_TOT
23 count for ENGINEER = "TTT" to TTT_TOT

    * Join device type data with project data
24 @ 10, 0
25 @ 10,10 say "Joining device type data with project data ..."
26 select 2
27 use PROJXREF
28 select 1
29 use TEMPDATA
30 join with PROJXREF to ACTIONS for PROJNR=PROJXREF->PROJNR

    * Index on DWGNR for report listing
31 use ACTIONS
32 index on DWGNR to ACDN

    * Replace duplicate drawing numbers with spaces
33 set index to      && Turn index off
34 go top
35 do while .not. eof()
36     store DWGNR to mDWGNR
37     skip
38     if DWGNR = mDWGNR
39         do while DWGNR = mDWGNR
40             replace DWGNR with space(10)
41             replace PROJNR with space(12)
42             replace DOCDATE with ctod(' ')
43             skip
44         enddo
45     endif
46 enddo
47 set index to ACDN

    * Get heading for report.
48 set confirm on
49 set intensity on
50 store space(16) to HEADING
51 @ 10, 0 clear
```



```

52 @ 10,10 say "Enter heading for report: " get HEADING
53 read
54 set confirm off
55 set intensity off

56 @ 12,10 say "Ready to print - Press any key to start"
57 wait ""

58 set console off
59 set device to printer
60 set print on
61 set margin to 0

```

```

62 HEADING = "Drawing Completion Report for " + HEADING
63 HDG_COL = 40 - len(trim(HEADING))/2
64 store 51 to LINE
65 do while .not. eof()
66   if LINE > 50
67     LINE = 1
68     @ LINE,HDG_COL say HEADING
69     LINE = LINE + 2
70   text

```

DRAWING NUMBER	DEVICE TYPE	VENDOR PN	REV	PROJECT NUMBER	DOCUMENT DATE
-------------------	----------------	-----------	-----	-------------------	------------------

```

71   endtext
72   LINE = LINE + 4
73   endif

74   @ LINE, 7 say DWGNR
75   @ LINE,21 say DEVTYPE
76   @ LINE,27 say VENDPN
77   @ LINE,46 say REV
78   @ LINE,50 say PROJNR
79   @ LINE,62 say DOCDATE

```

```

80   LINE = LINE + 1
81   skip

```

```
82 enddo
```

```
83 text
```

REPORT SUMMARY

```

84 endtext
85 LINE = LINE + 4
86 @ LINE, 9 say "Total new documents: " + ltrim(str(NEW_DOCS))
87 @ LINE+1,13 say "Total revisions: " + ltrim(str(REV_DOCS))
88 @ LINE+3,25 say "Project Count by Engineer"
89 @ LINE+5,29 say "ABC: " + ltrim(str(ABC_TOT))
90 @ LINE+6,29 say "JAD: " + ltrim(str(JAD_TOT))
91 @ LINE+7,29 say "JZB: " + ltrim(str(JZB_TOT))
92 @ LINE+8,29 say "TTT: " + ltrim(str(TTT_TOT))
93 eject
94 set console on
95 set device to screen
96 set print off
97 return

```

REPORT SUMMARY

Total new documents: 46  
Total revisions: 2

Project Count by Engineer

ABC: 20  
JAD: 14  
JZB: 8  
TTT: 6

Figure 2  
Report Summary



## ACCOUNTING & TAX

Not sure if you need the expensive 'Chinese Flower 1-2-3', or 'Spanish Numeral Four' spreadsheet programs? Then find out for only \$20! "CheapCalc" will do double precision addition, subtraction, multiplication, division, power, SUM, and roots (using fractional powers). CheapCalc has many other functions too numerous to mention (just like the expensive spreads)! CheapCalc is available for all Heath/Zenith computers and operating systems. For more information, check out page 58 of the Software Catalog Update #1, or call HUG and order your copy today.

### Classified Ads

**FOR SALE OR GIFT TO A QUALIFIED ORGANIZATION:** H-8 Digital Computer, H-8-1 Static Memory, H-19 All-in-one Case (Terminal/Monitor - B/W), H-8-5 Serial I/O and Cassette Interchange Card, H-17 Floppy Disk. Will deliver within reasonable distance of Chicago or St. Joseph. Call 1-312-643-9497.

**TURBOSPORT 386 LAPTOP:** 2400 baud internal modem, 3 meg. RAM, math co-processor, 40 meg. drive, excellent condition, asking \$2,900. Call Steve at (415) 521-0434.

**RETIRED ZENITH DATA SYSTEMS SERVICE CENTER:** 10 years worth of Zenith Data Systems service manuals, Z-100 parts nearly all new. Z-100 software. Also Z-100 computers for sale. Send self-addressed, stamped envelope for complete list to Harry Bailey, P.O. Box 6487, Panama City, FL 32404 or call (904) 769-5081.



# dBASE III

## Part 10

# Getting the Bugs Out

**D. R. Cool**  
**7421 Troy Manor Drive**  
**Dayton, OH 45424**

There are basically three types of bugs that can occur in programs. This is true not only with dBASE III programs, but with virtually any program written in a high-level language. The first, and easiest to spot, is the type that results from a typographical error, such as typing "BROVSE" instead of "BROWSE". With this type of error, you get an immediate response from dBASE, such as "Unknown command" or "Syntax error". In the first instance, the basic command verb is wrong; in the second instance, an error has been made in the scope, expression list or condition phase of the command. In this instance, dBASE III will repeat the statement with a question mark immediately above or to the right of the error. For example, the following command will elicit the "Syntax error" message:

```
COPY TO TEMP PROJNR, ENGINEER
```

The intent of this command is to copy the fields PROJNR and ENGINEER of the data base in use to a file called TEMP. What is missing is the word "FIELDS":

```
COPY TO TEMP FIELDS PROJNR, ENGINEER
```

This particular error is easy to spot, especially if you check the syntax of the COPY command in the dBASE III manual. Other errors in longer and more complex commands may not be so easy. For example, take a look at the following command:

```
START_DT = "01/01" + year(date())
```

The intent is to create a variable START\_DT by using the YEAR and DATE functions. Issue this command and you will get a "Data mismatch" error message. The problem is that this command is attempting to combine a numeric variable with a character string. The numeric variable "year(date())" must first be converted to a character string using the STR function:

```
START_DT = "01/01" + str(year(date()))
```

The second type of error is not usually the result of a typographical error. It occurs during the running of a program and possibly only under certain conditions. The types of error messages you get are "Variable not found" or "File not found". The

message "Variable not found" probably occurs more often than any other. A typical case would be a program that manipulates two or more data base files using the SELECT command. At a particular point, you may have forgotten to switch to the correct file and the program attempts to use a file name which does not exist in the currently selected data base.

Whenever a program bombs out, for whatever reason, you will always be presented with three options: ignore, suspend or cancel. If you choose "ignore", the program will attempt to continue. In some cases, you will simply get another error message; at other times, the program will continue without additional errors. I usually use this option only when I already know what the problem is and I simply want the program to return to a main menu where I can exit normally. (My program "exit" points usually reset all environmental variables to the default values.)

The second option, "suspend", is the option to choose for debugging purposes. In the "suspend" mode, all memory variables in existence at the time of suspension will not be released and all files that were open at the time of suspension will remain open. This allows you to examine the memory variables with DISPLAY MEMORY (F7) or see which file is currently in use with DISPLAY STATUS (F6). DISPLAY STATUS will also display all files in use with their indices, whatever procedure file is in use, if any, and the status of various environmental variables.

The third option, "cancel", will close all open files and release all memory variables. You generally use this option when the problem is obvious and you want to immediately edit the program to correct it. The one thing the "cancel" option does *not* do is reset all environmental variables to their default value. For instance, if you program SET TALK OFF, it will remain OFF when you "cancel".

The third type of bug is the most difficult

to trace. This bug never produces an error message. Your program simply doesn't do what you think it should. This type of error can cost you an entire day of effort. I know from experience. There are a number of options for tracing this type of error. The command SET DEBUG ON coupled with SET ECHO ON will cause all commands to echo to the printer as they are being executed. Only those commands actually executed are echoed, so you can follow exactly the program logic. The technique is to insert the two commands at a point in the program where the problem is occurring. Watch out for inserting these commands prior to DO WHILE loops, such as those that check on key input using the inkey() function. This type of loop continuously executes commands until a key is pressed and you will fill up a printer buffer pretty fast. The advantage of this technique is that it doesn't mess up any screen displays. Once you have a printed output, you can compare it to the program and check the program flow.

Even then, you may know what a program did but still not know why. This is where you have to narrow down the search. Since most program flow involves decisions which are based on the values of variables, you will want to start checking the values of memory variables at various points in the program. This can be done by inserting the SUSPEND command at strategic points in the program. When the program gets to the SUSPEND command, the program will stop execution and return to the dot prompt, allowing you to enter commands, examine variables, and do whatever you think is necessary to locate the bug. You can then resume execution of the program with the RESUME command. There is still a further option which you can use alone or in conjunction with SET DEBUG ON and SET ECHO ON. This is the SET STEP ON option, which causes a program to execute one command at a time. This option, coupled with SET TALK ON,

can be very effective in tracing program flow.

Finally, if all else fails, you have the option of typing the program commands one by one at the dot prompt. Of course, for IF...ENDIF, DO WHILE and DO CASE structures, you will have to simulate the conditions and you run the risk of simulating a condition that is not true when you actually run the program. Nevertheless, there have been times when this technique was the only way I could finally isolate a bug.

After you have written and debugged a number of programs, especially more complex programs, you learn to identify certain error messages with the type of programming error which causes them. For example, unless you have misspelled a field name or memory variable, the message "Variable not found" can usually be traced to one of two causes: (1) the program is trying to use a field name of a data base that has not been selected or (2), the program is trying to use a variable that was initialized in a local procedure. In the first instance, the program has set up two or more data bases using the USE and SELECT commands. Then, at some point in the program, data base "A" is the currently selected data base, but the program is trying to use a field name is data base "B".

The second instance occurs whenever you call a procedure from a program or

another procedure. The called procedure initializes a memory variable. When the program returns to the main program, an attempt is made to use the memory variable in a command. Since any variable initialized within a procedure is released when the procedure is exited, the main program will respond with "Variable not found". The solution is to define all global variables at the start of the controlling program. This makes them available to all subroutines, regardless of where they are called. An even better technique is to store global variables in a memory file and use the RESTORE command at the beginning of the controlling program. This way, if any variables or global constants have to be changed, only the memory variable file needs to be edited.

Another common error message is "End of file encountered". The usual meaning of this message is that the program attempted to move the record pointer past the end of the data base in use. There is, however, another condition that can cause this message to be generated. For an indexed data base, a record could exist in the data base for which the key expression is not contained in the index. This can happen when a record is added to a data base without turning on all appropriate indices. Then, whenever a program attempts to retrieve this particular record with the FIND or SEEK command, the "End of file" message will be

generated. The remedy at this point is to REINDEX and to make certain that all appropriate indices are turned on when you do an APPEND or EDIT.

If you are running a large program that uses a large number of data bases and indices, sooner or later you will probably see the message "Too many files are open". What this means is that you attempted to open more than 15 files, which is the maximum allowed by dBASE III. This count includes the dBASE program itself, all data base files, all index files, all report and label form files, format files and memory variable files. When you see this message, take the "suspend" option and DISPLAY STATUS to see what files are open. If upon counting the number of files in use, the number is less than 15, you probably have a problem with your MS-DOS config.sys file. Check this file with an editor or EDLIN to see how many files are set with the "FILES =" statement. If it is less than 15, this is the problem. Change the value to at least 20.

Occasionally, while running a program, you may see the message "internal error: Illegal opcode" or "Internal error: Unknown command code". The reason for this type of error is not known. My usual solution is simply to exit dBASE and start over. If that still doesn't correct the error, you will have to contact Ashton-Tate Software Support.

In my next article, I will discuss some advanced programming techniques. ✻

**SURPLUS  
COMPUTER PARTS**

Surplus Trading Corporation has purchased the entire obsolete inventory from ALL the Heath/Zenith Computers & Electronics stores in the United States. This purchase covers all new and refurbished parts at the Zenith Data Systems locations for computers, monitors, Heath products and any related products.

If you are in need of parts for your Heath/Zenith units, call us NOW!! Parts are selling quickly, as our customers are now starting to order. There are only so many parts available.

Whether you need power supplies, board assemblies, hard and floppy drives, cables, modems or other parts. Call us, we just might have what you are looking for and save you money, too.

Keep in mind we also have surplus used parts available, such as 20 Meg. hard drives for \$60.00 - Just mention this ad.

Call now for a quote on the parts you need. We will explain our ordering procedure when you call . . . so call now!

---

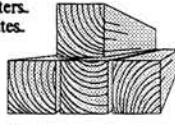
**Surplus Trading Corp.**  
THE HOUSE OF EVERYTHING "ALMOST!"  
2700 N. M-63, P.O. BOX 1082  
BENTON HARBOR, MI 49022

---

CALL: LES - (616) 849-1800  
FAX (616) 849-2995 (ORDERS ONLY)


Reader Service #153

Splinters.  
termites.



With this board  
all you have is  
a stick in the mud!

Messages.  
Orders.



Welcome to  
Zenith Data Systems  
ZLink-COM!  
First Name \_\_\_\_\_

With this board,  
you're really talking!

**Which is more useful?**

# Zenith Data Systems



---

# Adding Laserjet Soft Fonts to Wordperfect 5.X with Bitstream's Facelift

(Or If You Aren't Printing Graphics, Who Needs Pagemaker?)

William G. Nabor  
27172 Huerta Street  
Mission Viejo, CA 92692

More and more frequently now, there are coming on the market computer products that are so outstanding, so wonderful, so magic, that you want never again to have to do without them. Such a product is Facelift from Bitstream, of Cambridge, MA. Facelift does for WordPerfect what typesetting does for printing. It allows Wordperfect to send instructions to a laserjet-compatible printer and have the result look typeset. All this with hardly any additional effort on the part of the user, other than selecting the softfont you wish to use. Finally Wordprocessing comes of age. You can now type a letter and have the result look like a professional printer's.

This has been possible for some time now, of course. Desktop publishing programs such as Pagemaker and Ventura were written for this express purpose. Font cartridges have been available for laserjets from the start. Many softfont packages have been put on the market. All of these suffered from one or more severe defects that made their use desirable only for the most dedicated users. Desktop publishing packages were expensive, hoggish of computer resources, hard to learn, harder to use, and slow, slow, slow. Font cartridges were expensive and offered only a limited selection of fonts. Softfonts were slow and required more computer disk space than you would ever have and bunches of printer memory that you probably didn't have either. For example, my set of fonts (from Bitstream, no less) took up 27 MB of hard disk storage. Naturally I didn't keep them all on the disk. Shuffling them onto the hard disk from floppies as needed was such a royal pain that I rarely made use of any but the most basic fonts. All of this is now changed.

I guess it's time to discuss fonts and typefaces. As you are all aware, the alphabet is composed of characters whose shapes are only loosely defined. Indeed, there have been several scholarly studies

into just how deviant a character can be and still be recognized as a part of the alphabet. We can have printing or script, serifs (the little doo-dads at the tops and bottoms of some characters) or not, square shapes or free-form, proportional spacing (with or without kerning) or monospaced, large or small print, landscape or portrait orientation, and so much else besides. All of these characteristics can be broken down into 5 major subjects:

- Typeface
- Size
- Spacing
- Orientation
- Character set

A typeface is a set of printable characters of a certain design. Typefaces are not to be confused with fonts (a fact that many printer and cartridge buyers learned to their disappointment). There have been so many different typeface designs over the years that nobody has seen them all. Designing a good one takes skill and patience and a great deal of work. More on this later.

Size can be measured in two ways. For monospaced typefaces, the preferred unit is characters per inch (CPI) - the number of characters you can print in one inch across the paper. Because proportional type can vary the number of characters per inch depending on the characters printed, proportional typefaces are usually sized in points. 72 points equal about one inch in height. Width varies with the character.

Spacing can be either monospaced, where each character occupies the same horizontal space, or proportional, where each character occupies space proportional to its shape. The classical typewriter courier typeface was monospaced. This characteristic, above all else, is what makes typewritten material look typewritten. "WHIM" occupies as much space as "inch". Your computer's monitor almost certainly outputs monospaced characters at 80 to the line. This makes it easy for users (and

manufacturers) to print to the screen, since they know in advance how much space a character will take without having to know exactly what that character will be. Proportional print, on the other hand, spaces the characters according to their shape. "W" is wider than "i" and so forth. A program that outputs proportional print has to have a look-up table that tells it how far over to move the printhead on the paper (or cursor on the screen, or whatever) after printing each character. You have to move farther after printing an uppercase M than after printing a lower case p, for example. Much harder to do, but the result looks so much better. That is the major difference between the looks of a typeset document and one that is not. In olden days this was easy for printers to do since each letter was actually a chunk of lead that was only as thick as it needed to be. The printer had only to load the lead letters in the rack. The computer has a much harder task.

In addition to proportional spacing, certain character pairs can be kerned. That is, moved closer together so that one actually overlaps the other. WA can be printed so that the end of the leading tail of the A is actually farther left than the upper tail of the trailing leg of the W. "fi" can be printed as one character.

There is, next, the matter of orientation. You can print so that the reader holds the paper with the long side down or so that the reader holds the paper with the long side horizontal. The former is called "portrait" orientation and the latter "landscape". Hold this issue of REMark with the staples in the vertical plane and that is "portrait". Hold it with the staples horizontal and that is "landscape".

The character set is the collection of things that can be printed. The most basic is the ASCII set, which comprises, for the most part, only those characters found on your keyboard. Other sets may include such characters as the copyright symbol,

the ace of spades, the smiley face, and so forth. The more characters in a set, the more disk space is required to store it, the more printer memory is required to load it and the slower it will be loaded.

A certain typeface, with a certain character set, at a certain size, with certain spacing characteristics, in a certain orientation is called a font. When a printer manufacturer claims that his printer has, say, "12 resident fonts", don't think that he is offering 12 different typefaces. He isn't. Those 12 fonts may be all of one typeface in two orientations and six sizes. That may be generous, but it doesn't compare to what Facelift offers.

The information necessary to get a printer to print in a certain font can come in one of three ways:

1) **Resident** - The information is stored in the printer's ROM (or on the printer's printwheel in the case of a daisywheel printer). All dot matrix, laser, and similar printers have to have at least one resident font stored in ROM. Most have several.

2) **Cartridge** - The fonts are stored on ROM chips that you plug into the printer (and, of course, purchase separately at extra cost).

3) **Softfont** - The information is stored on the computer's hard disk and sent to the printer at print time where it is stored temporarily in the printer's memory. Softfonts, too, are sold separately, but on floppy disks, at extra cost.

Heretofore, there were tradeoffs with each format. Resident fonts were included in the cost of the printer and were instantly available for use, but were limited in number and couldn't be modified by the user to fit a particular circumstance. Cartridge fonts were expensive and limited in number. Softfonts took up a gross amount of expensive hard disk space, needed a lot of printer memory, and were slow to load, making printing time long. You paid your money and lived with the limitations of what you got. Now, however, you can have it all.

Facelift is a softfont generating program that creates fonts as needed with the characteristics you choose, then sends the information to the printer when called for. What is stored on the computer disk are only the instructions on how to create the font from the chosen typeface, and this takes up surprisingly little space. I have 55 different typefaces (Bitstream offers more than 200) stored in 1.2MB of disk storage space. I can create any font from any of these 55 different typefaces in any size from 4 point to 1000 point in 0.1 point increments (albeit anything over 150 point is probably not useful). Let's see, 9,960 point sizes times 55 fonts times 2 orientations is over 1 million fonts. Find a cartridge to match that. Now find that cartridge with a price tag of less than \$500.00 (plus 1MB hard disk storage). If you are willing to do with only the fonts that come with the

basic Facelift package, you can get by with spending only \$60-80, less than the price of some games. A bargain no matter how you look at it.

To use Facelift, you will need the following:

- A Zenith-compatible computer with 640K memory (Facelift can make use of expanded (EMS - LIM 4.0) memory for those of you that have expanded memory and a driver to manage it), and a hard disk with about 1MB of storage space.

- WordPerfect V5.0 or 5.1, installed with the proper .ALL file in a directory you can get at.

- A Hewlett-Packard laserjet compatible printer (a Laserjet IIP is highly recommended if you have yet to purchase a laser printer). The more memory you can get with the printer, the better (see the section on BackLoader at the end of this article).

- Facelift

- One or more Bitstream typefaces.

Facelift comes bundled with 13 typefaces: Dutch - a serif style face in roman, italic, bold & bold italic; Swiss - a sans-serif style face in roman, italic, bold & bold italic; Cooper - a bold headline; Park Avenue - a fancy script; Formal Script - a formal script (what else?); Brush script - an informal script; Monospace - a non-proportional sans-serif face (the other typefaces are all proportional). In addition, Bitstream offers a companion "Value Pack" with 24 more typefaces for \$125.00. You can add more typefaces as you see the need by buying them from your local software store. The price ranges from \$89.00 to \$120.00 per disk of 4 typefaces, depending on where you buy them (the "list" prices for Facelift and the ValuePack are somewhat higher than the ones I quote here. My quotes are what I paid via mail order). Some special packages offer more typefaces at greater cost. When selecting a typeface package, remember that boldface, italic, roman, and bold italic are really 4 different typefaces, even if they all have the same name (e.g. "Amerigo"). Thus the Amerigo typeface comes on one disk, costs \$89-120, and has roman, italic, bold, and bold italic Amerigo typefaces.

Installing Facelift is simple and automatic. You just run the setup program that comes with the Facelift package. You have to have Wordperfect V5.0 or 5.1 installed with the HP laserjet printer driver specified and installed. WordPerfect will have generated an .ALL file for that printer. You must know where on your hard disk the .ALL file is so you can tell Facelift, which has to modify it. Wordperfect 5.0 users will have to have the latest version (check with Bitstream or WordPerfect). Facelift installs onto the subdirectory where the .ALL file is. The typefaces should go in their own subdirectory, such as \WP51\FONTS. During the

installation procedure, you will be asked for the location of the .ALL file, which typefaces you wish to install, and where, and a few other basic questions. After you have loaded all the typefaces you wish, the install program then assembles the necessary instructions and stores the output in the font subdirectory. This may take some time (minutes to hours depending on your computer's speed, the presence or absence of a math co-processor, and the number of fonts you are installing), but is only done once. Upon completion of this task, Facelift is ready to use (but WordPerfect isn't - see below). If you purchase more typefaces later you can load them in via the LOADTYPE program that comes with Facelift.

Having installed Facelift, you need to tell WordPerfect about it. This strikes me as a step that, with a little bit of effort on WordPerfect's part, could have been eliminated, but there it is. Invoke WordPerfect with a new command -FLWP. This will load Facelift into memory (about 31KB, most of which will be in EMS if you have it), and then automatically invoke WordPerfect. Call up WP's printer screen (shift-f7) choose SELECT PRINTER (S), choose HP LASERJET (if not already chosen), choose EDIT (3), choose CARTRIDGE & FONTS (6), choose SOFFONTS, choose FL fonts. You should now see a list of the Bitstream typefaces you have loaded. Mark each one of these fonts with an asterisk. Normally this would tell WordPerfect to download each of these fonts to the printer at the start of the printing session. Don't worry, with Facelift loaded this won't happen. By marking each typeface with an asterisk, you are only telling WP that these fonts are to be made available for use. Exit out of this section by hitting f7 until you are back to the edit screen. Enroute there you will see WordPerfect update its files.

When you are ready to choose a font while in WP's edit screen, move the cursor to the beginning letter of the first word you want in the new font, hit CTRL-f8 then 4 (base font). You will be presented with a list of fonts available. Use the arrow keys to highlight the font you wish to use then hit ENTER. You will then be asked the point size you want. Remember that the larger the number the larger the print. 12 is the "typical" size for normal output. Enter this number and then hit ENTER one more time. From that point on, until you give WP a new font command, the words in your document will be printed in the font you chose. The key is to remember to have the cursor positioned just to left of the word you want in the new font before hitting Ctrl-f8. Also, be sure to switch back to your main font after italicizing a single word. Remember that WordPerfect will continue to print out in the last font selected until encountering another font command. Facelift will create the font at print time and



download it to the printer. The downloading delay is surprisingly short until you get into point sizes greater than, say, 60.

Nothing is perfect, not even (or perhaps I should say especially) WordPerfect. There are some unfriendly aspects of the program that you have to work around. The most disagreeable one is due to WordPerfect's attempt at presenting you with what the programmers decided was the best compromise on a WYSIWYG (What You See Is What You Get) interface. Personally, I hate the results, especially when you try to put a document on the screen after you have called for some of Bitstream's fonts. If there wasn't a kludge workaround, I wouldn't use either product on a bet. The problem is this: A wordprocessor is different from a typewriter. Vive le difference. Anyway, some people have gotten so used to banging away on a typewriter and seeing the results on the paper instantly, that they either cannot or will not adapt to the special needs of a computer screen. They still want to see on the screen exactly what is going to be on the paper (WYSIWYG). This is especially true when they need to see where the pages will break. I will admit that this latter need can be acute. Those of you who have been around long enough to remember Peachtext know what a non-WYSIWYG interface worked like. You always got 80 characters across the screen and 25 lines down, regardless of what was going to appear on the paper. Sometimes when you printed you got "widows" and "orphans", single words or lines that looked way out of place on the paper. You could not see this on the screen prior to printing. You had to print, edit, print and repeat until you got it right. I will admit that that was a pain and a half. Some people will not put up with this. They want to see where on the line each word will be and where on the page each line will be. The technology isn't there yet to allow word processors to do this. Even some rather cheap printers have a resolution too fine to be duplicated, even on a VGA screen. A laser printer so far out performs even the best computer screen that there is no comparison. If you doubt this, type something then print it out using Bitstream's Swiss font at 4 points. You will probably need a magnifying glass to see it on the paper, yet it will be readable. Try to put something that small on the screen. Can't be done. In fact, screen characters smaller than the usual 80 across and 25 down already show marked degradation in readability. What do you do, then, on the screen when you have selected a font that will print out more than 80 characters per line on the paper? You can either make the screen characters smaller and risk making them unreadable, or you can scroll them off to the side. Wordperfect will allow you to do either, with some monitors and some drivers, but not all. There are some WordPerfect add-ons, such as Screen-

Extender from Stairway Software that will allow you to adapt to monitors that WordPerfect alone will not. You are still left with hard to read characters and some fonts that will put out to paper even more than these programs can put to the screen. What happens then is that Wordperfect will scroll the characters off the screen to the right. Some people can live with this. I cannot. When text scrolls off the bottom of the screen I have no problem. Recalling more text up from the bottom is like turning a page in a book and I can easily adapt to that. Having to "turn the page" back and forth at the end of each line, however, is unacceptable, and that is exactly what WordPerfect does with most Bitstream fonts below about 14 points (i.e., the most common sizes). In order to edit such a mess, it is necessary to print it. I lose track of my thoughts when I have to scroll back and forth (not to mention fighting down a feeling of seasickness). BAH!

There is a fix, as I have said, or else I would not be recommending Facelift. The trick is to select Courier 10 CPI as your default font (hit shift-f7 then S then 3 then 5 and use the arrow keys). This selects a monospaced font that prints 10 characters per inch. Since your paper is almost certainly less than 8 inches across (allowing for the margins necessary for laser printers), this means that WordPerfect will think that your paper holds  $8 \times 10 = 80$  characters per line, exactly the same as the default font on your screen. Therefore, WordPerfect will not scroll the line off the edge. Compose your whole document in this font, edit it to your satisfaction, spell check it, and get it ready for printing. Save it to disk in this format. Then go back and change the fonts to some of Bitstream's. Edit the document only for page breaks, orphans and widows. Save this modified file with the same name but a different extension. Then print the modified file. If you like what you see, erase the unmodified version. If not, then erase the modified one, edit the first, and reselect the fonts. This way you will be able to read your document on the screen and print it in fancy fonts, too. A bit of a bother, but it allows you to get the work done, and the results are really worth it. Someday WordPerfect might have a switch that allows you to switch back and forth between WYSIWYG and WYSIWIR (What You See Is What Is Readable), but for now, it's up to you.

Another problem with Wordperfect, which I really find inexcusable, is the way the program codes for the font you have selected. Unlike Peachtext, which included the control characters right on the screen along with the printable characters, WordPerfect hides the codes from view. After all, if you are trying to look like WYSIWYG, then having control characters mixed up with the printable ones on the screen simply won't do. Well, that would

be OK except that WordPerfect is sloppy about how it keeps track of the darn codes. If you select, say, Bitstream's Charter 12 point roman font, then decide that you really wanted it Dutch italic, you move the cursor back to the word you originally marked Charter roman and change it to Dutch italic. So far so good. However Wordperfect does not erase the old code for the roman font! If you slip up and move the cursor even one space to the left of where you had selected the first font, you will never get the second. The program will get to the second font call, download it to the printer, move over one space, see the original call and print out in the first font. ARRRGH! If you are indecisive as to what font you really want, and choose back and forth several times, you can get so fouled up that starting completely over with a blank screen becomes a viable option. To avoid this gotcha, make it a habit to always manually erase old fonts when you change to new ones (alt-f3 to reveal the codes, then use the arrow keys to highlight the font call (easily seen) then press Del). You do this only when you change your mind about a font and need to erase the old one. Do not do it if you are changing a font on purpose after printing out a few characters in the original font (as in making a single word italic).

The next "problem" is really an annoyance only, and it occurs only if you purchase the Bitstream Dingbats-Symbols typeface package (sold separately at extra cost). Dingbats, by the way, are little symbols, such as a telephone, dagger, pointing hand, or snowflake, that one puts in a document for decoration or to call attention to something. To use the dingbats after you have loaded them you cannot just print a character and then tell Facelift to print it as a dingbat. You have to use a tedious WordPerfect routine instead. Move the cursor to the place in your document where you want the symbol to go, then press Ctrl-V. At the bottom of the screen will pop up a Key= question. Punch in the two-numbers-separated-by-a-comma code of the dingbat you want. Bitstream gives you a list of the two-number codes for each dingbat or symbol, but that is just one more key you have to have around to operate WordPerfect. Annoying, as I said, but you can get used to it. You do not use symbols that often. Or do you?

Finally, there is a problem managing the different fonts in the printer's memory.

### Backloader

This is probably a personal fault of the author's, but Bitstream, for all its technical savvy and slick professional output, has always left behind a small whiff of something unsavory in the marketing of it, kind of like a car dealer who, after selling you a new luxury automobile, blandly asks you if you want an engine to go with it. There

always seems to be something else you need. First it was the Fontware installation package you needed to get the typefaces to work, now Facelift makes the Fontware program unnecessary and obsolete, but comes with an insert advertizing yet another program, Backloader, from the Laser Toolworks division of Roxolid Corporation. The people at Bitstream, it turns out did not write Facelift themselves, Roxolid wrote it for them. The people at Roxolid, almost certainly with an eye toward their own profits, did not include a printer memory manager with Facelift. They left it up to the purchaser of Facelift to discover the need of such a thing and then to supply it, at extra cost, of course. It's called Backloader.

Backloader is not quite the engine to the Facelift chassis, but it's certainly the windshield and bumper. You can use Facelift without any auxiliary programs, but you might soon discover trouble. The more fonts you use, the bigger they are and the less memory your printer has, the more likely the trouble gets. Here is the problem: In order to print with a certain font, the printer has to have that font stored in its memory. WordPerfect, upon sensing that the following character(s) are to be printed in a certain font, calls upon Facelift to generate the font and send it to the printer first, a process called downloading. Most fonts at 12 points take up something like 50-60 KBytes of printer memory. The bigger the font, the more memory it takes (and, incidentally, the longer it takes to load). After downloading the font, Facelift turns control back to WordPerfect which then sends the characters to be printed. Upon encountering another font command, this process is repeated, each additional font taking up more printer memory. Use several big fonts and sooner or later you fill up the printer's memory. Facelift-WordPerfect then becomes less than user friendly. What should happen at this point is for Facelift to unload the least-used font to make room for the next one. At the very least the printing process should stop with a message flashed on the screen "Out of Memory". We users hate that message, but at least we know what's wrong and have the opportunity to do something about it. What does happen instead is that the printer either continues printing using the last font loaded, or else it selects one of the resident softfonts it thinks would look good there, or else it prints out gobblegook. Pray that the gook is not one character per page until the printer goes through your entire paper supply (oh, how I hated that one!)

Backloader is a TSR (terminate and stay resident) program that manages the laser printer's memory doing what Facelift ought to do. It knows how much memory your printer has (because you told it upon setup), how much memory each softfont takes,

which softfonts have been downloaded, and how to remove the font(s) least used to make room for the newest. It will also prevent Facelift from downloading a softfont for a second time if it's already in the printer (from a previous WordPerfect session, for example). So long as you do not turn off the printer without telling Backloader (which would then become confused), you will never have to wait while WordPerfect unnecessarily downloads a font for the second time. Nor will you be faced with printer output in the wrong font or in no recognizable font whatsoever. If you unload Backloader from memory after each WordPerfect session, it will reset the printer when it is reloaded. Backloader only becomes confused if you turn off the printer without telling Backloader (via the hot-key popup menu or uninstalling Backloader then reinstalling it).

You will probably want to get Backloader, even though this doubles the cost of Facelift. Compared to even the cheapest cartridge, it's still the bargain of the year. Too bad this is not carefully explained in the advertizing.

Backloader is easy to install. You simply create a sub directory on your hard disk and copy the Backloader programs onto it. Next you run the configuration program (this is where you get to tell Backloader how much memory your printer has), answer the questions that pop up on the screen and you are set. Of course, you can customize the program to your needs, but you do not have to do this. To use the program you simply change to the backloader sub directory and invoke the BKL program before loading Facelift-WordPerfect (using the new command - FLWP). Don't load Facelift or WordPerfect before loading Backloader. Backloader loads itself into memory and takes over from there. When you invoke Facelift, it first looks for Backloader and coordinates with it if it's there. You need do nothing yourself, but you can call up the Backloader menu by hitting the hot key (alt-B, unless you change it) and follow the instructions that pop up. You can do this at any time during your WordPerfect session. One installation problem I encountered seems to be a remnant of the pre-Facelift days. Since you have created and stored on your hard disk no softfonts (Facelift does this "on the fly"), Backloader cannot find any and complains. Just ignore this nagging.

With Backloader and Facelift you have an unbeatable font package. You cannot but be pleased with it. This is so because Bitstream is superb at what it does. Designing typefaces is a real art. Getting one to look good is not simply a matter of sitting down with a piece of graph paper and filling in the squares. You can get some really ugly typefaces that way. Bitstream's artists (and it is an art, as I said) are simply the best in the business. I like all the 55 typefaces I have used, with the exception

of Cloister, their version of "olde English", where I think the uppercase letters are unrecognizable, but this is a minor complaint. I have not seen results that look any better. Not from Adobe, not from Hewlett-Packard, not from Pacific, not from anyone. What's more, Bitstream's computer people have done a near-perfect job of rendering the typefaces laserjet printable as smooth, flawless, good-looking characters. No jagged edges here. Although I have only used Facelift for a month or so now, I have not encountered any computer glitches. It hasn't hung up the CPU, has not printed out garbage (after Backloader took care of the printer memory problem). I'm impressed. I'm sure you will be, too. Incidentally, Facelift is also available for Windows V3, so Pagemaker and Word fans can get in on the action, too. Here things work even better, since Facelift does not have to contend with the limitations of WordPerfect. Instead of downloading fonts to the printer, Facelift for Windows uses the computer to compose an entire page, then sends it to the printer as a graphic image. You do not need Backloader. You do need enough printer memory to handel a whole page of graphics, however, (about 2MB).

Aside from making your printed document look professional, there is one advantage to Facelift that I find remarkably handy. What do you do when you have composed a document (a memo, for example) that runs one page and 5 lines on the second page, even after adjusting the margins to their smallest value? Two-page memos (at least at my place) might as well never be written, as they will never be read. Or what about a flyer? A second page might double the printing cost, and for only 5 lines? How do you get your memo down to the obligatory one page? Before Facelift you had to either edit your memo and leave out something important, or (if you had it) change to a font quite a bit smaller, making the memo now only 75% of a page of fine print. Ugh! Some people couldn't read it now. With Facelift you simply select the same typeface, but maybe 0.5 points smaller and it fits exactly on one page. Bravo! As I said, you cannot help but like this program.

#### Addresses of Vendors

Bitstream (Facelift)  
215 First Street  
Cambridge, MA 02142-1270

Roxolid Corporation (Backloader)  
3345 Vincent Road  
Pleasant Hill, CA 94523

Stairway Software (ScreenExtender)  
700 Harris Street  
Suite 204  
Charlottesville, VA 22901





---

---

# On the Leading Edge

William M. Adney

P.O. Box 531655

Grand Prairie, TX 75053-1655

Copyright (C) 1991 by William M. Adney. All rights reserved.

## **Windows 3.0, Static Electricity and Humidity, LaserJet III, Your HP LaserJet Handbook**

Windows 3.0 is one of the best-selling programs today. Is it really as great as everyone says or is it just a lot of hype with very little substance? I have spent some time working with it in the last few months, and I have included some observations about it in this column that I think you will find interesting.

It seems appropriate to discuss the weather in the winter in this column because more people spend more time working with their systems. Perhaps you are considering adding some new hardware, such as a hard drive or more memory, and the weather has more to do with that than you might think. Why? Because the air in the winter tends to be extremely dry, which causes the static electricity to build up that results in those unpleasant shocks when you touch something. And that static electricity is not only just unpleasant. It can be deadly to disks and electronic components in computers.

### **Static Electricity**

Generating static electricity is quite easy, and it most often happens when you walk on a carpet. You can build up a "charge" of literally thousands of volts, which is usually discharged in an instant. Like any electric current, it flows from a high potential (e.g., your body) to a lower potential (e.g., an earth ground) in an attempt to "equalize" these potentials. For example, you may get a static shock when you reach for a light switch and your hand brushes by the metal screws that hold the switch cover. Those metal screws are frequently connected to a metal box which is connected to a ground system for the electrical distribution in your home or at work. For three-prong

electrical outlets, the large round prong is also grounded as part of the same system. That's why you can also get a static shock by touching the metal case of any unit that has a three-prong plug, including stereos, computers, and kitchen appliances.

Static electricity is generated all the time, but you do not normally notice it when the humidity is high. That's because the water content of the air acts as a "conductor" (like a wire) that dissipates the static charge as it is generated. So, you won't get a static shock when the humidity is high.

The amount of water in the air, which we call the humidity, is directly related to the air temperature. The warmer the air, the more water it can hold. If the air temperature is twenty degrees, it can hold much less water than air at seventy degrees, even if the measured humidity is the same. And that's what causes dry air in the winter. That 20-degree air may have a humidity of 70%. But when it is heated to 70 degrees or so, the humidity may drop to much less than half that, perhaps on the order of 20% or so. Aside from causing possible problems with electronic equipment, dry air can also irritate sinus conditions and a number of other problems.

### **Electronic Components**

If you have ever bought any kind of add-on board, memory or disk drive for your computer, you may have noticed that it most often is wrapped by some kind of anti-static bag or paper. ROMs, for example, are normally shipped with their pins embedded in what looks like black plastic foam, which is another type of anti-static material. You will also generally find detailed instructions about how to handle

those components because of possible static electricity problems.

If you are adding some hardware to your computer, there are a variety of safe ways you can do it to prevent static electricity from zapping any of the electronics. Here's the way I usually do it. First, I remove the cover from the computer (after powering it off and unplugging it first of course). Then I get the package containing the new component(s) and place it on the work table next to my system. I touch the computer's metal chassis to discharge any static electricity that may have built up, just in case. Then I open the package and keep the component in its anti-static packing in one hand while again touching the metal chassis. Now there should be no "potential difference" (i.e., voltage), and it should be safe to remove the component from the packing and install it in the computer.

Regardless of precautions, I prefer to work with the hardware on days when the air has enough humidity to eliminate static electricity. If I don't get zapped by static electricity after walking around during the day, then I figure it is a "good" day to work with hardware. At least it is a little safer for the electronic components. Perhaps you will find some of these ideas useful on your system too. And it is probably worthwhile to mention that these precautions seem to have worked fine for me because I have never had a problem with static electricity zapping a board or component.

### **A Sticky Subject**

As most of you probably know, the Graphical User Interface (GUI, pronounced "gooey") is the latest fad in PC software. And although the GUI has been around in

one form or another for at least 10 years, it has only been identified as a software "category" in the last year or so. For personal computers (in contrast to "office systems" like the Xerox Star), it was introduced in the Lisa computer by Apple. Some proponents of the GUI have embraced its use with an incredible religious-like fervor, to say the least. Indeed, a couple of Apple computer-owners have told me that THEY have a Macintosh, NOT a PC or "personal computer." And I also know some PC-compatible owners who will not even look at a Macintosh because they see it as a toy ("kid stuff"), which is not suitable for adults. My comment to them is that I would switch to a Macintosh in a second if I could make as many millions as Tom Clancy has. And if you don't know who Tom Clancy is, what can I say?

Like most software products, the GUI has its advantages and disadvantages, regardless of what kind of computer it runs on. It does not matter in the slightest whether you are looking at a PC-compatible or a Macintosh, even though some Macintosh owners would disagree with that statement. I have spent over 23 years working with computers and data processing, and during that time, I have seen a LOT of user interfaces come and go (I've even designed a few myself). And the type of hardware, brand or "size" of the computer has very little to do with the user interface, except perhaps the display capability of the system.

What are the advantages of the GUI? The biggest advantage is a "common" interface that is used by all programs, which essentially means that you only have to learn the interface one time. Once you learn how to use the GUI, you know how to use virtually any program that is written for it, whether it is a word processor, spreadsheet, data base, or some utility program or game. That is an advantage for a computer user, especially a new user, because there are fewer new things to learn. From a software developer's point of view, the primary advantage is that a program need not do everything, such as control screen displays and printers, because the GUI generally takes care of that. As a result, all programs written for a GUI really *SHOULD* be less expensive for a user to buy because less work is theoretically required, but that is obviously not the case for any software for any computer yet. In short, the GUI generally makes software much easier to learn, but it does not seem to make the software less expensive, at least for a compatible system.

If the GUI is so great, then what are its disadvantages? Well, the biggest disadvantage of a GUI is that it is slow. Because a GUI must handle many of the routine chores, such as controlling the display and the printer, the program is quite large. And you need not be a technical expert to

understand that larger programs take longer to run on a given computer which slows everything down. If you want to see a demonstration of this, take a look at the standard DOS version of Microsoft Word and compare it to Word for Windows on a system with an 80286 CPU. EVERYTHING takes longer, including the initial "startup" time, because you have two programs: Windows and Word. Unfortunately, that kind of comparison cannot really be done effectively on a Macintosh, but the same problem exists on that system too. The only cure for this is to buy a newer and more powerful computer which will run a program faster with a less noticeable impact on speed, but that obviously gets somewhat expensive.

In summary then, one must balance the primary advantage with the primary disadvantage of the GUI for PC-compatible systems. If you want a computer that is easier to learn, then you probably want to use a GUI like Windows. Keep in mind that the "cost" of that ease-of-learning will be that your computer will be slower, not to mention the potential cost of required additional hardware such as a hard drive and more memory. And there is also an additional software cost because you must buy new software that is specifically designed to take advantage of Windows' features; otherwise, you really do not get the full advantages of using Windows.

If you are primarily interested in the speed of getting a job done, then you probably do not want to use Windows, at least not yet. And depending on your perspective, Windows-based software may or may not be easier to learn. To some extent, that really depends on how well the software is designed for Windows, but that is also true for command-driven software too.

### Windows 3.0

This report is specifically on Windows 3.0 that is available from Zenith Data Systems. If you have faithfully sent in your registration card for a previous version of Windows, you should have received the special upgrade offer for Windows 3.0 at a special price of \$49.00, which expired on December 31, 1990. I always recommend sending in software registrations because occasionally manufacturers provide special upgrade offers, such as this one, and some users are inevitably disappointed because they did not receive the offer.

My first impressions of Windows 3.0 were that it was extremely easy to install and it presented some displays (even during installation) that I did not know my system was capable of. I also realized that Windows 3 was going to take a considerable portion of my hard drive because there were seven, 3.5-inch distribution disks. The test system was my usual Z-386/16 with 1 MB of standard memory, 1 MB of extended memory, and 3 MB of expanded

memory running under Zenith Data Systems MS-DOS 4.0. I installed Windows 3 on my 80 MB hard drive, which is a Seagate ST-4096 unit with a 28 millisecond average access time. My display system includes the Heath HVB-550 video card (which provides auto-switch CGA to VGA support) and one of the original ZCM-1490 FTM monitors. And I continue to prefer the Logitech serial mouse that I also installed.

The Windows 3 SETUP program checks out your hardware and basically installs itself with little assistance, other than changing the floppy disks. The program correctly sensed the video hardware and recommended installing VGA video support (which I later changed to EGA). The program also sensed the 80386 CPU and recommended installation in the 386 enhanced mode. I also permitted the installation program to update my AUTOEXEC.BAT and CONFIG.SYS files as necessary, but these files were special "vanilla" files that I constructed to see what would happen. The AUTOEXEC.BAT file only included a PROMPT command, and CONFIG.SYS included the SHARE command, a SHELL= command to expand the DOS environment space with COMMAND.COM, and the usual BUFFERS= and FILES= commands. It took just over 20 minutes to install the program, which included building some font files. I also configured Windows 3 to work with the LaserJet III and Epson FX-850 printers, but more on that in a minute.

After installation, I began exploring the various program groups: Accessories, Main, and Games. As I was looking through the various displays, I noticed that the cards in the Solitaire game looked more square than rectangular, which made the display look a little odd. I restarted the SETUP program (within Windows) and changed the display to EGA which fixed that problem. Aside from that, I did not notice any major change in the other displays after changing to EGA resolution. After that, I exited Windows to see how much disk space was required, and I was more than a little surprised.

The installation program created three directories: \WINDOWS, \WINDOWS\SYSTEM, and \WINDOWS\TEMP. The \WINDOWS directory contains 76 files which requires just over 3,181,000 bytes of space. I should note that the installation program originally added HIMEM.SYS to the root directory, but I moved that file to the \WINDOWS subdirectory and updated the CONFIG.SYS file with the new path. The \WINDOWS\SYSTEM subdirectory contains 41 files which uses just over 1,710,000 bytes of disk space. The \WINDOWS\TEMP subdirectory is used for temporary files, and no files were created there. As you can see, Windows 3 is a BIG software package that requires nearly 5 MB of disk space. It seems to me that the



disk space requirement is excessive, especially since Windows is supposed to be an "operating" environment, but I should note that a lot of "accessories" are also included and installed with Windows 3.

Aside from the disk space requirements for the installed version, I should review the basic software and hardware requirements for Windows 3 that are documented on page x of the Zenith Data Systems (Microsoft) Windows 3.0 User's Guide. The only software requirement is that you must have MS-DOS version 3.1 or later.

Although I ended up with approximately 5 MB of disk space used after the installation, the manual clearly states that you must have a hard drive with 6-8 megabytes of free disk space. Part of the additional space requirement is necessary for files created under Windows (with the Notepad for example). Of course you must also have a floppy drive that is required for the initial installation.

You will also need to have a monitor supported by Windows which should be no problem because all of the basic display modes from CGA to VGA are supported. A printer is also "required" but that should be no problem because it appears that Windows 3.0 has a wide variety of printer drivers, although at least one (LaserJet III) does not work like it should. A mouse is highly recommended, but not absolutely necessary according to the manual, but I think that it is absurd to use a graphical interface like Windows without a mouse.

I saved the CPU and memory requirements for last because I think some additional explanation is necessary beyond what the manual says. In order to understand these requirements, one must first know something about the three Windows 3.0 operating modes.

### The Real Mode

Windows 3 has three operating modes: real mode, standard mode, and 386 enhanced mode. The *real mode* is used on computers that have less than 1 MB of total memory, and it is the *ONLY* mode available on those systems. That specifically includes a system that has an 8088 CPU (or an 8086 CPU) with 640 K of memory (required). If you have one of the Z-150 series of computers for example, you will be limited to using the real mode, even if you have installed expanded memory. From an operational perspective, you will not be able to run a program in a "window" because there is just not enough memory to do so; that is, you will NOT be able to switch from one application to another within Windows, unless you terminate the first application and start the second one.

Attempting to run Windows 3 on an 8088 CPU would seem to be ridiculous for at least one other reason: that CPU really does not have enough "horsepower" to run Windows, even though it will run in the

real mode. What you will find is that system performance is incredibly slow. That, plus the fact that you will not be able to take advantage of task switching, seems to defeat part of the purpose of using Windows in the first place. If you have a system that has an 8088 CPU and are considering Windows 3, my recommendation is to forget it.

There is one particular "advantage" of the real mode. Real mode does provide the best compatibility with version 2 Windows applications, and you may need to use this mode for older Windows-based software. In other words, you may need to run some older software in real mode, regardless of whether you have an 80286 or later CPU. That's because there are some technical changes in Windows 3 that may not permit running older applications in the standard or 386 enhanced mode. I think you will generally find that you must buy an upgrade to any Windows-based software if you wish to take advantage of the new features (such as task switching). If you are considering buying new software for Windows 3, be sure that you verify that the program is specifically designed to run under this version. Regardless of how you install Windows 3, you can always start Windows in the real mode by entering `WIN /R` (for real mode) on the command line.

### The Standard Mode

The standard mode is available on 80286 or later CPUs (including the 80386 and 80486). Memory requirements include 640 K of conventional memory and 1 MB of EXTENDED memory (expanded memory will NOT meet this requirement). If you have a Z-241 or Z-248 (or any IBM AT), note that you must have used a memory card to increase the standard 512 K of conventional memory to 640 K, and a 2 MB Z-405 or Z-415 memory card is sufficient to meet those requirements for those Z-200 series of computers.

The standard mode runs Windows 3 applications (with multitasking) and also permits task switching among non-Windows applications. In this context, task switching means that you can open a window with the specific application (task), and you can switch among the different applications within Windows. Remember that the multitasking capability for Windows applications will probably not work for older software that was designed for Windows 2 as discussed earlier in this article. Regardless of how you installed Windows 3, you can always start it in standard mode by entering `WIN /S` on the command line, assuming that your system meets the minimum hardware requirements of course.

### The 386 Enhanced Mode

The 386 enhanced mode requires a minimum of an 80386 CPU (either the standard DX or 80386SX) with a minimum

of 640 K of conventional memory and 1 MB of extended memory (expanded memory will NOT meet this requirement). This mode provides everything the standard mode does, plus it supports the virtual memory feature of the 80386 CPU and allows for multitasking of non-Windows applications.

Regardless of how you installed Windows 3, you can always start it in 386 enhanced mode by entering `WIN /3` on the command line, assuming that your system meets the minimum hardware requirements of course.

### About Windows

Windows 3 is certainly one of the best selling software packages for 1990, and I seem to recall reading that over 2,000,000 copies have been sold. Up to now, I have presented only the facts about Windows 3 which I don't think knowledgeable persons would disagree with. But I have some thoughts about the subject that will, no doubt, be somewhat controversial, and I should state clearly that these are strictly my own opinions.

Those of you who have read my column for a while know that I have never been especially wild about Windows in any version. One reason for that was simply because using Windows added so much additional system overhead that overall response was very slow. Windows 3 does not seem to be much better, even on a 16 MHz Z-386 system with zero wait state memory and a fast hard drive. When I started Windows 3 on my system, it seemed to take a long time to start in the 386 enhanced mode. So, I decided to do some testing to see if the mode makes any difference. Not surprisingly, it does.

I started Windows in the real mode (`WIN /R`), and it took about 9 seconds to bring up the first display (Program Manager only). Then, I tried the standard mode (`WIN /S`), and that took 14 seconds to display the Program Manager. And finally, I tried the 386 enhanced mode (`WIN /3`), and that took 20 seconds to display the Program Manager. I also noticed that starting Windows 3 in the 386 enhanced mode also seemed to beat the living daylight out of my hard drive, probably because there is more program code to be loaded.

Even though I certainly have a slow 386 system, I think that 20 seconds is an extremely long time to load an operating environment like Windows. None of my other software comes even close to taking that amount of time to load, and I have all kinds of popular heavy duty software, especially graphics, that one might expect would take longer. For example, I use the GEM presentation team 3.0 and Draw Perfect 1.1 for graphics, and both present their initial displays in less than 10 seconds on my system. Word processors, such as WordStar 6.0 and WordPerfect 5.1, also

come up in considerably less than 10 seconds (about 5 seconds). And version 2 of Quattro Pro only takes about 8 seconds to load. I really have a problem trying to like software that is supposed to be an operating environment and takes longer to load than any of my application software. All of the programs mentioned also take about 25% longer to load under Windows 3 in the 386 enhanced mode, but that may be partially because they are non-Windows applications, and I have not yet attempted to fine-tune any of the Windows parameters. I also noticed that the time required for that software to respond to any command takes longer too. And when I consider the disk space and memory requirements for Windows 3, it becomes really difficult for me to find a lot to recommend it. But I did find one thing I really like.

The Windows implementation of Solitaire is probably reason enough to buy the software if you like games. Even though I don't care much for Windows as an operating environment for the reasons stated, the implementation of Solitaire is dynamite, and it is fun to play. Although it runs fine on my 386 system, I saw it running on an old IBM AT, and the response time was noticeably slower.

I have seen articles in which Microsoft apparently believes that implementing Windows 3 and Windows-based software can improve productivity up to 25% over a period of two or three years. From what I can tell, those productivity improvements seem to be based on the idea that it will be easier to learn new software in the Windows environment. While it is probably true that software will be easier to learn initially, I disagree that there can possibly be that much improvement on an overall basis.

The learning curve is admittedly steep when one is learning completely new command-driven software, but it flattens out as experience is gained. And if you are upgrading to a more current version of software that you are already familiar with, then the learning curve is minimal and may be non-existent, unless you specifically need to use one of the upgrade's features. Since most people seem to be familiar with WordPerfect, I'll choose an example from it.

Say you want to bold a line that contains a title, such as the subheadings in this article. To do that in WordPerfect, you use ALT-F4 to turn on the Block, hit the End key (to bold the entire line), and press F6 to bold it. You can also use the pull-down menus in WordPerfect 5.1, but that takes longer (because there are more keystrokes), even if you use a mouse. And when you use a mouse to activate those features, it takes longer yet because you have to remove at least one hand from the keyboard, find the mouse, position it, and then

perform the specific operation. Somehow, I really don't see much productivity improvement there because it takes longer, and I have completely ignored the possibility that the software may take more time to perform an operation (under Windows) in that example.

The whole point is that any productivity improvements resulting from a decreased learning curve in the short term would seem to be negated if even common word processing operations, such as bolding, ALWAYS take more time in the long term. Perhaps I'm stupid, but I just don't see any advantage or improvement in productivity there. For example, some of today's Windows-based software, such as Ami Professional, does not inherently support any command keys for common word processing operations, such as bolding or underlining, although you can assign operations (i.e., macros) to the Function Keys.

In summary, I think that a short-term improvement in the learning process is more than offset by the additional time required to perform any operation. If you want to see how that works, I would encourage you to see a demonstration of Ami Professional in terms of what it takes to perform a bolding or underlining operation. While this kind of software is certainly great for performing what now seems to be called "word publishing", I think it is considerable overkill for word processing. So far as Windows 3 is concerned, I have asked a few local friends the question: "What do you use Windows 3 for?" Although a half dozen people is not a very good sample, it was interesting to note that each one unhesitatingly said: "Playing solitaire." I agree, and I don't recommend Windows 3 unless you are interested in playing games. But if you need task switching and/or multitasking, what can you do?

### The Alternatives

One of the more obvious alternatives for multitasking is to use the OS/2 operating system. Unfortunately, that has many of the same disadvantages of Windows 3, especially the hardware requirements. And to take full advantage of OS/2's multitasking features, you really need to buy new software that was designed for OS/2. If you think software designed for DOS is expensive, you might want to take a look at the prices for OS/2 software. Although that is one way to go, I have been using another way that allows me to use all of my "old" DOS-based software and still have the same basic features that OS/2 has.

That other way is to use Quarterdeck's DESQview software in combination with their Expanded Memory Manager QEMM386. Even if you do not need the DESQview features, anyone who has an 80386 system should have QEMM386. And if you have an 80286 system, especially a Z-241 or Z-248, you REALLY need

Quarterdeck's QRAM software if you are using the Z-405 or Z-415 memory cards. Among other features, QRAM (and QEMM386) can "convert" extended memory into expanded memory that can be used with most popular software. More on this software in future columns.

### Windows and the LaserJet

I was also disappointed to find that Windows 3 seems to have problems printing on an HP LaserJet printer. I first noticed the printing problem when I attempted to print the help information (from the Notepad) on my LaserJet III. For those of you unfamiliar with the LaserJet, there is a "reserved" margin of about a half inch on all four edges of the paper that is "unprintable." When I attempted to print the help information, the top half of the last line on the page was printed, and similar problems occurred on subsequent pages. Sometimes the bottom half of the characters in the first line on the page was printed. In other words, the Windows 3 LaserJet III printer driver does not seem to know about the unprintable areas on the page, and it attempts to print there. That seems kind of silly to me because LaserJet printers have been around for years. Out of curiosity, I decided to create and print a 10-page ASCII file to test the PRINT command (from DOS). As I expected, the PRINT command does not have that problem, and it does not have a printer driver at all. I used /L:5 switch (which sets the left margin to avoid the unprintable area), and the PRINT command works just fine.

According to various publications I have read, Windows has always had a printing problem on the LaserJet, which Microsoft has never seemed to be able to correct. I had a vague recollection of reading something to the effect that Hewlett Packard finally wrote a printer driver for Windows that does work, but I called Microsoft first to see if they had a fix. They did not, so I called Hewlett Packard. I talked to a very friendly and extremely knowledgeable technical support representative who knew about the problem, and she said that Hewlett Packard had indeed developed a LaserJet driver for Windows 3. She told me to call the distribution warehouse at (303) 353-7650 and request the FREE LaserJet III printer driver for Windows 3. So I did. And I talked to another friendly service representative who told me that the printer driver would be shipped the same day. I should get it next week, but due to publishing deadlines, I will have to let you know the results in the next column.

### The HP LaserJet III

I recently decided that it was time to upgrade to a new printer and retire my faithful DTC StyleWriter, which has been in service for about five years. I used the StyleWriter to print letters which many of you



have received in response to your questions. I was more than a little reluctant to retire that printer because it has performed quite well and has been extremely reliable. I bought the printer originally at the Fort Worth Heath/Zenith Computer Center after the printer had been discontinued. When I bought it, I also managed to get nearly all of the accessories (except the keyboard), which includes both the cut sheet feeder and the pin-feed tractor. I even figured out how to upgrade the printer's internal memory to 67 K to help reduce the time that the computer is tied up during printing. And when Heath Company had a clearance sale, I bought about 12 boxes (24) of red ribbons and eight extra daisy print wheel cartridges of all kinds for that printer, not to mention all of the nylon and carbon multi-strike ribbons that I have. Part of the reason that I was reluctant to retire the StyleWriter is that it still works just fine, and I always find it difficult to retire something that still works quite well. Due to space, I suppose I will eventually have to sell that printer.

I still have the Epson FX-850 that I use for lots of report and manuscript drafts, and although it works just fine with most of my software (except Samna Word IV), it is quite slow in printing graphics, which I need to do more and more frequently. And although the Epson does a fairly good job with graphics, I really needed a better printer for that.

After reviewing a number of the laser printers, I finally decided to buy an HP LaserJet III for a variety of reasons. Perhaps the most important is that it is supported by virtually all of the software that I use on a regular basis, and the LaserJet III has all kinds of advantages including 1 MB of standard memory and scalable fonts. And although most laser printers have HP emulation in one form or another, the best way to assure that the printer is HP compatible is to buy a Hewlett Packard printer. So I did.

I have been extremely pleased with the LaserJet's performance, and it performs even better than I expected it to. The various manuals supplied with the printer are excellent, and I had no problems with the physical installation. I had the printer running in about 20 minutes after I opened the box and unpacked it. Unpacking it amounted to removing it from the box, setting it on my printer table, reading the Installation Manual, removing the shipping protectors, installing the toner cartridge, and plugging in the parallel cable that was previously attached to my DTC StyleWriter. After plugging in the power cable, I ran the self-test as described in the manual, and then I printed my AUTOEXEC.BAT with the PRINT command. Although that part of the installation was easy, there is one other part of the installation that *MUST* be performed when you add any new printer.

I seem to get more than a few letters about software and printer problems, especially when a new printer is installed. The physical installation of a new printer is generally quite easy as I described above, but there is one additional step that many users forget about. And that step involves installing the correct printer driver in *ALL* software you are using. Installing a new printer generally means that you must dig out the original installation disks for the software and install a new or additional printer driver. Then, you must read the software manual to find out how to install a new printer driver. That takes a while depending on how well the manual is indexed. And then you must actually install the new printer driver and test it. That whole process took me most of the rest of a day, primarily because I have a lot of software that must be "updated" and tested.

Getting a new printer to work correctly with "old" files is a time-consuming task, and it takes some time to get everything working correctly. For example, the LaserJet III obviously has different fonts than the DTC StyleWriter, and it took me some time to figure out what had to be changed to print some old files on that new printer. I had to reread the README files provided with most of my software to find out what the manufacturer's recommendations were for the LaserJet. Another fact about the LaserJet is that it has about a half-inch margin that is "unprintable" on all four edges of the paper, and the spacing was a little different than I was used to on the old printer.

I have found the LaserJet III to be an excellent printer. And I found that Hewlett Packard's technical support is excellent because I had a printing problem with a LaserJet IIP and Quattro Pro while I was traveling, and the technician who helped me fix the problem was extremely knowledgeable. It turned out that the problem was due to my ignorance of laser printers, but that's a story for another article. If you need laser printer performance, the HP LaserJet III is highly recommended. And if you ever need any technical assistance, my personal experience is that the Hewlett Packard's technical support is outstanding.

#### Learning About the LaserJet

I have found that there are generally two ways to learn about something new. The first way is to read the manuals provided with a new item, and the second way is to experiment with it. I have usually found that a combination of the two seems to be most effective for me. Due to my rapidly expanding consulting business however, I have had less time to experiment in the last year or so, and I have had to rely on other means. So, I began looking around in the local bookstores for something about the LaserJet.

There were a few books about the printer,

but after perusing them, I found that *Your HP LaserJet Handbook* by Alan R. Neibauer was by far the best. This 556-page book contains 20 chapters in four parts plus four appendices with all kinds of useful information. Part 1 (An Introduction to the LaserJet) contains four chapters on the basics of LaserJet printing. Part 2 (Using the LaserJet with Your Applications) contains nine chapters on specific application software such as WordPerfect, WordStar, dBase, and spreadsheet printing. Part 3 (Enhancing Your System) contains three chapters on graphics, including scanners and PostScript. And Part 4 (Programming Your LaserJet) contains four chapters on the Hewlett Packard PCL printer language and softfonts. This book is very well organized and written, and it contains a wealth of useful information. If you need a reference on the LaserJet, this book is highly recommended.

If you want an idea of some of the information covered in this book, perhaps you will want to refer to several back issues of *REMark*. If Alan's name seemed to be familiar, that's because he wrote several articles about the LaserJet that were published last year. The article on "Programming LaserJet Printers, Part 1" was published in the June 1990 *REMark* (page 31). Part 2 on "Programming the LaserJet" was published in the September 1990 issue (page 29). And if you are interested in "An Introduction to PostScript", you may find Alan's article published in the August 1990 issue (page 13) to be of interest.

#### Powering Down

All things considered, I was nearly as disappointed with Windows 3 as I have been with previous versions. But I really like the Solitaire game implementation, which is probably worth the \$49 price tag. We will take a look at the very cost-effective alternatives available from Quarterdeck in future articles, and I think you will agree that DESQview and the memory managers provide more features at less cost.

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 systems in general, be sure to include a self-addressed, stamped envelope (business size preferred) if you would like a personal reply to your question, sugges-

tion, comment or request.

### Products Discussed

#### Software

*Your HP LaserJet Handbook* \$24.95  
by Alan R. Neibauer  
Sybex, Inc.  
2021 Challenger Drive, #100  
Alameda, CA 94501  
(800) 227-2346  
(415) 523-8233

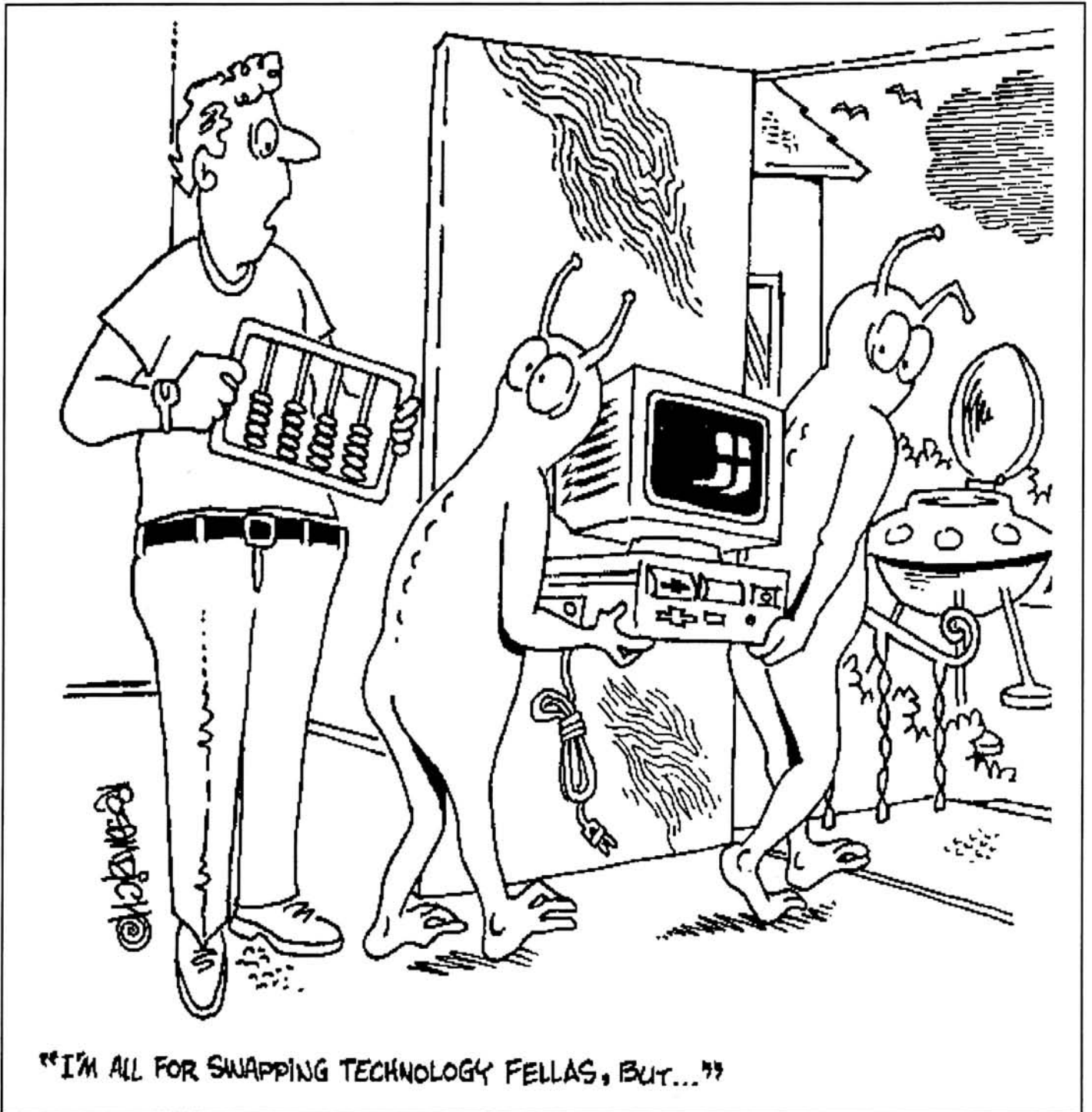
Windows 3.0 149.00

3.5" (720 K) Format (MS-31-28)  
5.25" (360 K) Format (MS-51-27)  
5.25" (1.2 MB) Format (MS-51-28)

Zenith Data Systems  
ATTN: Cash Control  
P.O. Box 1000  
St. Joseph, MI 49085

#### Hardware

HP LaserJet III  
Payload Computers  
15718 Sylvan Lake  
Houston, TX 77062  
(713) 486-0687 (Call for current price)\*





# Enable Revisited

## Part Four Basic Graphics

George Elwood  
1670 N. Laddie Court  
Beavercreek, OH 45432

In the last article, I covered the enhancements that have been added to Enable OA. In this article I will carry this one step further by going into the graphics area. In my first series of Enable articles a couple of years ago, I touch briefly on the graphics capability of Enable. I will go into greater detail on basic Enable graphics in this article and cover the Perspective color graphics option in the next article.

As a legal owner of LOTUS 1-2-3 on my Z-100 (yes, there was a version of LOTUS for the Z-100) I became quite proficient in its uses, including the graphics. I was using Lotus to develop parts of a business plan. The plan required five years of projected data and graphics to present the information. Using LOTUS, I had to create my graphs one at a time and then save them as a file. I had six graphs per year in the presentation and LOTUS would only save the last graph settings with the spreadsheet. After creating and saving the graphics pictures, you had to leave the spreadsheet and move to the graphics section and print them. You could print all of the graphs at one time but any changes to the spreadsheet required that the setting, including the headings be reconstructed. It would have been possible to build a macro to do this for the charts as an automation function.

When I switched to Enable, I was pleasantly surprised to find out that Enable saved eight graphs settings with the spreadsheet. I could now change my spreadsheet and all of the graphs would be updated. I feel that this is one of the biggest pluses using the Enable spreadsheet graphics function.

To create a chart in Enable, you must be in the spreadsheet and have data available. You can also be in the database, as Enable permits you to create and save graphs using database information. Pressing the F10 or slash (/) key will display the Top Line Menu. On this first screen of the menu is the Graphics option. To select it, move the highlighted bar over "Graph" and press <Enter> or simply press the letter

"G" (see Figure 1).

Enable will now display the graphics options. On this first line you can Select a graph that already exists, Create a new graph, Delete a graph that exists, or Copy an exiting set of graphic options. You would use Copy to save you from having to creating another graph that is similar to an already existing one (see Figure 2).

If this is the first graph you are creating, you should select C(reate). Enable will prompt for a graph name. This can be any eight characters. After inserting a name and pressing the <Enter> key, Enable will take you to the main graphics menu. As you select from this menu, more options will be displayed (see Figure 3).

Note that Enable displays the name of the graph that you are creating in the upper left corner of the screen. In the upper right corner is the graph number you are working on. Remember, you can have up to eight graphs per spreadsheet in Enable.

The first thing you will need to do is define the data elements you wish to use for the graph. To select these, press <Enter> with Options highlighted. Enable will now move down one menu level. Here you will select the Data Group option 1-8. Enable permits you to select and display up to eight data sets. Press the number "1" or highlight "1" and press <Enter> (see Figure 4).

Enable will now display the Group options menu. You need to select Data, so highlight and press <Enter> or press the letter "D". Enable will now permit you to select the range for your graph. You can type in the inclusive cells or move the to first cell in the block, press the <Enter> key to lock the cursor and move to the last block. Enable will highlight all selected cells during this procedure. Press <Enter> when all cells have been selected. You can also set the color and insert a legend of this data group if desired. Press {ESC} to move back one level to select another data

```
ES: (F,2) @RAND*B5                                MENU
Worksheet MCM Print Graph Save Combine DBMS Link Quit
Display range(s) of cell values in a graph
  A B C D E F G H
1
2
```

Figure 1. Top Line Graphics Option

```
Select Create Delete 1=Copy
Create a new graph setting
  A B C D E F G H
1
2
```

Figure 2. First Graphics Options

```
Select BAR1                                X12345678
Options Display Print L=Plot E=Perspective{tn}
Change current graph settings
  A B C D E F G H
1
2
```

Figure 3. Enable Graphics First Menu

Select BAR1 Options X12345678  
 Global Data Group:  2  3  4  5  6  7  8 Device  
 Change options that apply to entire graph

	A	B	C	D	E	F	G	H
1								
2								

Figure 4. Options Menu Selections

Select BAR1 Options Group X12345678  
 Data Scale Color 1=Shade Legend Reset-data Explode 2=Symbol  
 Specify Spreadsheet range or DBMS field for this group

	A	B	C	D	E	F	G	H
1								
2								

Select BAR1 Options Group Data  
 Data Scale Color 1=Shade Legend Reset-data Explode 2=Symbol  
 Enter range for data group:

	A	B	C	D	E	F	G	H
1								
2								

Figure 5. Data Selection Menus

Select BAR1 Options Global X12345678  
 Type Grid Scale Axis Headings Fonts Color XYL-format Ign-pts Layout  
 Specify primary and secondary MAIN, X-AXIS, and Y-AXIS titles

	A	B	C	D	E	F	G	H
1								
2								

Figure 6. Global Options Menu

Select BAR1 Options Global Headings Main X12345678  
 Enter Main title:

	A	B	C	D	E	F	G	H
1								
2								

Figure 7. Main Title Menu

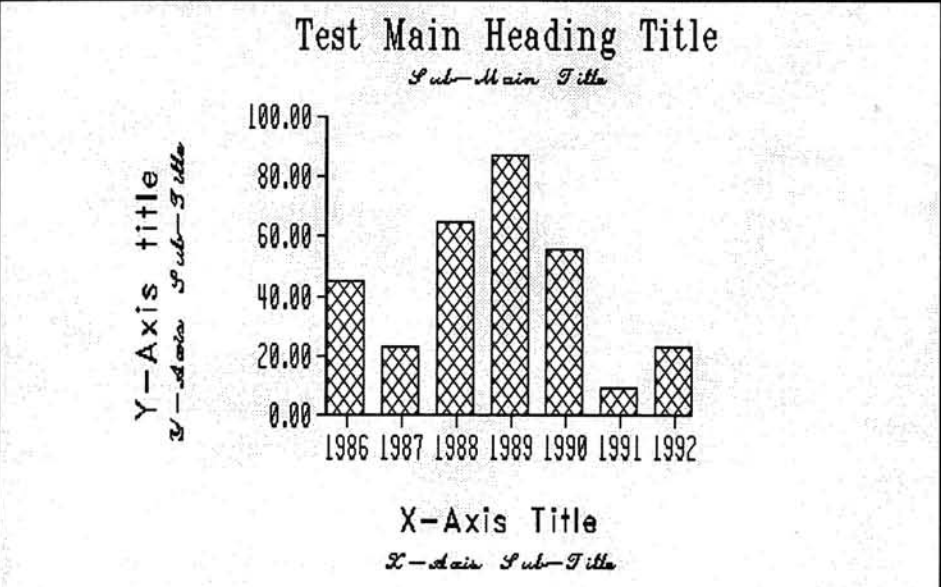


Figure 8. Basic Graph with all Titles Shown

group if necessary (see Figure 5).

Before we display the graph, we will add the titles, sub-titles and other text. You should now be at the Options Global menu. Select Global to move down one last level. One this level, select Headings (see Figure 6).

When you select Heading, Enable will prompted for the type heading desired. Select Main and insert the main title. Enable permits you to insert Main, Sub-Main, X-Axis, Sub X-Axis, Y-Axis, and Sub Y-Axis. After inserting the desired titles, you can press the [ESC] key until the Display option is shown. Select that and your chart is now displayed on the screen. The graph will not be in color unless you specified the colors in the Group option. The graph will be displayed as a bar chart which is the default setting. This function is probably one of the most confusing. You must move down the menu to for each label. In the new version 4.0, all of the labels are shown and can be inserted on one screen (see Figure 7).

You also have the option to change the fonts of groups of titles. Figure 8 is a basic graph with all of the titles displayed and different font styles selected for each group.

I have included a sample of graphs you can make in Enable. A new graph type in Enable OA is the linear regression graph type. This is included in the samples below.

In addition to the normal outputs to a printer or plotter, Enable OA now support plotter output to file. This selection will generate a plotter file that can be taken to another computer for final output.

One of the key features of Enable is its ability to copy files between applications. A key stroke of so will move the graph you generated in the spreadsheet or database into a word processing document. If you have a graph to move and you need to resize it for the other document, pressing F10 will display the Window options menu. Select Modify permits you to change the size of the graph. In earlier version of Enable, you have to manually select the redraw after resizing the window. In OA, this is handled automatically (see Figure 9).

Another feature is that Enable permits rapid "what if's". After you have generated a graph, it is possible to display it and the spreadsheet on one screen. This will permit you to make changes in the spreadsheet and see the result immediately on the graph. To use this procedure, you must have completed a graph. Return to the spreadsheet and press Shift/F7. This will display the spreadsheet on the left side of the screen and the graph on the right.

Included with this article are samples of charts that can be created with Enable's graphics function. Remember, these charts can be created in both the spreadsheet and database modules of En-

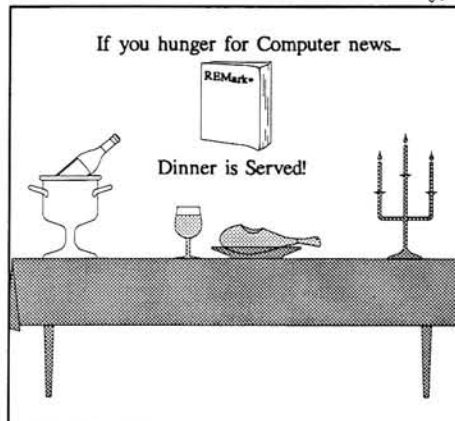


able. The various type charts can be selected by using the Global Type function in the graphics menus. The enclosed charts provide you with a visual presentation of what is possible in Enable basic graphics.

The screen displays in this article were created using a graphics screen saver program available from Software Publishing. It is an add on package to Harvard Graphics. The screens were saved as .PCX files and then included with the text using the graphics import capability of Enable.

(Note from Patrick Swayne: For the magazine version of this article, I re-created the screens in Figures 1 through 7, and I used Mr. Elwood's .PCX files for figures 8, 9, and 10. If you would like to include screen images in an article you are preparing for REMark, I have written a program called SCIM that can save any text mode or standard CGA, EGA, VGA, or Hercules monochrome graphic screen as a GEM .IMG file. These files can be directly imported by Ventura Publisher, or they can easily be converted to .PCX files for import by Page Maker. We have been using both to prepare this magazine. SCIM is available for download as SCIM.ZIP from the ZLink-COM1 BBS.)

This concludes a look at Enable's basic graphics capability. In the next article I will look at Perspective, the super graphics function available in Enable.



Bottom of this page and the next page: Sample charts (graphs) that you can create with Enable basic graphics.

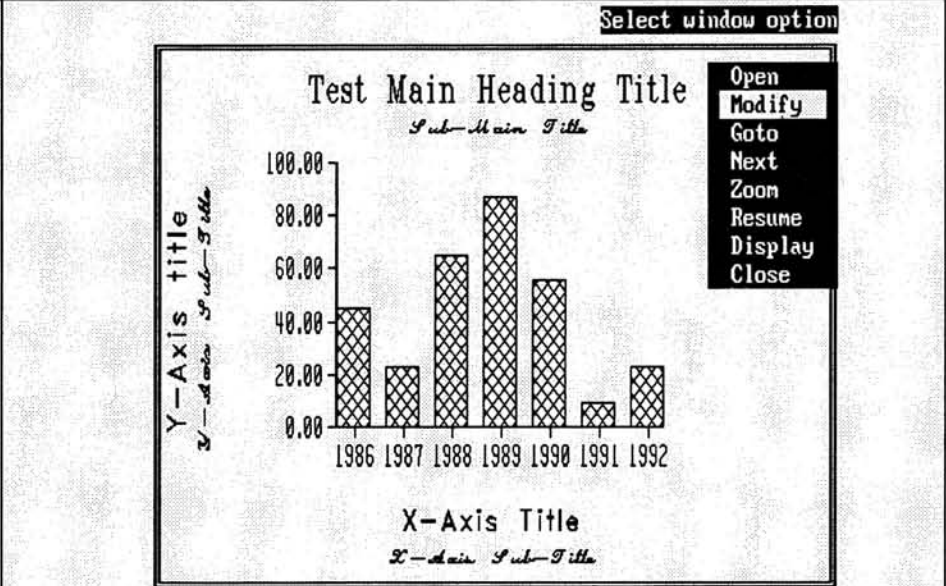
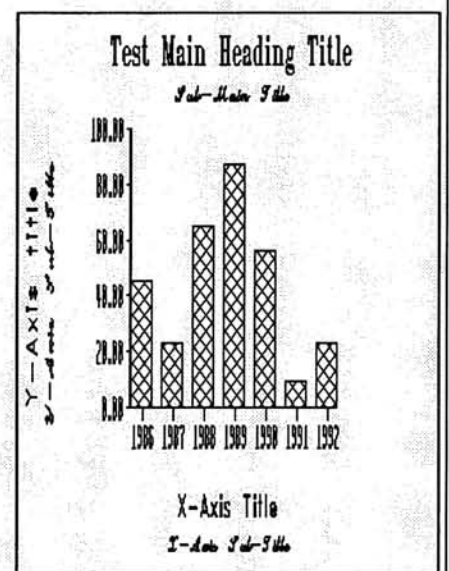


Figure 9. Window Modify Options

CS: (F,2) @RAND\*100

READY

	A	B	C
1			
2			
3			
4			
5	45.00	67.00	74.93
6	23.00	81.00	80.70
7	65.00	63.00	4.31
8	87.00	76.00	1.85
9	56.00	34.00	33.36
10	9.00	98.00	49.94
11	23.00	56.00	42.91
12			
13			
14			
15			
16			
17			
18			

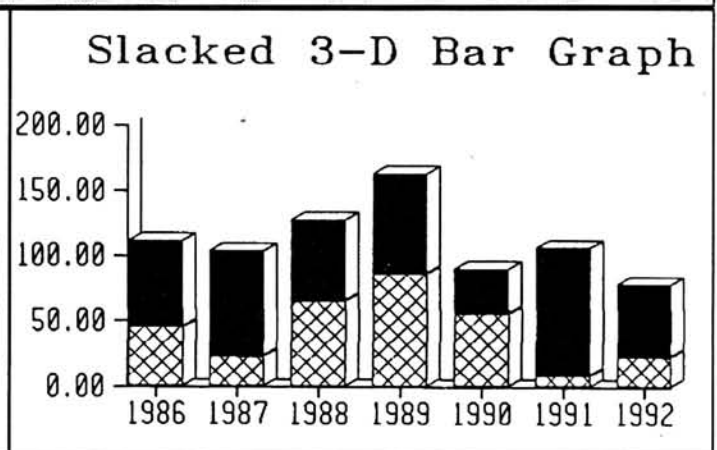
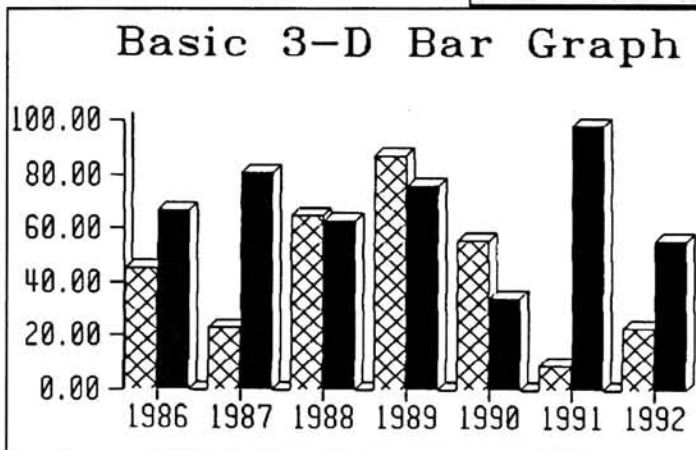


#1 C:\EN300\REMARK.SSF

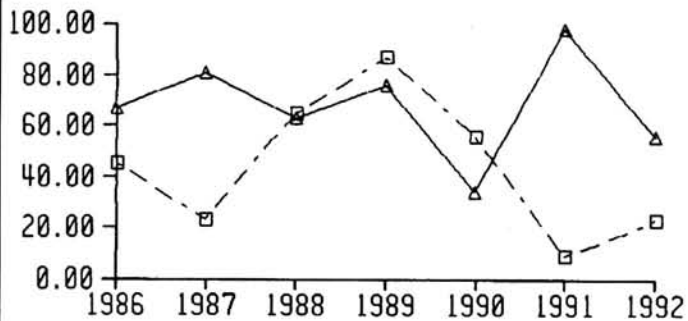
Gr

F11

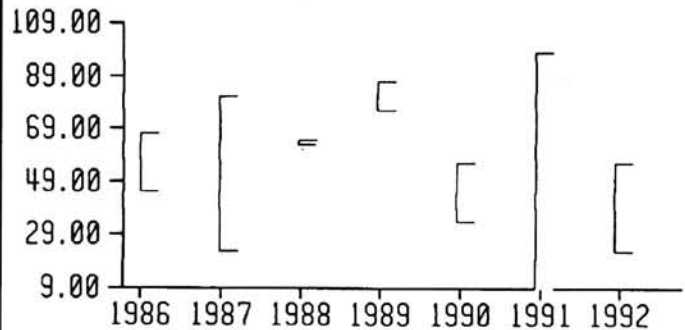
Figure 10. Spreadsheet/Graph Display



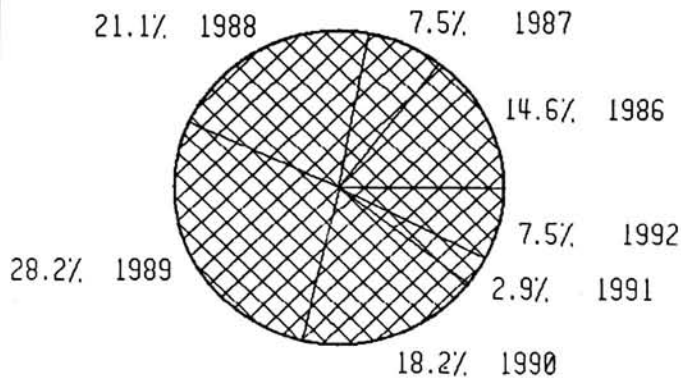
### Basic Line w/Symbols



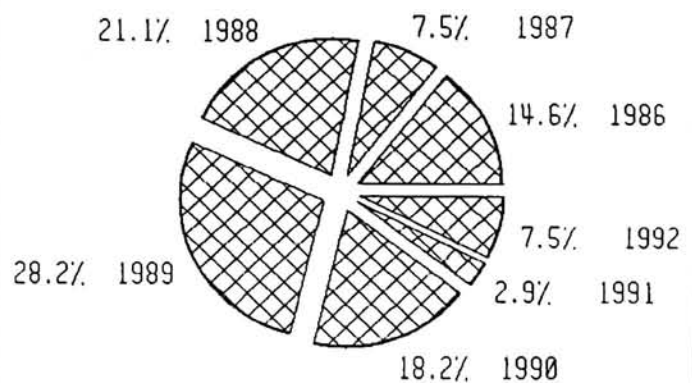
### Hi-Lo Closewd Graph



### Basic Pie Graph



### Exploded Pie Graph



### EGAD Screen Print

#### Now Available for Z-100 -or- PC

Graphics & Text Screen Print for VGA, EGA, CGA displays (PC version), or Z-100 series.

- \* Print any part of the screen - crop box pops up when Shift-PrtSc (PC) or Shift-F12 (Z-100) pressed; use arrow keys to select region.
- \* Enlarge graphics 1-4 times
- \* Prints in color (color printers) or black & grays
- \* SET program selects printer colors, other options.

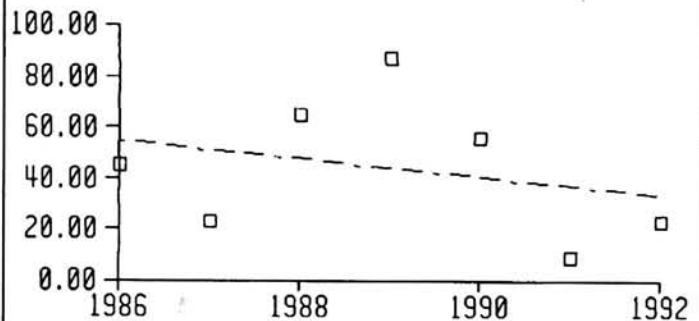
Supports most popular dot matrix, laser, and ink jet printers (including Epson, NEC-8023, MPI, Okidata, etc.) EGAD for PC's, Order # 270; EGAD for Z-100, Order # 271. Either is \$35.00 postpaid.

### Z-Screen Library for 'C'

Screen & Keyboard I/O library for Microsoft 'C'. Two sets of source code included (one makes programs which will run on PC; the other, for Z-100). Both libraries have same interface to your code. No royalties for use. Call or write for complete description. Z-Screen, #246, \$20.00

**Lindley Systems—4257 Berwick Place,  
Woodbridge, VA 22192-5119 (703) 590-8890**  
Call or Write for Free Catalog. Reader Service #136

### Linear Regression



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



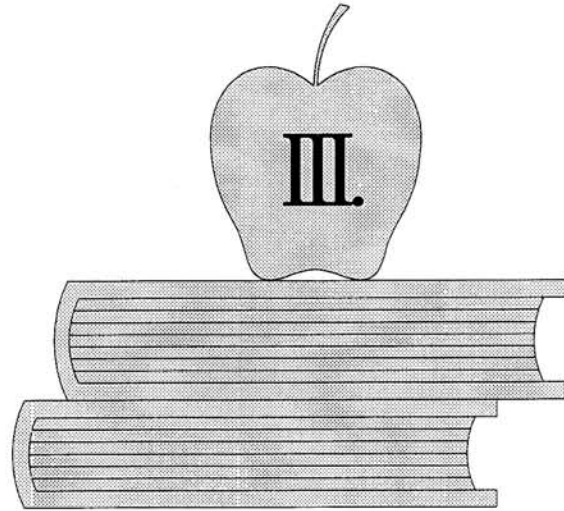
---

---

# Introduction to C++

## Third Installment

Lynwood H. Wilson  
2160 James Canyon  
Boulder, CO 80302



C++ is a fine and useful language even if you don't use the Object Oriented Features. (Would that be OOF?) The class I teach (Programming in C) just started and already I've caught myself using C++ features by mistake. And several useful C++ enhancements found their way into ANSI standard C. I think C++ will start replacing C even before we are all comfortable with the Object Oriented Paradigm.

### Resources

The Annotated C++ Reference Manual by Margaret A. Ellis and Bjarne Stroustrup is an excellent book of a kind which is seldom seen. It is a complete reference manual for the language, version 2.1, but the annotations make it much more than that. The book explains why some of the elements of the language are as they are, how one might implement some of the language features, and with explanations and sample code, how the language is intended to be used. It is not (nor is it intended to be) a good book to learn the language from, but it is an indispensable reference to answer the difficult questions. And it provides an insight into the thought processes which produced C++ which I find fascinating as well as useful. No wonder it was chosen as the ANSI base document for the language, the beginning from which ANSI standard C++ will be developed. You will want this one if you plan to continue on with C++.

Turbo C++ is going to do more for the popularity of C++ than anything that has happened so far. The entry of one of the major players legitimizes the language in a way that no amount of theoretical or practical usefulness could. It is safe to use it now. It's no longer a fringe phenomenon.

It's OK. Fortunately Turbo C++ seems to be a very good product. I haven't had a chance to do much with it yet, but I like it. The editor is quite an improvement over the Turbo C editor.

I still use the Zortech C++ compiler too. I am still more familiar with it, and it is very useful to be able to compare two compilers to find out whether some oddity you have discovered is a compiler bug or a language feature. The Zortech compiler seems to produce much smaller programs. It is an excellent product and I am sure it will hold on to a fair share of the market.

BRIEF is my favorite programming editor. In addition to all its other features, it works very nicely if you are using two compilers. It can compile from within the editor, just like the integrated environments, and return the error messages, and it can pick the compiler to be called based on the file extension. I can write a program in Brief called TST.CPP, and compile it under Turbo C++. Then I make a copy under the name TST.CXX, and compile it under Zortech C++. Very convenient.

For those of you who may be using a similar setup, there seems to be a bug somewhere which can interfere with the error messages. I was working on a program called 3.CXX (to compile under Zortech) and the error messages did not get back to BRIEF. They were there in the file 3.ERR, but did not get displayed. All I got was the file extension "CXX" on the message line. I changed the name to THREE.CXX and it worked fine. No problems under Turbo with the name 3.CPP either. Apparently an interaction between Zortech and BRIEF.

Enough about the tools, let's get back to the language.

### Mathematical Operators

We've been using some of the mathematical operators without talking about them, but now it's time for a formal introduction.

There are only five mathematical operators which are part of the core language. There are many more in the libraries, and more yet can be purchased for special purposes. The five are:

- + the addition operator,
- the subtraction operator,
- \* the multiplication operator,
- / the division operator, and
- % the modulus (remainder) operator.

The first four operators work pretty much the way they did in high school. Multiplication, division, and modulus have a higher precedence than addition and subtraction.

$7 + 3 * 2$  evaluates to 12 not 20 because the multiplication is performed first.

Parentheses override precedence. When in doubt about precedence, use parentheses to make it clear.

$(7 + 3) * 2$  evaluates to 20.

Integer division truncates. The fractional part of the answer (if any) is lost.

$6 / 3$ ,  $7 / 3$ , and  $8 / 3$  all evaluate to 2.

The modulus operator can only be used with integral operands. It performs a division and returns just the remainder.

$6 / 3$  evaluates to 0.  $7 / 3$  evaluates to 1.

The answer is the same data type as the operands, in this case the decimals force the operation to be floating point.

$7.0 / 2.0$  evaluates to 3.5

If the operands are of different types, the simpler is temporarily promoted to the

larger more complex datatype and the answer will be that type.

7 / 2.5 evaluates to 2.8

All these things work the same with variables, but it is easier to show data types with constants

### Shorthand

To add 5 to an integer variable named *x*, you could do this:

```
x = x + 5;
```

and it would work fine. However, there is a shorter way of doing the same thing and it is:

```
x += 5;
```

These two lines do exactly the same thing, the only difference is that one has fewer characters in it.

This works with the other mathematical operators too.

```
x *= 5; is the same as x = x * 5;
```

```
x /= 3; is the same as x = x / 3;
```

```
x -= 2; is the same as x = x - 2;
```

```
x %= 2; is the same as x = x % 2;
```

By now, if I asked you for a short way to add one to *x*, you would not be fooled and would surely answer "*x* += 1;". Very good, you have been paying attention. However, there is another bit of shorthand in the form of the decrement and increment operators.

*x*++; is the same as *x* += 1; is the same as *x* = *x* + 1; and likewise for *x*-. But this is a bit more complicated, because the ++ or -- can be before or after the variable. In the case of ++*x* the variable is incremented before it is used, and in the case of *x*++ the variable is incremented after it is used.

```
x = 5;
cout << x++;
```

This will print 5 to the screen, and then increment *x* to 6.

```
x = 5;
cout << ++x;
```

This one will increment *x* to 6 first then print 6 to the screen. Which brings up another point . . .

```
x = 5;
cout << "first x = " << x++ << "second = " << x;
```

We cannot predict what the second *x* will be when it is printed because in most cases the language does not guarantee the order in which a line of code is evaluated. In order to avoid this kind of bug, which can be quite hard to see, do not use the increment or decrement operators in a line where the variable you're modifying appears more than once. For that matter, the same applies to any change in the value of a variable.

As we will see, some statements do guarantee the order of evaluation, and so it is safe to change the value of a variable which is used more than once.

### Input

Stream input is similar to the stream output we used in the first installment.

Note that the header file which contains

the stream declarations has different names in Turbo and Zortech. The rest of the program is the same.

```
// INANDOUT.CPP input and output
// Zortech V 2.0
```

```
#include <stream.hpp> // #include <iostream.h> for Turbo
```

```
main()
{
    int x;

    cout << "Please enter an integer. ";
    cin >> x;
    cout << "The square of " << x << " is " << x * x <<
        ".\n";
}
```

*cin* is a pre-declared input stream similar in all but direction to *cout*. It brings in data from the standard input (usually the keyboard) and the ">>" operator tries to fit it into the variable "*x*". It does a pretty good job, as long as the input makes sense.

After you get this program running, try to confuse it with bad data. Try entering "7.3" or "r5" or "six" or anything else a user might come up with. You will quickly discover the reason that this kind of input is rarely used in real programs.

The program puts some value into the variable "*x*" for almost any input you give it, and there is no way to know whether the input made sense or not. Soon we will learn ways to check the input.

In addition to stream I/O, C++ retains all of the standard C I/O functions. (All I/O in both languages is performed by library functions, not by the core language.) *printf()*, *scanf()*, *getch()* and *puts()* are still available. The new stream I/O offers better data type checking and improved readability and so the old I/O functions are seldom used in C++.

Note that the input routine fills the first variable with the first integer and

What is worse, programs from the two compilers do not agree. Entering "r5 3" for the Zortech program gives

"The sum of 1132 and 590 is 1722." The same data to the Turbo program gives "The sum of 8717 and 0 is 8717." I suppose there is no good reason why they should give the same results from bad data, but I find it disquieting.

Documentation on the workings of the stream input routines is oddly scarce. I have six books on C++ and none of them explain it completely. Perhaps that is why the compiler writers produced different results. Both Turbo and Zortech C++ offer the source code for their run time libraries, so after we learn a bit more about the language we can read the source and figure out what it actually does.

### For Loops

Anything worth doing is worth doing repeatedly. The for loop is the most popular way of repeating something in C++. It is much more versatile than the for loops in most other languages, and lends itself to tricky programming which is difficult to read. Given the perversity of many C (and C++) programmers, about 90% of the loops in C are for loops.

There is an old tradition in C, going back to Kernighan & Ritchie's "The C Programming Language", to use *i* as a loop counter.

```
// MOREIN.CPP input and output
// Zortech V 2.0
```

```
#include <stream.hpp> // #include <iostream.h> for Turbo
```

```
main()
{
    int x, y;

    cout << "Please enter two integers. ";
    cin >> x >> y;
    cout << "The sum of " << x << " and " << y << " is " << x + y <<
        ".\n";    h0
}
```

the second with the second and then returns to the program.

If you try to confuse this one, you will find that the input routine isn't always sure where one number ends and the other begins.

If you need more counters, *j* and *k* are often used.

The for loop consists of five elements. First there is the word "for". This identifies the construct as a for loop. Second is the initialization expres-



```
// FORLOOP.CXX Zortech V 2.0
```

```
#include <stream.hpp> // #include <iostream.h> for Turbo
```

```
main()
{
    int i;

    for(i = 0; i <= 10; i++)
        cout << i << " squared is " << i * i << ".\n";
}
```

sion or initializer. In the case above it is "i = 0".

Third is the test, in the case above "i <= 10".

Fourth is the modifier, sometimes called the increment expression. In the case above, i++.

Fifth is the body of the loop, the entire line beginning with "cout" above. As in the while loop, the body is indented because it is controlled by the line above it.

This whole thing is a single statement, and has a semicolon at the end. Note that there is no semicolon after the line beginning with "for".

The initializer is executed once when the loop begins, and never again.

The test is then executed, and if false the body of the loop is skipped and control passes to the next line of code after the loop. In this case, the test is true so...

The body of the loop is executed.

The modifier is executed.

The test again,

if false end the loop,

if true, the body

the modifier

the test

and so on, until the test is finally false.

The for loop expects its body to be one statement. If you want more than one, wrap them up in curly braces and they will look like one statement from outside.

If you mistakenly put a semicolon after the first line of a for loop...

```
for(i = 0; i <= 10; i++); // WRONG!
    cout << i << " squared is " << i * i << ".\n";
```

the line beginning with "cout" will not be the body of the loop and will be executed only once, after the loop ends.

The null statement (a statement with nothing in it, an empty statement) is a valid expression in C++. Putting a semicolon after the first line of a for loop creates a null statement as the body of the loop.

In the case that you want to write a for loop with no body, it is customarily done like this:

```
for(i = 0; i < 10; cout << i++)
```

In this case it is clear that the null statement is the body of the loop, and the code is much easier to read.

A smart editor which automatically indents your code is a help here. If you

mistakenly put a semicolon after a for statement the editor will not indent the following line, calling your attention to the fact that it is not controlled by the for. If you forget the curly braces around the body, only the first line of the body will be indented.

### Variations on the For Loop

Any of the three elements in the parentheses after the keyword for can be omitted, but the semicolons must remain. For example, we could rewrite the first example like this:

```
main()
{
    int i = 0;

    for(; i <= 10; i++)
        cout << i << " squared is " << i * i << ".\n";
}
```

```
main()
{
    int i = 0;

    for(; i <= 10;) {
        cout << i << " squared is " << i * i << ".\n";
        i++;
    }
}
```

and it would work just the same. We could also do this:

Note the curly braces which enclose the two lines of the body of the loop. There are several equally readable styles of indenting and positioning curly braces. I think consis-

tency is more important than the particular style you choose.

In this form the for loop is just a while loop with a different syntax.

In C++ (but not in C) it can also be done like this:

```
main()
{
    for(int i = 0; i <= 10; i++)
        cout << i << " squared is " << i * i << ".\n";
}
```

One way to write an endless loop in C++ is:

```
for(;;)
    body of loop
```

although I think it is clearer to write

```
while(1)
    body
```

perhaps because it is not obvious to me

that the nothing in the test position of the for loop will evaluate to true.

Since any of the elements in the control section of the for loop can be omitted, it seems reasonable that they could be replaced with expressions which don't perform the normal functions of the three elements. For example, this loop.

```
for(cout << 'A'; cout << 'B'; cout << 'C') {
    cout << 'D';
    cout << '\n';
}
```

Note that a single character has single quotes around it, instead of the double quotes used for a character string. Also note that '\n' is a single character rather than two.

This loop prints

```
ABD
CBD
CBD
CBD
etc.
```

until you stop it with CONTROL C. The expression in the initializer position is no initializer but an output statement. The test, modifier and body are likewise output statements. The for loop doesn't care. It executes whatever is in the first position first (and never again). Then it executes whatever is in the second position and

takes the result as false if it is 0 and true if it isn't.

Almost every expression evaluates to some number. For example, an assignment expression (x = 5;) evaluates to the number being assigned (5). I haven't found

any mention in the documents of what this output statement should return, but it is not zero.

Then if the result from the statement in the second position was not zero, it executes the body. Then it executes whatever is in the third position. Then the second position. If it is true, the body. Then the third position. And so forth. If the statement in the second position ever evaluates to 0, then the loop ends.

We think of the three positions in the control section of the for loop as initializer, test, and modifier, and they are usually used that way, but we can fill the places with anything we choose. The test always controls the repetition of the loop, but the other two (along with the body) are just executed at the appropriate times.

It is also legitimate to put more than one

expression in any of the three sections of the for line by using the comma operator to separate them. Here is an example.

```
for(i = 0, j = 0; j--, i < 10; i++, cout << j)
    cout << "\nI is " << i << ".\n";
```

When two expressions are separated by a comma, they are executed left to right and the value of the whole thing is the value of the second expression. The first expression is evaluated for its side effect.

For example, look at the two expressions in the test section above. The `j--` is evaluated and `j` is incremented but it has no direct effect on the `for` loop. Then the `i < 10` is evaluated and its value, 1 or 0, is used to determine whether to continue or end the loop.

Whenever you write a non-standard `for` loop like these remember to make it readable. If the code doesn't make sense use a comment or two. Odds are the next person who will have to work on it is you.

### Do While

The `do while` loop is just a `while` loop with its test at the end instead of the beginning. Here's an example.

```
// THREE.CXX Zortech V 2.0
```

```
#include <stream.hpp> // #include <iostream.h> for Turbo
main()
{
    int i = 0;

    do {
        cout << i << " squared is " << i * i << ".\n";
        i++;
    } while(i <= 10);
}
```

The only functional difference between this and the `while` loop is that it performs the loop once before making the first test. After that it's test, body, test, body, and they are both the same. This is probably the least used of the three loops.

Note that the body of the loop, which consists of two lines, is enclosed in curly braces. Surely the compiler could figure it out, since the body is contained between the `do` and the `while`, but the curly braces are kept for consistency.

Here is another common variation of the above loop.

```
do
    cout << i << " squared is " << i * i << ".\n";
while(++i <= 10);
```

The plus signs are before the `i` so `i` will be incremented before it is tested, just as it is in the example above. If the plus signs are after the `i` the loop will print one more line, the square of 11, before it ends.

### Choosing a Loop

Why do we need all these loops anyway? After all, Alan Turing proved in the 30's that all loop constructs are interchangeable, and we have seen how to simulate a `while` loop with a `for` loop.

Even though we can perform all the functions with any one of the loops, it is more convenient to have three. It also enables us to write shorter and more readable code.

As a very rough guideline for choosing a type of loop, I offer these rules:

If you know in advance how many repetitions of the loop are to be performed, use the `for`.

If you don't know in advance, but the data upon which you will decide whether to run the loop is available at run time, use the `while`.

If you may not want to run the loop at all depending on the data available at run time, use the `while`.

If you need to run the loop once regardless of the data and then decide whether to run it again, use the `do while`.

If the data upon which you decide whether to run the loop is generated in the body of the loop, use the `do while`.

When in doubt, use the `for` like all the rest of us.

### Constants

It is often useful to use constants in a pro-

gram. They can be established at the head of the code so that they can be changed easily if desired but they cannot be changed by mistake.

In many cases this can improve the readability of the code, to deal with an appropriately named variable rather than a number. A number buried in the code with no apparent referent is often called a magic number. It takes real magic to know what it does, and we try not to program like that.

There are two main ways to do this. The most common way in C is with a `define` which is handled by the preprocessor.

```
#define LIMIT 10

main()
{
    for(int i = 0; i <= LIMIT; i++)
        body
}
```

In this case the preprocessor which runs before the compiler replaces all instances of `LIMIT` with 10. This is a simple text replacement, just like your word processor does.

Traditionally these are written in upper case so you can easily tell when reading the program that they are not variables, which are traditionally written in lower case.

Incidentally the preprocessor does a lot of other interesting things which I won't have room for here. Check it out after class.

The other way of doing the job is by declaring a constant.

```
const int limit = 10;
```

```
main()
{
    for(int i = 0; i <= limit; i++)
        body
} =
```

This is preferable, primarily because the compiler then knows what data type `limit` is and can use that information in type checking. For example, if `limit` is passed to a function which expects a float, the compiler can produce an error message.

Constants are particularly valuable for numbers you might want to change all through the program. For instance, you might write a program to run on 80 column displays and use a constant called `display_width` which is set to 80 at the head of the program. If you should ever want to modify the code to run on a 40 column monitor, it's a lot easier to change it once at the constant declaration than to hunt through the code for every 80 and accidental change one to 40 which had nothing to do with the display.

### The End

You know enough now to write complex and useful programs, and I encourage you to do so. In order to learn to write programs you must write programs. As you use the language you will become more comfortable with it, and you will evolve your own style and habits. Gradually you will think less about the language and begin to think in the language, begin to see your task in terms of the constructs you will use to perform it. Nothing for it but practice. After all, it's too cold to go sailing.

### Sources

The Annotated C++ Reference Manual by Bjarne Stroustrup and Margaret Ellis Addison-Wesley, 1990.

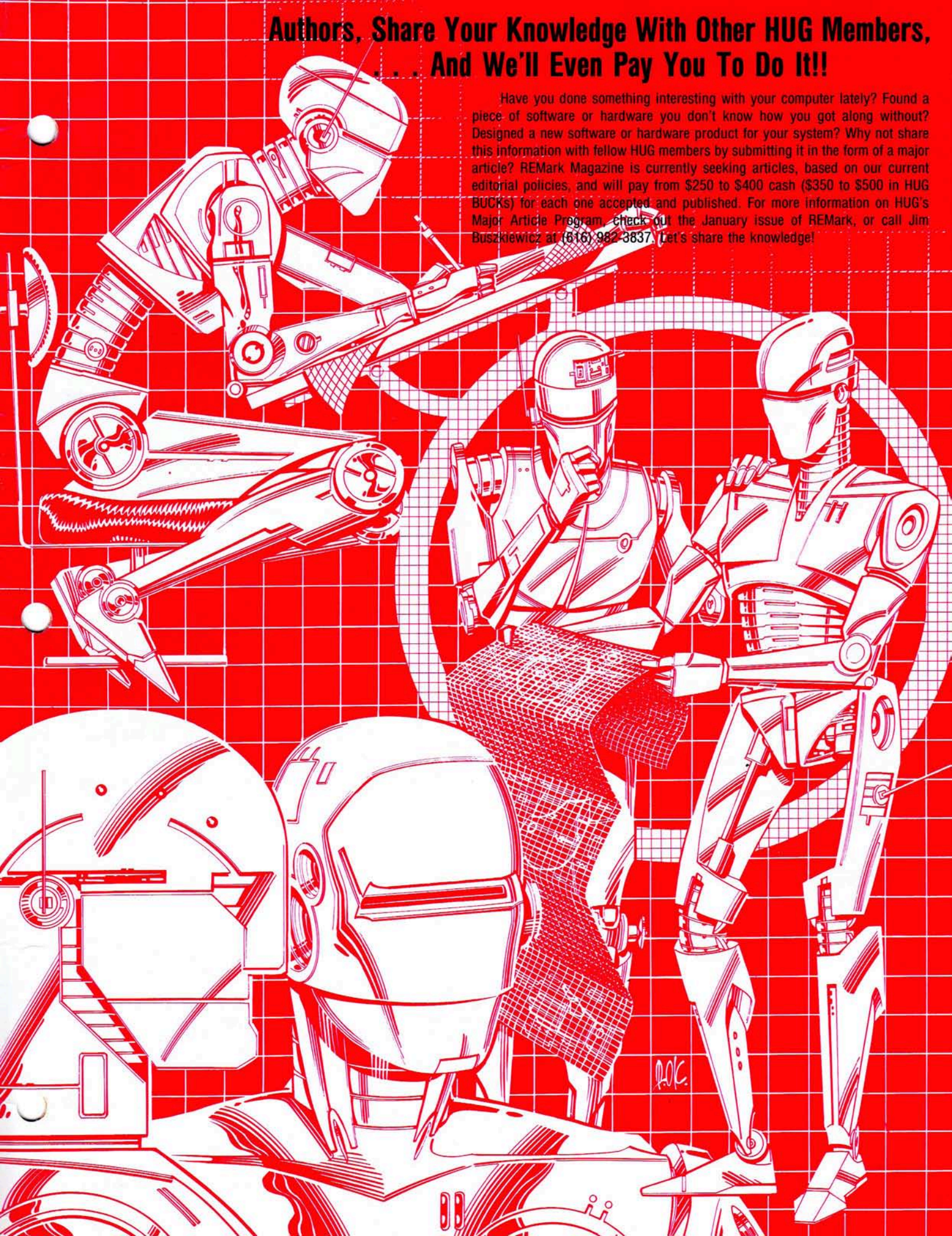
BRIEF Solution Systems 541 Main St., Suite 410 South Weymouth, MA 02190 (800) 821-2492	\$199.00
Turbo C++ Turbo C++ Professional (Includes debugger, profiler and assembler.)	\$199.95 \$299.95
Borland International Inc. P.O. Box 660001 Scotts Valley, CA 95067-0001 (800) 331-0877	
Zortech C++ compiler Zortech Developer's Edition Zortech Inc. 4-C Gill St. Woburn, MA 01801 (800) 848-8408	\$199.95 \$450.00





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

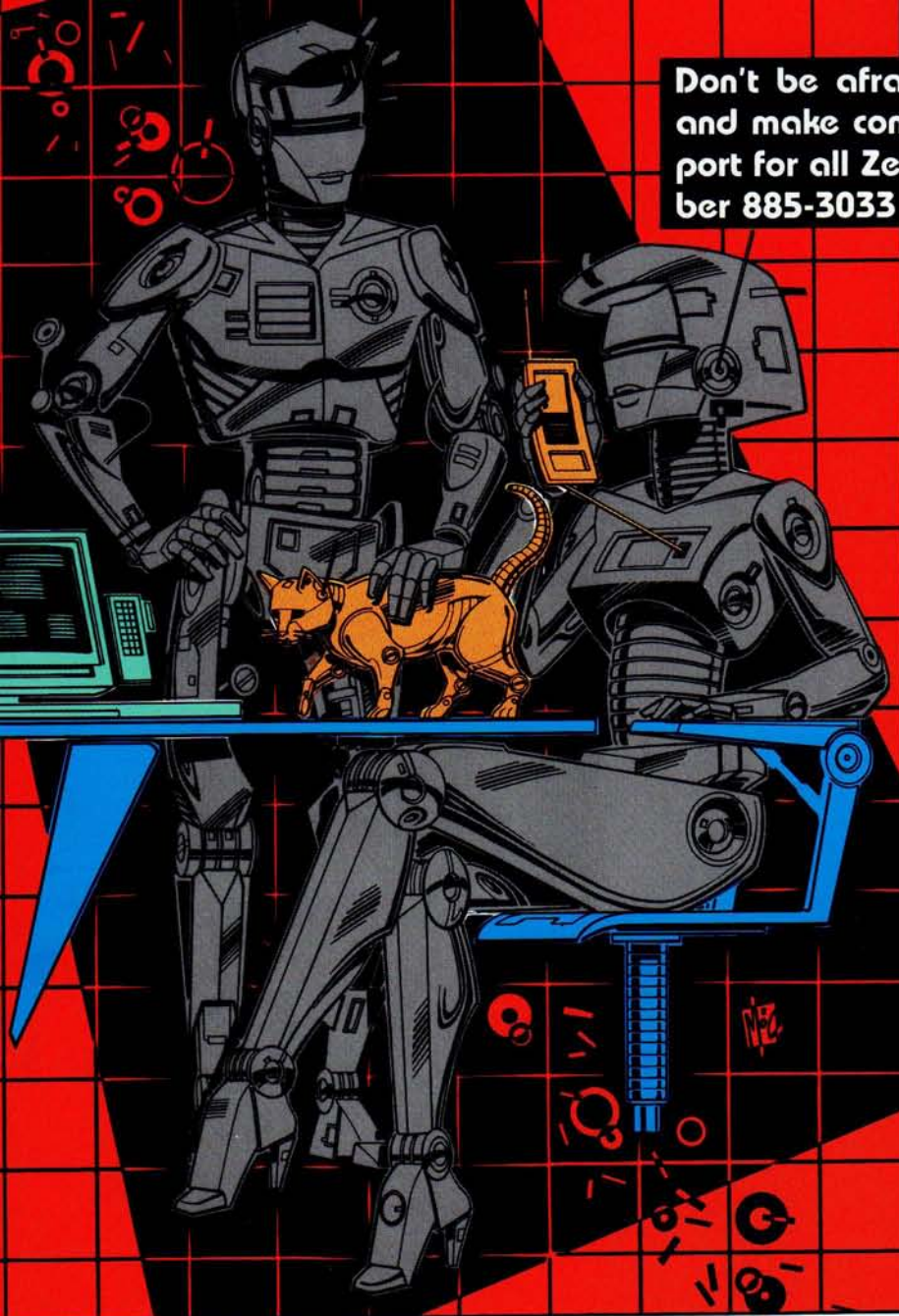
Have you done something interesting with your computer lately? Found a piece of software or hardware you don't know how you got along without? Designed a new software or hardware product for your system? Why not share this information with fellow HUG members by submitting it in the form of a major article?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



BOC



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



```
HUGMCP Commands
F1 -- Points This List, Your Storage Buffer Size, And How Many
    Bytes Are Presently In The Storage Buffer.
F2 -- Allow 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 (Buf) 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-0), (F1) To List, Anything Else To Abort --> _
F1:Wlp F2:Msg F3:Buf F4:Save F5:Send F6:Cfgp F7:Clr F8:Print F9:Exit COM
```

```
HUGMCP Configuration Help #1
1 This Function Allow The Baud Rate To Be Changed, Depending Upon Which
  Modem You Are Using. Normally, 2400 must Be 1st To Either 300, 1200, Or
  14400 Baud. Correct Connection As A Host. All Allow Higher Baud Rates.
2 This Selection Allow You To Change The Word Parity. Normally, you
  Should Use Even Parity. But It Acceptable By Most Remote System
  And It Is Not Necessary For XMODEM Protocol To Work Properly.
3 This Function Allow The Changing Of The Word Length. Normally The
  Length Should Be Set To 8 Data Bits. This Value Is Acceptable By Most
  Remote System, And It Necessary For XMODEM Protocol To Work Properly.
4 This Selection Allow You To Enter Messages Which Can Be Automatically
  Sent With The (F2) Key. Up To 14, 32-Character Messages Can Be Saved.
  Selection (0) Is Special. It Should Contain Your Computer's IP Number
  and Network Selection (0) Is Also Special. This Selection Can Auto-
  matically Be Sent When This Program Is First Executed By Selecting The
  Escape Option During Setup.

Type (SPACE BAR) For More Help, Anything Else To Continue.
F1:Wlp F2:Msg F3:Buf F4:Save F5:Send F6:Cfgp 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 Permanent

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: XON/XOFF
Modem Port Set To: COM1

F1:Wlp F2:Msg F3:Buf F4:Save F5:Send F6:Cfgp F7:Clr F8:Print F9:Exit COM
```



Heath  
Users'  
Group

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

**BULK RATE**  
**U.S. Postage**  
**PAID**  
**Heath Users' Group**

POSTMASTER: If undeliverable,  
please do not return.

**\$2.50**  
P/N 885-2132