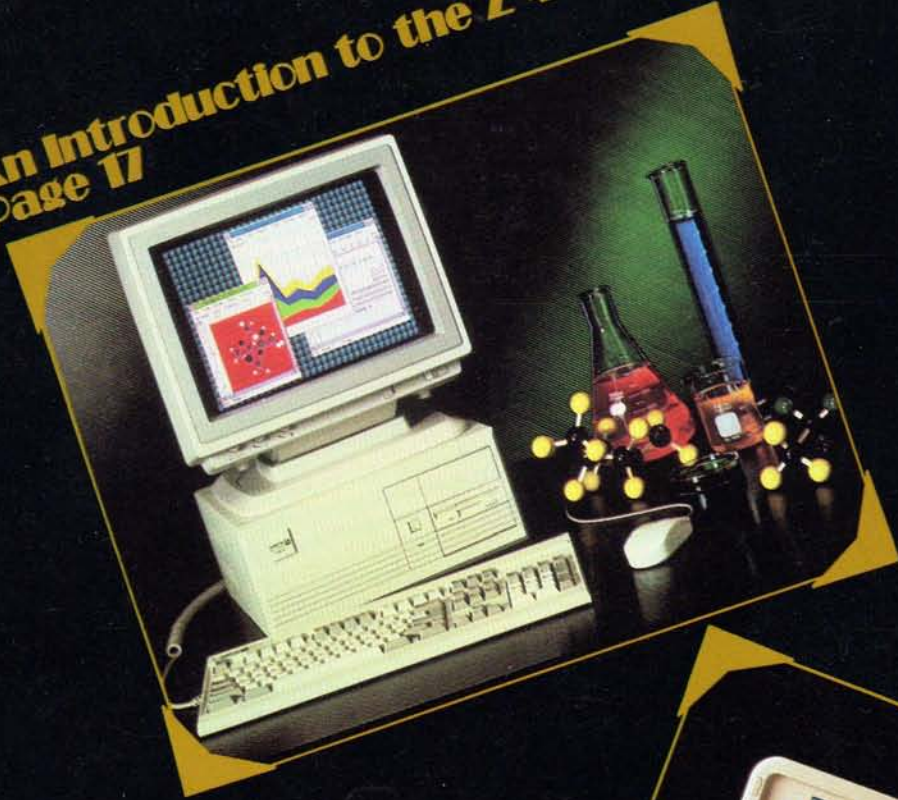# REMark

July 1991

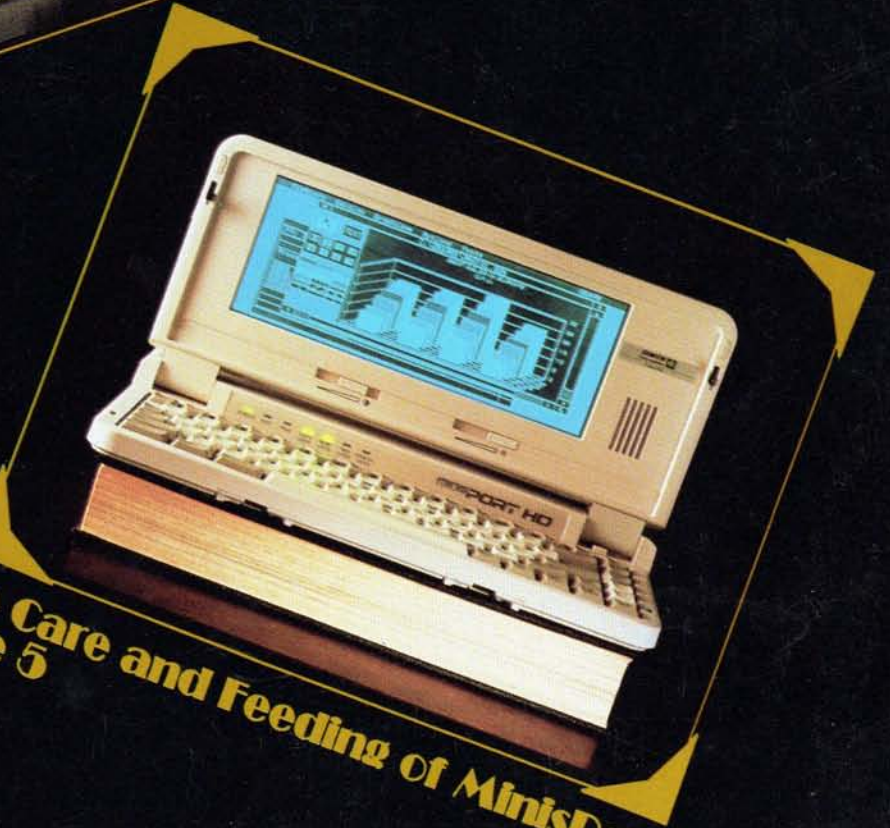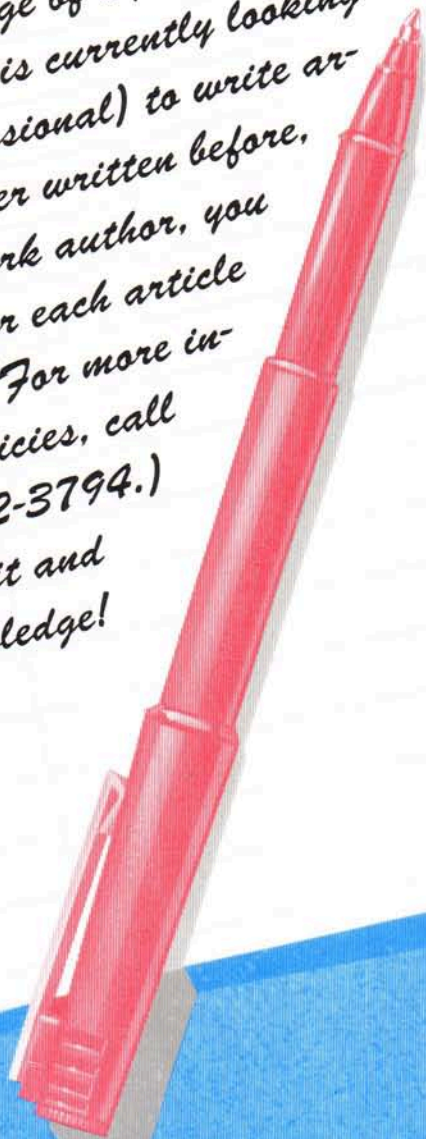## The Official Zenith Data Systems Magazine

An Introduction to the Z-486/25E
Page 17

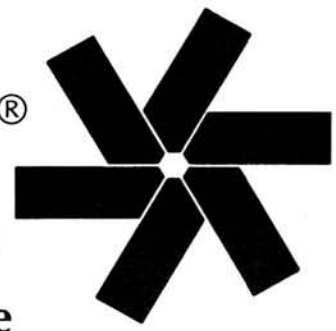The Care and Feeding of MinisPorts
Page 5

Share the Knowledge!

Have you done something interesting with your computer lately? Found a piece of software or hardware you don't know how you got along without? Designed a new product, be it software or hardware, for your system? By submitting this information in the form of a major article, you can share with others the knowledge of a particular subject. REMark magazine is currently looking for authors (novice or professional) to write articles. Even if you have never written before, give it a try! As a REMark author, you will recieve up to $400 for each article accepted and published. (For more information on current policies, call Lori Lerch at 616-982-3794.)

So... Let's get to it and Share the Knowledge!

REMark®

July 1991

The Official Zenith Data Systems Users Magazine

Advertising

Resources

Software

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
	H8 - H/Z-89/90			
ACTION GAMES	885-1220-[37]	CPM	GAME	20.00
ADVENTURE	885-1010	HDOS	GAME	10.00
ASCIRITY	885-1238-[37]	CPM	AMATEUR RADIO	20.00
AUTOFILE (Z80 ONLY)	885-1110	HDOS	DBMS	30.00
BHBASIC SUPPORT PKG	885-1119-[37]	HDOS	UTILITY	20.00
CASTLE	885-8032-[37]	HDOS	ENTERTAINMENT	20.00
CHEAPCALC	885-1131-[37]	HDOS	SPREADSHEET	20.00
CHECKOFF	885-8010	HDOS	CHKBK SOFTWARE	25.00
DEVICE DRIVERS	885-1105	HDOS	UTILITY	20.00
DISK UTILITIES	885-1213-[37]	CPM	UTILITY	20.00
DUNGEONS & DRAGONS	885-1093-[37]	HDOS	GAME	20.00
FLOATING POINT PKG	885-1063	HDOS	UTILITY	18.00
GALACTIC WARRIORS	885-8009-[37]	HDOS	GAME	20.00
GALACTIC WARRIORS	885-8009-[37]	CPM	GAME	20.00
GAMES 1	885-1029-[37]	HDOS	GAMES	18.00
HARD SECT SUPPORT PKG	885-1121	HDOS	UTILITY	30.00
HDOS PROG. HELPER	885-8017	HDOS	UTILITY	16.00
HOME FINANCE	885-1070	HDOS	BUSINESS	18.00
HUG DISK DUP UTILITY	885-1217-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG	885-4500	VARIOUS	PROD TO 1982	9.75
HUGMAN & MOVIE ANIM	885-1124	HDOS	ENTERTAINMENT	20.00
INFO SYS AND TEL. & MAIL SYS	885-1108-[37]	HDOS	DBMS	30.00
LOGBOOK	885-1107-[37]	HDOS	AMATEUR RADIO	30.00
MAGBASE	885-1249-[37]	CPM	MAGAZINE DB	25.00
MISCELLANEOUS UTILITIES	885-1089-[37]	HDOS	UTILITY	20.00
MORSE CODE TRANSCEIVER	885-8016	HDOS	AMATEUR RADIO	20.00
MORSE CODE TRANSCEIVER	885-8031-[37]	CPM	AMATEUR RADIO	20.00
PAGE EDITOR	885-1079-[37]	HDOS	UTILITY	25.00
PROGRAMS FOR PRINTERS	885-1082	HDOS	UTILITY	20.00
REMARK VOL 1 ISSUES 1-13	885-4001	N/A	1978 TO DEC '80	20.00
RUNOFF	885-1025	HDOS	TEXT PROCR	35.00
SCICALC	885-8027	HDOS	UTILITY	20.00
SMALL BUISNESS PACKAGE	885-1071-[37]	HDOS	BUSINESS	75.00
SMALL-C COMPILER	885-1134	HDOS	LANGUAGE	30.00
SOFT SECTOR SUPPORT PKG	885-1127-[37]	HDOS	UTILITY	20.00
STUDENT'S STATISTICS PKG	885-8021	HDOS	EDUCATION	20.00
SUBMIT (Z80 ONLY)	885-8006	HDOS	UTILITY	20.00
TERM & HTOC	885-1207-[37]	CPM	COMMUN & UTIL	20.00
TINY BASIC COMPILER	885-1132-[37]	HDOS	LANGUAGE	25.00
TINY PASCAL	885-1086-[37]	HDOS	LANGUAGE	20.00
UDUMP	885-8004	HDOS	UTILITY	35.00
UTILITIES	885-1212-[37]	CPM	UTILITY	20.00
UTILITIES BY PS	885-1126	HDOS	UTILITY	20.00
VARIETY PACKAGE	885-1135-[37]	HDOS	UTILITY & GAMES	20.00
WHEW UTILITIES	885-1120-[37]	HDOS	UTILITY	20.00
XMET ROBOT X-ASSEMBLER	885-1229-[37]	CPM	UTILITY	20.00
Z80 ASSEMBLER	885-1078-[37]	HDOS	UTILITY	25.00
Z80 DEBUGGING TOOL (ALDT)	885-1116	HDOS	UTILITY	20.00
	H8 - H/Z-89/90 - H/Z-100 (Not PC)			
ADVENTURE	885-1222-[37]	CPM	GAME	10.00
BASIC-E	885-1215-[37]	CPM	LANGUAGE	20.00
CASSINO GAMES	885-1227-[37]	CPM	GAME	20.00
CHEAPCALC	885-1233-[37]	CPM	SPREADSHEET	20.00
CHECKOFF	885-8011-[37]	CPM	CHKBK SOFTWARE	25.00
COPYDOS	885-1235-[37]	CPM	UTILITY	20.00
DISK DUMP & EDIT UTILITY	885-1225-[37]	CPM	UTILITY	30.00
DUNGEONS & DRAGONS	885-1209-[37]	CPM	GAMES	20.00
FAST ACTION GAMES	885-1228-[37]	CPM	GAME	20.00
FUN DISK I	885-1236-[37]	CPM	GAMES	20.00
FUN DISK II	885-1248-[37]	CPM	GAMES	35.00
GAMES DISK	885-1206-[37]	CPM	GAMES	20.00
GRADE	885-8036-[37]	CPM	GRADE BOOK	20.00
HRUN	885-1223-[37]	CPM	HDOS EMULATOR	40.00
HUG FILE MANAGER & UTILITIES	885-1246-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CAT UPDT #1	885-4501	VARIOUS	PROD 1983 TO 1985	9.75
KEYMAP CPM-80	885-1230-[37]	CPM	UTILITY	20.00
MBASIC PAYROLL	885-1218-[37]	CPM	BUSINESS	60.00
NAVPROGSEVEN	885-1219-[37]	CPM	FLIGHT UTILITY	20.00
SEA BATTLE	885-1211-[37]	CPM	GAME	20.00
UTILITIES BY PS	885-1226-[37]	CPM	UTILITY	20.00
UTILITIES	885-1237-[37]	CPM	UTILITY	20.00
X-REFERENCE UTIL FOR MBASIC	885-1231-[37]	CPM	UTILITY	20.00
ZTERM	885-3003-[37]	CPM	COMMUNICATIONS	20.00

Price List

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
H/Z-100 (Not PC) Only				
CARDCAT	885-3021-37	MSDOS	BUSINESS	20.00
CHEAPCALC	885-3006-37	MSDOS	UTILITY	20.00
CHECKBOOK MANAGER	885-3013-37	MSDOS	BUSINESS	20.00
CP/EMULATOR	885-3007-37	MSDOS	CPM EMULATOR	20.00
DBZ	885-8034-37	MSDOS	DBMS	25.00
DUNGN & DRAGONS (ZBASIC)	885-3009-37	MSDOS	GAME	20.00
ETCHDUMP	885-3005-37	MSDOS	UTILITY	20.00
EZPLOT II	885-3049-37	MSDOS	PRINTER PLOT UTIL	25.00
GAMES (ZBASIC)	885-3011-37	MSDOS	GAMES	20.00
GAMES CONTEST PACKAGE	885-3017-37	MSDOS	GAMES	25.00
GAMES PACKAGE II	885-3044-37	MSDOS	GAMES	25.00
GRAPHIC GAMES (ZBASIC)	885-3004-37	MSDOS	GAMES	20.00
GRAPHICS	885-3031-37	MSDOS	UTILITY	20.00
HELPSCREEN	885-3039-37	MSDOS	UTILITY	20.00
HUG BKGRD PRINT SPOOLER	885-1247-37	CPM	UTILITY	20.00
KEYMAC	885-3046-37	MSDOS	UTILITY	20.00
KEYMAP	885-3010-37	MSDOS	UTILITY	20.00
KEYMAP CPM-85	885-1245-37	CPM	UTILITY	20.00
MATHFLASH	885-8030-37	MSDOS	EDUCATION	20.00
ORBITS	885-8041-37	MSDOS	EDUCATION	25.00
POKER PARTY	885-8042-37	MSDOS	ENTERTAINMENT	20.00
SCICALC	885-8028-37	MSDOS	UTILITY	20.00
SKYVIEWS	885-3015-37	MSDOS	ATRONOMY UTILITY	20.00
SMALL-C COMPILER	885-3026-37	MSDOS	LANGUAGE	30.00
SPELL5	885-3035-37	MSDOS	SPELLING CHECKER	20.00
SPREADSHEET CONTEST PKG	885-3018-37	MSDOS	VARIOUS SPRDST	25.00
TREE-ID	885-3036-37	MSDOS	TREE IDENTIFIER	20.00
USEFUL PROGRAMS I	885-3022-37	MSDOS	UTILITIES	30.00
UTILITIES	885-3008-37	MSDOS	UTILITY	20.00
ZPC II	885-3037-37	MSDOS	PC EMULATOR	60.00
ZPC UPGRADE DISK	885-3042-37	MSDOS	UTILITY	20.00
H/Z-100 and PC Compatibles				
ADVENTURE	885-3016	MSDOS	GAME	10.00
BACKGRD PRINT SPOOLER	885-3029	MSDOS	UTILITY	20.00
BOTH SIDES PRINTER UTILITY	885-3048	MSDOS	UTILITY	20.00
CXREF	885-3051	MSDOS	UTILITY	17.00
DEBUG SUPPORT UTILITIES	885-3038	MSDOS	UTILITY	20.00
DPATH	885-8039	MSDOS	UTILITY	20.00
HADES II	885-3040	MSDOS	UTILITY	40.00
HEPCAT	885-3045	MSDOS	UTILITY	35.00
HUG EDITOR	885-3012	MSDOS	TEXT PROCESSOR	20.00
HUG MENU SYSTEM	885-3020	MSDOS	UTILITY	20.00
HUG SOFTWARE CAT UPD #1	885-4501	MSDOS	PROD 1983 - 1985	9.75
HUGMCP	885-3033	MSDOS	COMMUNICATION	40.00
ICT 8080 - 8088 TRANSLATOR	885-3024	MSDOS	UTILITY	20.00
MAGBASE	885-3050	VARIOUS	MAG DATABASE	25.00
MATT	885-8045	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025	MSDOS	UTILITIES	20.00
PS' PC &Z100 UTILITIES	885-3052	MSDOS	UTILITIES	20.00
REMARK VOL 8 ISSUES 84-95	885-4008	N/A	1987	25.00
REMARK VOL 9 ISSUES 96-107	885-4009	N/A	1988	25.00
REMARK VOL 10 ISSUES 108-119	885-4010	N/A	1989	25.00
REMARK VOL 11 ISSUES 120-131	885-4011	N/A	1990	25.00
SCREEN DUMP	885-3043	MSDOS	UTILITY	30.00
UTILITIES II	885-3014	MSDOS	UTILITY	20.00
Z100 WORDSTAR CONNECTION	885-3047	MSDOS	UTILITY	20.00
PC Compatibles				
CARDCAT	885-6006	MSDOS	CAT SYSTEM	20.00
CHEAPCALC	885-6004	MSDOS	SPREADSHEET	20.00
CLAVIER	885-6016	MSDOS	ENTERTAINMENT	20.00
CP/EMULATOR II & ZEMULATOR	885-6002	MSDOS	CPM & Z100 EMUL	20.00
DUNGEONS & DRAGONS	885-6007	MSDOS	GAME	20.00
EZPLOT II	885-6013	MSDOS	PRINTER PLOT UTIL	25.00
GRADE	885-8037	MSDOS	GRADE BOOK	20.00
HAM HELP	885-6010	MSDOS	AMATEUR RADIO	20.00
KEYMAP	885-6001	MSDOS	UTILITY	20.00
LAPTOP UTILITIES	885-6014	MSDOS	UTILITIES	20.00
PS' PC UTILITIES	885-6011	MSDOS	UTILITIES	20.00
POWERING UP	885-4604	N/A	GUIDE TO USING PCs	12.00
SCREEN SAVER PLUS	885-6009	MSDOS	UTILITIES	20.00
SKYVIEWS	885-6005	MSDOS	ASTRONOMY UTIL	20.00
TCSPELL	885-8044	MSDOS	SPELLING CHECKER	20.00
ULTRA RTTY	885-6012	MSDOS	AMATEUR RADIO	20.00
YAUD (YET ANOTHER UTIL DSK)	885-6015	MSDOS	UTILITIES	20.00

ZENITH data systems

REMark Magazine Subscription
& ZLink/COM1 Bulletin Board Information

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

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

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

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

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

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

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

- - - - ✂ - ✂ - - -

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

REMark Subscription / Renewal Form

New Member: ☐ Yes   ☐ No   Credit Card # _____

ID Number: _____   Exp. Date _____

Address Change? ☐

		Renew	New
Name: _____	U.S. Bulk Mail	☐ 19.95	☐ 22.95
Address: _____	U.S. First Class	☐ 32.95	☐ 37.95
City, State, Zip: _____	APO/FPO Surface Overseas	☐ 32.95	☐ 37.95
Daytime Phone #: ( ) _____	Air Printed Overseas	☐ 52.95	☐ 57.95

The Care and Feeding of MinisPorts

Mary Ellen Schutz
Senior Technical Writer
Zenith Data Systems
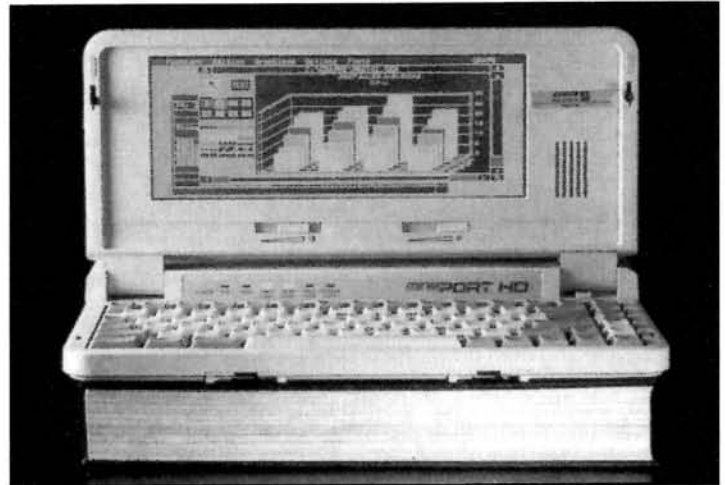
Ever had one of those days? You took your MinisPort out of the closet (it's been in mothballs all summer), plugged in the charger, and let it charge for three hours (almost) while you got ready for class. Ran out the door (leaving the charger on the kitchen table) and off to class. It's going to be high tech this term, no paper — notes, assignments, schedules the whole kit and caboodle on disk — and then disaster. Twenty minutes into the two hour lecture, the battery light comes on. Two minutes later the computer shuts down. What happened? You charged it just like the manual said. Where's that three hours they said you had?

Or how about this one? You've been making presentations to clients for weeks. Now that you've gotten used to it, this computer bit is really OK. Yes sir! You flip through that twenty minute presentation once more, but this time the guy is a hard sell. He keeps asking questions. You flip back and forth through the screens, explaining in more detail, reviewing the information. Twenty minutes is getting closer to forty, but the guy is just about sold — and then ... disaster. The battery light comes on and a second later, the computer shuts down. "What happened?", you ask as you trudge out to the car to get the AC adapter. You charge the battery every time you go back into the office, just to make sure that you've got lots of power for guys like this. Where's that three hours they said you had?

Had one like this? You saved up enough to get that new modem. Finally, you can send those articles right into the editor from wherever you are. It's such a hassle to go into SETUP, enable the thing and re-boot. It hasn't hurt anything to leave the serial and parallel ports on. What's the difference? Off to the beach you go to get that morning's work done. The sun is shining, the water is beautiful, the words are flowing (so well that you haven't even thought to save) — and then ... disaster. The computer shuts down, and an hour and a half's work is gone. Where was that annoying beep? Oh yeah, you turned it off. Who thought to look at the light, you always get at least two and a half hours of power. What happened? Where's that three hours they said you had?

What happened in all these cases is that the battery packs were acting exactly like rechargeable NiCad batteries. Read on to learn a bit more about rechargeable batteries and how to avoid the disasters.

History of Rechargeable Batteries

Believe it or not, there would be no batteries as we know them today, had it not been for an Italian frog. In 1791, Luigi Galvani, a scientist working in Bologna discovered that muscles in frogs contracted when touched by metal. Based on this animal electricity phenomenon, a second principle was discovered in 1800 by another Italian, the Conte Alessandro Volta. Electricity was generated when two dissimilar metals were placed in an electrolytic solution. The electrolytes acted chemically on one or both of the metals and generated electromotive force. Volta used this discovery to build the world's first battery, the voltaic cell. Finally, in 1831

Michael Faraday, an English chemist and physicist deduced the last piece of information required for the development of the modern battery. Faraday's law of electromagnetic induction provided a stable foundation for the study of electricity. Fast and furious development of different types of cells followed.

In 1859, the first rechargeable battery, a lead storage battery, was introduced. When the electrical generator was invented in 1866, rechargeable batteries began to play a major role in energy storage. Improvements in this original lead battery led to the sealed lead acid batteries used today to power automobiles and larger battery backed-up computers. In 1899, the first NiCad battery, a rechargeable alkaline storage battery using nickel for the positive electrode and cadmium for the negative electrode was invented. Thomas Edison, in 1900, produced an alternative design that used nickel and iron. Both batteries were shelved until 1932, since the alkaline battery materials were so much more expensive than dry cell or lead batteries, without any real advantage in terms of power or number of discharge cycles.

In 1932, two scientists (Shlecht and Ackermann) managed to increase the number of discharge cycles and improve the discharge characteristics of NiCad batteries dramatically by using a sintered pole plate. Conventional batteries packed the active materials into a metal tube or pocket; these new batteries placed the active materials inside a porous pole plate. These porous plates were formed by sintering (heating a metal powder until it becomes a

becomes a solid, but without actually melting the metal) nickel. These cells however, like other storage cells, still leaked electrolytes, oxygen, and hydrogen (causing the electrolytes to dry up) when charging and discharging, and their use was limited.

Finally in 1947, when a fully sealed NiCad battery was developed, the NiCad came into its own. Further improvements over the years have lead to rechargeable batteries capable of delivering reliable power at a stable output voltage to a wide range of products with minimal maintenance. Amazing, and it all began with a frog.

MinisPort NiCad Battery Packs

NiCad batteries were chosen to power the MinisPort (and other portable) computer series because they deliver a reliable, constant voltage level over the majority of a discharge cycle and then drop off quickly at the end. Other batteries (such as carbon, lithium, and alkaline) deliver gradually decreasing voltages during their discharge cycle, resulting in less reliable computing time for the same battery capacity.

Two different battery packs are used in the MinisPort computers. The original packs (Model number ZA-1-2), used in ZL-1 and ZL-2 computers, were rated for 10.8 watt-hours. A higher capacity battery pack (Model number ZA-1-22) was introduced with the hard disk drive versions, ZL-1-H and ZL-1-HA. These new packs are rated at 13.2 watt-hours, to handle the more demanding internal hard disk drive. Both packs deliver 6 volts and can be expected to handle approximately 1000 charge/discharge cycles. Both contain built-in overcharge and short circuit protections.

The rechargeable battery packs contain sealed nickel-cadmium cells with sintered poles. They use nickel hydroxide for the positive pole, a cadmium compound for the negative pole, and potassium hydroxide (an alkaline) for the electrolyte. As they charge and discharge, the positive plate changes between nickel oxyhydroxide and nickel hydroxide and the negative plate changes between cadmium metal and cadmium hydroxide. These chemical changes produce the electricity used each time you power your computer with the battery.

The chemical changes inside the cell also make the battery subject to a phenomenon known as memory effect or voltage depression. Repeated short discharge cycles followed by long charge cycles seem to reduce the amount of computing time available to you. What appears to happen is that the battery "remembers" how much energy you used in the last few discharge cycles, and only charges enough to allow you that much energy for the next cycle. Your battery discharge time gets shorter and shorter

with each cycle.

Some researchers believe that this phenomenon is caused by crystalline growth inside the cell. Each time a battery is not completely discharged before charging (and the MinisPort battery packs are charged every time you plug into AC) some crystal growth occurs. Overcharging can also cause this crystal growth. As the crystals grow bigger, they reduce the effective area of the poles and lower the voltage level available. Although the cell will still deliver the same amount of energy, more of the cycle is at a lower voltage level. This fact is used by some battery manufacturers to support claims that their batteries are not subject to memory effect. Since the same amount of total energy is available, their claims are somewhat valid, if not practical. The problem for the end user is that the energy available at the lower voltage level may not be enough to run the computer. The problem can be avoided by using full discharge cycles followed by full charge cycles or by exercising your battery at regular intervals. Abused batteries that do exhibit signs of memory effect can be restored to near full capacity. (There's hope for the salesman with the 40 minute battery.)

Another characteristic of NiCad batteries related to the chemical changes that occur during discharge is known as deep discharge. When a discharged battery pack is not recharged, but continues to have power drawn from it, the poles can actually reverse polarity, destroying the battery. MinisPort computers draw power first from AC, next from the battery pack, and then from the lithium backup batteries. When no lithium battery power is available, the computer continues to attempt to draw power from the NiCads. Battery packs left in computers during extended storage and those in computers left open after an automatic shutdown experience shortened discharge times or, in extreme cases, are destroyed. Abused batteries that exhibit early signs of deep discharge can also be restored to near full capacity with seasoning. (There's hope for the student with the 22 minute battery.)

Testing Battery Life

"So, wonderful", you say. "But where's my three hours?" Well, here it comes. Properly seasoned, exercised, and fully-charged MinisPort battery packs deliver approximately three hours of computing time while running a battery life test program set for 50% CPU usage, 20% disk drive accesses, and 30% video accesses, providing your computer is set up as follows:
- The LCD display enabled and no external video monitor connected.
- The serial, parallel, and modem ports disabled.
- The hard disk drive power down time

set to 60 seconds.
- The RAMDISK off.
- The CONTRAST set so the display is readable with the BRIGHTNESS set to maximum.
- The external floppy disk drive disconnected.
- The operating speed set to FAST.
- The BACKLIGHT TIMEOUT set to ALWAYS ON.
- The keyboard locks set to NUM LOCK on, PAD LOCK off, CAPS LOCK off, and SCROLL LOCK off.
- No modem installed.
- One megabyte of memory installed.

Changing any of these parameters will affect the amount of computing time. Using the computer in real life is likely to change almost all of them. I generally get between two and two and a half hours of computing time while generating manuscripts; your particular application will determine your computing time. To get a good feel for the amount of computing time you can expect, using a seasoned and fully charged battery pack, run your applications on the computer until the power LED turns red and the low battery warning beep sounds. Check the time, save your work, turn off the computer and charge the battery pack for three hours. Repeat this process three or four times and you'll have a good idea of what you can expect from your MinisPort.

The way you use your battery pack also affects the amount of computing time you can expect. In the best of all worlds, you would have complete discharge cycles, followed by full charge cycles. Heaven knows that's not always practical or even possible. Even your brand new battery pack sat on someone's shelf long enough to reduce its discharge cycle. You used your computer for an hour and a half today and need to have it going for a two hour meeting tomorrow. Your three year old opened the computer, pushed the button, and went off to play. You've been using it at home on AC for months, and now you have to finish that project by Monday, with the family reunion at the campground scheduled for this weekend. With all this memory effect, deep discharge, reverse polarity, and the like, how can anyone keep a battery pack healthy? Just use some simple procedures, known as seasoning, exercising, and restoring.

Seasoning Battery Packs

All new battery packs and most battery packs that have been stored for an extended period of time require seasoning. This process takes the battery through a couple of full charge and discharge cycles, conditioning the battery for optimum computing time. To season a battery pack:
1. Remove the battery pack from the computer.
2. Attach the AC adapter and allow the battery to charge for at least 10 hours.

3. Disconnect the AC adapter.
4. Install the battery pack in the computer.
5. Open the computer and turn it on.
6. Allow the computer to run until the power LED turns red and the low battery warning beep sounds, and the computer shuts down. You can use the computer while the battery discharges, but remember that you may only have 20 minutes of computing time before the computer shuts down.
7. Close the computer and attach the AC adapter.
8. Allow the battery to charge for at least 3 hours.
9. Disconnect the AC adapter, run the computer until it shuts down, close it, and then recharge the pack for at least 3 hours.

Exercising Battery Packs

Exercising the battery pack at regular intervals minimizes the impact of memory effect. Exercising is the process of periodically forcing your battery through a complete discharge and full charge cycle. To exercise your battery pack:
1. With the AC adapter disconnected, open your computer and turn it on.
2. Run the computer until the power LED turns red, the low battery warning beeper sounds, and the computer shuts down. Again, you may use your computer while you discharge the battery. Just be sure you save any work in progress when the low battery warnings come on.
3. Close the computer and attach the AC adapter.
4. Charge the computer for at least 3 hours.

The way you use your computer dictates the exercise interval. If you operate primarily on AC power, exercise the battery pack at least once every 6 months. If you use the computer daily on battery power, exercise it every two weeks. If you use the computer on battery power every few days, exercise it every 3 to 4 weeks.

Restoring Battery Packs

Badly abused battery packs, with as little as 10 minutes of computing time left in each discharge cycle can be restored to near full capacity. Restoring is the process of intentionally forcing the battery pack into the early stages of deep discharge and then recharging it. Note that seasoning and regular exercising are much healthier for your battery pack, and, if the pack has been too badly abused, the restoring process does not always work. To restore a battery pack:
1. Disable the lithium backup batteries with the insulating strip or remove them from the computer.
2. Open the computer and turn it on.
3. Run the computer until it shuts down. Leave the computer on with the lid open for at least 10 hours after it shuts down.
4. Close the computer and remove the battery pack.
5. Attach the AC adapter to the battery pack and charge it for at least 10 hours.
6. Disconnect the AC adapter and install the battery pack in the computer.
7. Remove the insulating strip or re-install the lithium backup batteries.
8. Open the computer and turn it on.
9. Leave the computer on until the power LED turns red and the low power warning beep sounds.
10. Turn off the computer and close it.
11. Reconnect the AC adapter.
12. With the computer turned off, recharge the battery pack for at least 3 hours.

If short discharge time with each cycle persists, repeat the restoring procedure. If the problem continues, you'll have to replace the pack.

Power-Saving Options

If you have already been using (and not abusing) your batteries, but are still not getting the kind of battery-powered computing time you'd like, try using some of these power-saving options. If you still cannot live with the discharge time, you may want to consider upgrading your computer. Remember that the MinisPorts are pioneers. A newer computer with power conservation improvements like suspend mode may be what you need.

In addition to the power-saving options listed below, one other option is available for those of you with older Minis-Ports. Purchase a higher capacity battery pack. The ZL-1-22 packs are the same size and install the same way as the ZL-1-2 packs. Using the higher capacity pack along with power saving measures may get you the computing time you need.

SETUP — The Setup program (found in the Monitor program) has two sets of parameters: one for AC power operation and one for battery operation. Use the power saving options in the battery operation SETUP. The computer will automatically invoke your power savings when operating from the battery pack and ignore them when operating from AC.

Modem — If you have an optional modem installed in your computer, remember that when enabled it draws a lot of power, even when you're not using it. Connect to AC power before using the modem whenever possible. Use the Setup program to disable the modem whenever you are not actually using it. It takes a few keystrokes to enable it when you need to use it, but the power savings are worth it. (There is hope for the author with the one and a half hour battery.)

Disk Drives — Disk drives are another major power draw. Connect to AC power before using an external floppy disk drive whenever possible. For MinisPorts with internal hard disk drives, use the Setup program to adjust the hard disk drive power down (WINCH POWER DOWN) time. A short timeout (60 to 120 seconds) allows the drive motor to be turned off when it is not actually being used. You can also save power by running programs from the silicon disks instead of the disk drive.

Other Peripherals — The serial and parallel ports also draw power when they are not being used, although not as much as the modem. Use the Setup program to disable the ports when you are not using them. Connect to AC before using a printer, scanner, mouse, or other peripheral whenever possible.

Operating Speed — When you invoke SLOW mode through the Setup program, the actual operating speed of the computer is reduced to 4.77 MHz and power is conserved. When you invoke SLOW mode through the keyboard (using the FN-F3 key combination) you merely add wait-states, causing slower operation without power savings.

Video — Use the LCD when operating on battery power; a CRT draws more power. If there is plenty of ambient light, turn the backlight off. You can also reduce the brightness and contrast, keeping the screen readable without drawing so much power. If there isn't much light to work by, set a short timeout for the backlight (1 to 5 minutes). This turns off the backlight when you aren't actually using the computer, saving power when your attention is elsewhere for a few minutes.

Extended Storage — The computer draws power from the batteries to maintain the date, time, Setup, and RAMDISK information even when it's turned off. If you are not going to use the computer for a while (a few days to a few months or more), remove the batteries from the computer.

Power Down — This is a novel concept, but turn the computer off when you are not actually using it. How many times have you stopped work for just a minute, and not gotten back to it for hours? With two small children running around, the habit of saving my work and turning the computer off when I stop working has saved me from a lot of nightmares.

The Lithium Backup Batteries

There are two other batteries in the MinisPort computer. These lithium backup batteries provide power to the real-time clock and RAMDISK when neither NiCad nor AC power is available. These can be real lifesavers when you absolutely have to have information stored on the RAMDISK, or real pains when they're dead again and you can't get the computer booted without entering Setup. Keeping a charge on your battery pack prevents these batteries from being drained. ✳

When HEPCAT™ sings...

...the other cats get to sing along!
And now, HEPCAT does Windows!

HEPCAT, the powerful, versatile pop-up calculator from ZUG, is now even better. HEPCAT not only can pop up over just about any DOS program you are likely to use, but now version 2 can also pop up over Windows™ 3, even when it is running in the standard (286) or 386 enhanced modes.

What Is HEPCAT?

HEPCAT (Handy Engineer's and Programmer's CAlculation Tool) is a floating point calculator with several scientific/engineering features built in, and a binary (programmer's) calculator combined into one tiny, powerful program. HEPCAT is a memory resident program that "pops up" on your screen whenever you activate it by typing a special "hot key" sequence.

The Other Cats Can Sing Along

Unlike other pop-up calculators, HEPCAT is concurrent. That means that when you pop it up over a running program, the program can continue to run. For example, if you pop it up while Lotus™ is busy loading a huge spread sheet, it will continue loading while you perform your calculations. And HEPCAT always pops up in the current video mode, rather than forcing the screen into a text mode like other pop-ups do. HEPCAT can pop up in any standard CGA, EGA, VGA, or Hercules™ graphic mode, as well as in any text mode. It can even pop up in some non-standard graphic modes (but it may not clear its window when you exit).

HEPCAT Works Harder

The floating point calculator in HEPCAT 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, and it does rectangular-to-polar and polar-to-rectangular coordinate conversion. It also includes several built-in US-metric and metric-US conversions. 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. HEPCAT is easier to learn, too, with commands that make sense.

If you are tired of pop-up calculators that can only sing solo, or calculators that can do DOS but not Windows (or Windows but not DOS) give HEPCAT a try. HEPCAT is available from ZUG as part no. 885-3045 for $35.00 (plus S/H). It works on any Zenith Data Systems computer that runs MS-DOS or Z-DOS (including the Z-100 series), and on most PC-compatibles. If you have HEPCAT version 1, send in your original distribution disk and $10 to upgrade to version 2.

ENABLE Revisited
Part 3

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

SpreadSheet

In the last article of this series, I discussed the changes in ENABLE's word processing module. This article will focus on the spreadsheet, the changes and improvements made in version 3.0.

ENABLE is an integrated software package. It combines word processing, spreadsheet, database, and telecommunications. Each of these modules can pass information to the other with a keystroke or two. The most current version of ENABLE, OA or 3.0, has added several new features to the spreadsheet. These features make the spreadsheet a very important and powerful part of the total ENABLE package.

Being an integrated package, ENABLE uses many of the same keystrokes between applications. This saves you time in learning the various applications, as the keystrokes are similar. As an example, in word processing, pressing the F3 key will create a new line. Pressing the F3 key in the spreadsheet will create a new row or line.

Like all of ENABLE, pressing the F10 key will display the Top Line menu. This will permit you to select options from this menu. The Top Line Menu normally will expand as you make a selection, providing even more options. As you become familiar with ENABLE, you can use the Expert Commands to perform these same functions with a minimum of keystrokes.

To use the spreadsheet option in ENABLE, all you have to do is highlight the "Use system", "Spreadsheet/Graphics", and then press the <Return> key to create a new spreadsheet. ENABLE will then display a long highlighted block in which you must type in the name of the file to create. Note, this block is long and will support several levels of DOS directories if desired. You may insert up to 64 characters in this block identifying the subdirectory and file name.

You may also type the first letter of each of these options and ENABLE will perform the same steps. Therefore, pressing the "U","S","C" keys will get you to the same point in the program. This function will work throughout ENABLE. You may press the first letter or the number of the option and ENABLE will perform the same steps as if you highlighted the option using the cursor keys and pressed <Return>.

After inserting the name of the spreadsheet file and pressing <Return>, ENABLE's spreadsheet screen will be displayed. The
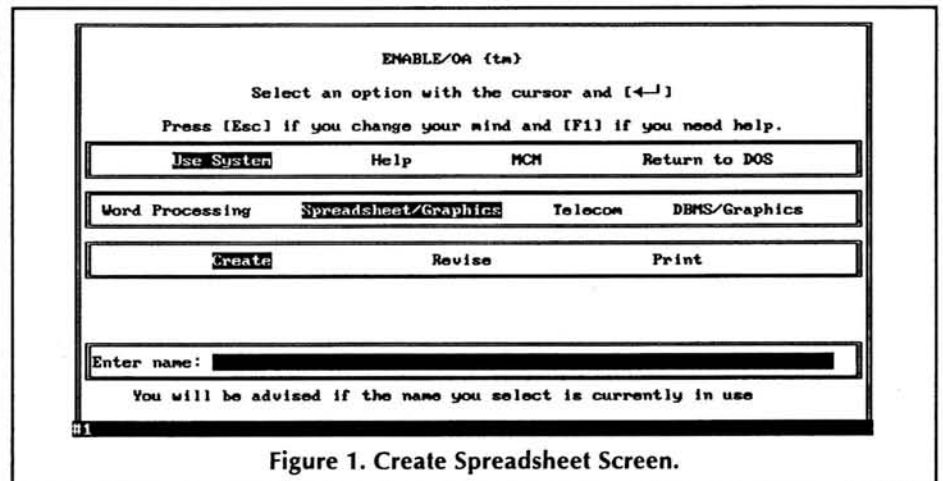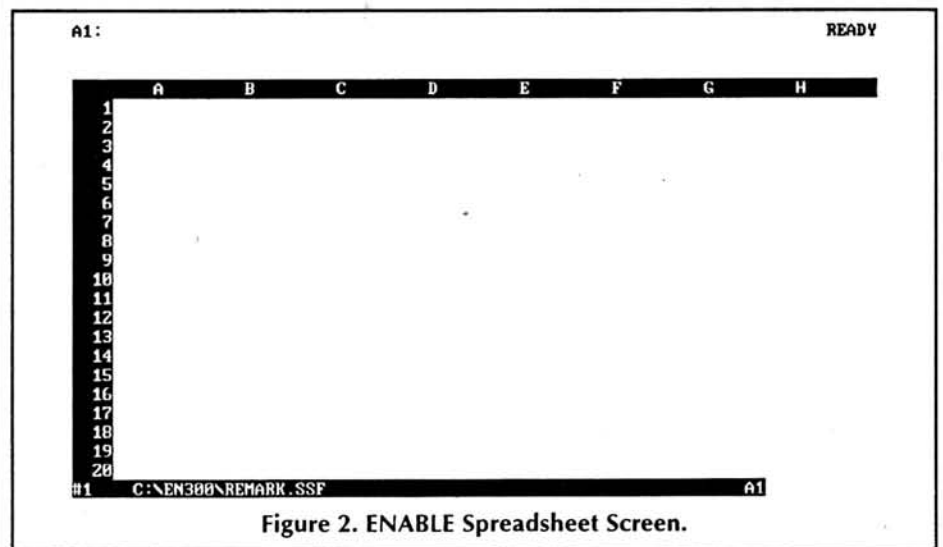

Figure 1. Create Spreadsheet Screen.

Figure 2. ENABLE Spreadsheet Screen.

spreadsheet looks like any of the other spreadsheets available on the market. You can move about the spreadsheet in the usual manner. The arrow keys will move you one cell in the direction of the arrow. PgUp or PgDn will move you one screen in the selected direction. Pressing the Tab key will move you one screen to the right, while Shift/Tab will move you one screen to the left.

As I said in an earlier article, ENABLE OA supports LIM 3.2 or 4.0. Because of this capability, the size of the spreadsheet has been increased. In earlier versions of ENABLE, the spreadsheet was limited to about 64,000 cells. You could control the shape of the spreadsheet, but you could not increase the size. ENABLE OA now supports over 10 million cells. Spreadsheets can now be up to 1024 columns by 9999 rows. With a basic 640K computer, you will still be limited in the size of the spreadsheet you can create. If you make a large spreadsheet, you will need the extra memory provided by EMS. As you work on your new spreadsheet, you can check the available memory by using the Top Line

By using these procedures, you can quickly check on your system and see how much memory you have remaining. It is better to know when you are about to run out of memory, than have it happen near the end and possibly lose some data.

The status window also shows the status of your overall system including the COM and LPT ports.

The new versions of LOTUS 1-2-3 now support 3-D spreadsheets. ENABLE OA, which came on the market about six months before LOTUS, also has 3-D spreadsheet capability. The ENABLE 3-D spreadsheet permits you to create up to 32 levels in a spreadsheet. Using this capability, you could create a spreadsheet which functions as a master. This could then be backed up with 12 individual spreadsheets which contain monthly data. As you enter the monthly data, it could, if desired, be car-

Figure 5. ENABLE Status Window.
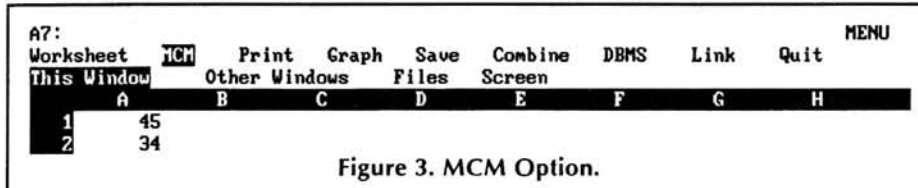
Figure 3. MCM Option.

Menu and MCM option. To access the Top Line menu you can use the F10 key or the slash (/) command. Either one of these options will cause ENABLE to display the Top Line Menu.

On this menu, highlight the "MCM" option and press <Return> or press the letter "M" for the MCM option. This will display

Figure 4. Other *Windows* Options.

additional options available under MCM. In this menu, select the "Other Windows" option. Again, you may highlight the option with the cursor keys or simply press the letter "D".

ENABLE will now open another window which will display the overall status of the system including memory available. Note in Figure 5, on the bottom of the display is the Environment section. The system I used to create this document has 1 Meg of memory. The basic 640K, plus an additional 256K for EMS. I had two windows open, the word processing document REMARK3.WPF and the spreadsheet REMARK.SSF. These two windows, plus status window used up all of the EMS memory and I still had 310K of base memory available.

Figure 6. This Window (3-D) Selection.

ried to the master sheet.

To move between the levels of the 3-D spreadsheet, just press Ctrl/PgDn to move down one level or Ctrl/PgUp to move up one level. The level you are on is displayed in the upper left corner of the screen. The level is identified by letter(s). "A" is the first level and "AF" is level 32. Once you move off the first level, the cell is identified by level and then actual cell address.

ENABLE's 3-D spreadsheet permits you
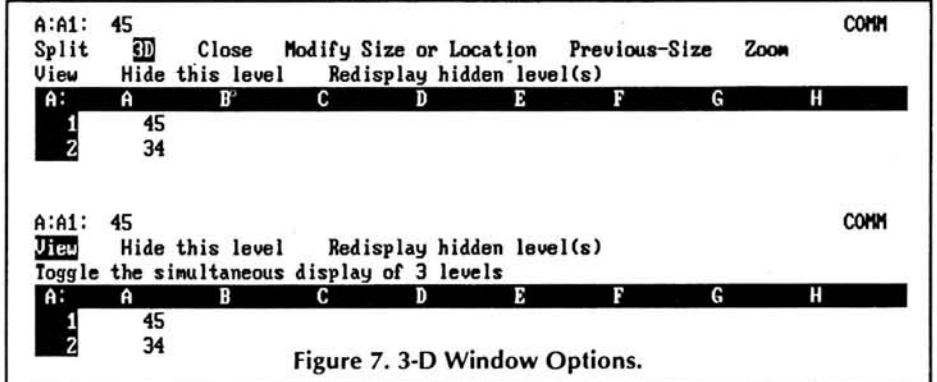
Figure 7. 3-D Window Options.

to display part of three spreadsheets on the screen at one time. You can select any three levels to be displayed. If you want to display levels that are not adjacent, you must go to the level(s) between the desired levels and turn them off using the option, "Hide this level," discussed below.

To display the three levels, you must first select another level by pressing Ctrl/PgUp or Ctrl/PgDn. If this is not accomplished first, the steps shown below will not work. You must first display the Top Line Menu by pressing the F10 key or slash. On the Top Line menu you must select "MCM." In the MCM option, under "This Window" option note that a "3D" option is now available. Select "This Window."

Once you select "3D", another set of options will be displayed. The options on this menu permit you to VIEW three levels, Hide the current level, or Redisplay the hidden level(s). At this option, select "View." Note, this is a toggle key, so by reselecting this option you will be returned to one level display.

As soon as you select this option, the ENABLE spreadsheet will be redisplayed. Now, instead of only one level being displayed, parts of three will be shown. You can move between the levels by pressing Ctrl/PgUp or Ctrl/PgDn. This will move the levels as a group of three. To move between the three levels shown on the screen, press the F6 key. This will move you one level at a time. This will not move the basic three levels shown on the screen.

Pressing PgUp, PgDn, Tab, Shift/Tab, or any of the cursor keys will move all three levels in the same direction. They are linked together for these operations.

To address cells on another level, you

Figure 9. Linked Spreadsheet Screens.

value from the selected cell will be displayed. You may use any of the functions that you would use on a single level spreadsheet on a multiple level spreadsheet, as long as you identify the level the information is located on. As an example, to display the sums of cell "B3" from levels "B" to "M", you would type in "@sum(b:b3..m:b3)" in cell B3.

Information on the ENABLE 3-D spreadsheet is provided in the Supplemental Documentation manual included with the program.

Another new capability that exists with ENABLE's spreadsheet is the Link function. Using the Link function, you can get information from another spreadsheet. While working on a business plan before ENABLE OA, I had to either write down the data on one spreadsheet for use on another or use the copy command between windows for data that was close together. Because of memory limitations, I was unable to directly copy the data between windows so
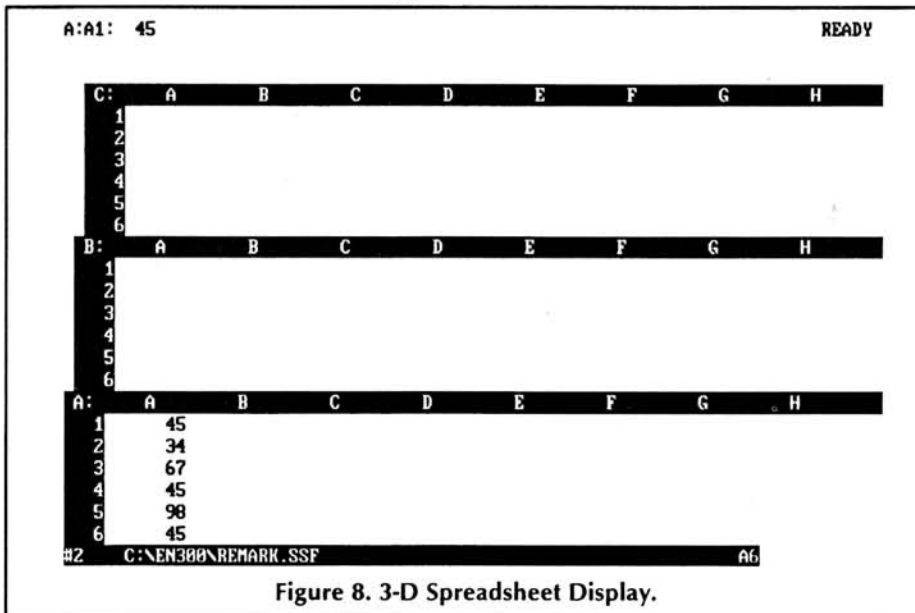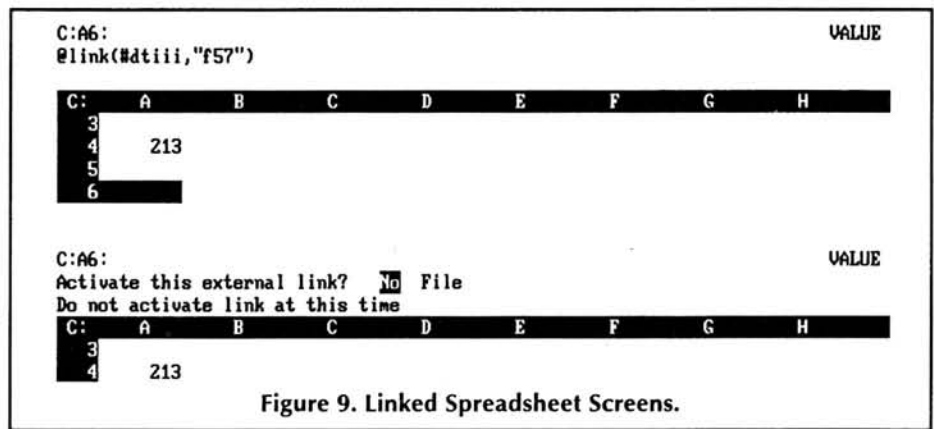

Figure 8. 3-D Spreadsheet Display.

must identify the level first and then the cell. To show the value of cell B3 on level C in cell B3 on level A, you must move to level A and cell B3. In this cell, type "+C:B3". When you press the <Return> key, the

I would have to copy the numbers to a short intermediate spreadsheet and then perform another copy into the larger destination spreadsheet.

Using the Link capability provided in

ENABLE OA, I can now identify the cells and the spreadsheets I want to copy data from. When I open the spreadsheet with the Links, ENABLE asks if you wish to update the Links. If you response Yes, ENABLE will immediately go to the other spreadsheets and get the information that resides in the indicated cells. This transfer occurs before any recalculations. If you have an ENABLE 2.0 spreadsheet, you must resave it in OA format before this function will work.

To establish this link, you must know the name(s) of the other spreadsheet(s) and the cell(s) you wish to copy. Move the cursor to the cell where the linked data is to be displayed. Type in, as an example "@LINK(#DTIII,"F57")" and press <Return>. ENABLE will then ask if you wish to establish the link now. If your response is yes, the data from the spreadsheet "DTIII" cell "F57" will be shown.

Link can be used with most functions available in the spreadsheet. If, for example, you wish to import a range of values from another spreadsheet into one cell in the current sheet, type @LINK(#DTIII,"F10..F30"). In a later article I will expand on the use of the LINK function in the new version of ENABLE.

ENABLE OA has added 17 new @Functions to the spreadsheet. Many of these new functions are related to the 3-D capability of the spreadsheet. Some of these @Functions provide new capability in the area of date arithmetic. One such function, @ADDM, permits you to add months to a date in the cell.

One feature of ENABLE that I use quite a bit is its MACRO capability. As an example, I have created a MACRO which inserts the highlighting around the spreadsheet figures in this article. I used the "learn" capability of the ENABLE macros. The macros are available throughout the program. This being the case, then macros are available in the spreadsheet.

Macros in the spreadsheet can be accomplished in two ways. The first method is using the macro procedures that are common throughout ENABLE. If you are familiar with the construction of macros in LOTUS 1-2-3, then you can use the same

procedures in ENABLE as a second method.

If you construct extensive macros in the spreadsheet using the procedures that are similar to LOTUS, you could be using extensive space in the file. A new feature in ENABLE OA is the @MACRO function. This function permits you to build a macro procedure inside a spreadsheet, but save it as a library of macro commands.

To use this capability, you must create a spreadsheet which contains the macro procedures you wish to use. Because this is a separate file, it can be called from any spreadsheet. Once you have created the macros, you save the spreadsheet with an extension of ".LIB". You may now create the operating spreadsheet. Anytime you wish to invoke one of the macros in the library, insert this command in the desired cell.

```
@MACRO(#FILENAME,"DISPLAY CELL,EXECUTE CELL")
```

If you invoke the macro in the operation spreadsheet, ENABLE will call up the LIB file. It will display whatever is inserted in the first named cell (DISPLAY CELL), and process the macro starting in the second named cell (EXECUTE CELL). This would cause the spreadsheet to display a message, like "Processing," while it actual performed the macro.

How many times have you deleted something from a spreadsheet and then decided you really didn't what it removed. If you delete a single cell by spacing through it, ENABLE will permit you to recover it. To recover the last deleted cell press F9 U.

```
C1:   (F,2)  @RAND*100                                MENU
Row-Borders    Col-Borders    ID-Borders
Select printing of cell identifier borders
      A        B        C       D      E      F      G      H
  1   45.00    67.00    56.30
  2   34.00    789.00   78.82
  3   67.00    456.00   17.40
  4   45.00    76.00    73.04
  5   98.00    34.00    34.77
  6   45.00    98.00    63.39
  7            56.00    23.39
  8
```
Figure 10. Basic Spreadsheet.

This will recover the lost information.

Many times when I print a spreadsheet I would like to have the borders printed along side the data. This would assist me in using the data and for developing other applications. It would also be of help to use the @LINK function in ENABLE. ENABLE OA has a feature that permits you to select borders with the output. If this option is

	A	B	C
1	45.00	67.00	56.30
2	34.00	789.00	78.02
3	67.00	456.00	17.40
4	45.00	76.00	73.04
5	98.00	34.00	34.77
6	45.00	98.00	63.39
7		56.00	23.39

Figure 11. Spreadsheet Output Using Border Option.

selected, both the data and the border ID will be printed. You could print the borders in older versions, but the output was in a long format and the formulas were printed. Figure 10 shows a basic spreadsheet as you would see it on the screen. Figure 11 shows the same spreadsheet after it has been sent to a printer using the border print function.

```
A1:   (F,2)  45                                        COMM
List  Highlight  Map
Status    Formulas     Range-Names
      A        B        C       D      E      F      G      H
  1   45.88    67.88    72.84
  2   34.88    789.88   31.22
```
Figure 12. Audit Options.

Note that only the borders and data are shown. No formulas are displayed. If you have dashes or other characters in a cell, these would also be shown. Also note that

```
A1:   (F,2)  45                                                   COMM
Recalc   Protection    Format    Alignment    Width  Col-Width  High-id  EMS-Pages
AUTO     Profile: U    (G,2)     V: R  L: C    9                 C11      2
      A        B        C       D      E      F      G      H
  1   45.88    67.88    72.84
  2   34.88    789.88   31.22
```
Figure 13. Audit Global Status Listing.

the output stops at the end of the data in row seven. Nothing below this is shown.

Another function in the printing area that has been added is the capability to print non-contiguous ranges. I have had requirements in the past to print a few cells here and there in a spreadsheet. The only way to accomplish this task before OA was to copy the values to another location in the spreadsheet and then print the area. With ENABLE OA it is possible to establish print ranges. You would move to an open area in your spreadsheet and create range names along with the range of cells you wish to print. You would then print the range names, which include the non-contiguous cells.

To enhance your printed output, you can use word processing attributes. You can always use the attributes in a cell by pressing the Alt and the desired attribute key. The attribute selected would only work for one cell at a time. If you have colors selected for the attributes, they would be displayed on the screen.

Another method of enhancing your printed spreadsheet is to use the %CONTROL codes for your printer. I have inserted the %CONTROL codes for landscape and other printer functions in a spreadsheet. You would insert the %CONTROL command above the line where you wanted the attribute to take effect. The printer would then respond to the control code and print as requested until another code is encountered.

A new capability added to ENABLE OA is the print attribute file. Using this file, you would create a spreadsheet file like the macro library discussed above. This file would contain the %CONTROL codes for the attributes you wish to use. This file could contain all of the printer codes as it can be used with all spreadsheets. This file contains two lines per attribute instruction. The first line is the %CONTROL for the printer. The second line would contain the actual name of the attribute such as, DOUBLE-HIGH, four spaces and an abbreviation for the attribute. You then make a range containing only the %CONTROL line naming it with the abbreviation shown on the second line. When you complete this file save it with a name, such as "ATTRIB.SSF." In the spreadsheet file in which you want to use the attributes, insert the command @PRTCTL(#ATTRIB.SSF, "attribute range name") where you want to start. You MUST also create a RESET line. You would use this range name to turn off the command.

You can also use the special character function I talked about in the word processing article in the spreadsheet. This per-

navigation
Continued on Page 28

```
A1:   (F,2)  45                                        COMM
List  Highlight  Map
Set or Clear Map mode
      A        B        C       D      E      F      G      H
  1   45.88    67.88    72.84
  2   34.88    789.88   31.22
```
Figure 14. Map Option Selection.

12   July 1991

One From the Pack:

Richard J. O'Connor
848 Fenske Drive NE
Olympia, WA 98506

A Review of
Central Point Backup

Introduction

Only the computer industry could come up with a term like "unbundling." Of course, the jargon shouldn't surprise me; after all, this is the industry that brought us "outsourcing," "aftermarketing," and the unforgettable "downsizing." I can understand "bundling," in which a company packages several components together to offer increased value for "less than the price you'd pay if you'd purchased these separately!" But "unbundling??"

"Here, honey, would you unbundle little Erin while I take off my parka?"

Despite the mind warp such terminology causes, bundling/unbundling are here to stay. Sometimes it's a little difficult to tell which of these two approaches is in vogue, since different players in the computer marketplace hold differing opinions. Years ago, when most computer manufacturers sold you bare hardware, Kaypro bundled a tidy software package with their CP/M and MS-DOS machines, and took an early lead in the market as a result. Other manufacturers have experimented with this to some degree since; but now most dealers offer no more than DOS (and perhaps Windows) with their machines.

In the software world, Lotus Development started the trend toward "integrated packages" with Lotus 1-2-3, which packaged a spreadsheet, a graphics program, and (arguably) a data base manager into a single product. Again, others experimented; but today nearly all such integrated products are aimed to meet the relatively low-powered needs of new users (for example, Microsoft Works). A noticeable exception in recent years has been Central Point Software's popular bundle "PC Tools Deluxe", which brings together a host of powerful applications and utility software programs into a single product with a consistent interface. In my use of PC Tools, I've found that while most of the programs aren't industry-leaders by themselves, the value of the package far exceeds list price. Thus, in this case, the whole is greater than the sum of the parts.

What's This, Then?

So I was quite surprised when Central Point announced in October that PC Backup, the hard disk backup program included in PC Tools had been renamed "Central Point Backup," and was now available as a separate product. Why would Central Point break up a winning team? What would happen to future releases of PC Tools? And why the Backup program, of all utilities?

On request, Central Point sent me some answers to these questions, so I decided to get a copy of Central Point Backup for review to help you decide if it's something worth adding to your collection of utilities. First, I'll cover a bit of Central Point's philosophy in making this move. Then I'll report on installation and testing of the product on my Zenith Data Systems Z-248/12 machine. Finally, I'll identify those users who would most benefit from obtaining a copy of the new and improved Central Point Backup.

Central Point Backup currently retails for $99, while PC Tools Deluxe, Version 6 retails for $149. Central Point Backup was "unbundled" from PC Tools, Version 6 as an improved version of the PC Backup program. However, Central Point Backup will continue to be offered as a part of PC Tools. Current owners of PC Tools Version 6 can download a maintenance release from the Central Point Bulletin Board (503-690-6650) which contains all of the enhancements that have been made to the backup program.

One thing Central Point Software is hoping to do via this move is to attract new users to the PC Tools family. When you purchase Central Point Backup, an upgrade form is included that allows you to buy the complete PC Tools package for $50. Now if you know you'll want the entire utility set, you could just purchase the whole thing for $149 at the outset. This approach, however, allows prospective clients to experience the PC Tools user interface by using Central Point Backup, and then add the DOS Shell, Disk Optimizer, Disk Cache, and Desktop Manager components if desired. According to Central Point President Corey Smith, "The introduction of a stand-alone backup program is in keeping with our commitment to respond to the evolving needs of our customers. Our . . . product lets customers purchase just the function they want right now while providing an easy and affordable upgrade path when they later require a full set of utilities."

Yes, but why Central Point Backup? Because this product is perhaps the one part of this package that can go toe-to-toe with the competition in a fair fight. Central Point Backup is the only "mainstream" backup program that backs up to QIC-40, QIC-80, and Irwin format tape drives, as well as floppy disks, which allows users who migrate from floppy backups to a tape system to use the same software. In fact, a new "bundle" is in the works; Central Point Backup is now being shipped with Irwin Magnetics and Wangtek tape drives. And in time trials conducted by InfoWorld in the fall of 1990, Central Point Backup was found to be competitive with or superior in backup speed to such standbys as Fast-Back Plus and Norton Backup.

Those of you who work with desktop computers in the corporate environment may know of another reason for the unbundling. According to Central Point, corporate Data Processing Departments often prefer to control which PC support functions and utilities will be supported centrally. Since file backup software is the most requested utility, offering Central Point Backup as a stand-alone program gives users in shops who haven't standardized on PC Tools a chance to use a leading product that otherwise would be inaccessible.

Installation and Testing

I installed a copy of Central Point

Backup on my Zenith Data Systems Z-248/12, which has a 40 MB hard disk, VGA video, and a ZCM-1492 Flat Tension Mask monitor. Figures 1 through 5 are screen shots that were captured by First Publisher's SNAPSHOT program, imported into First Publisher, "stretched" vertically to improve clarity, and touched up to more accurately depict the drop shadows and reverse video areas of the screen. The biggest problem with this SNAPSHOT program is that it cannot capture an entire 80 column screen; the limit is 72 columns. Therefore, those figures are really "partial" screens, and are presented to give you a feel for the appearance of the actual screens. SNAPSHOT's limitations are not an isolated case; I tested several public domain and shareware text screen-savers before returning to SNAPSHOT and First Publisher. PC Tools uses text-based screens, but makes frequent use of "graphics"-type characters at the high end of the ASCII spectrum which tend to confuse most text mode screen savers. Guess I'll just have to break down and treat myself to a "real" screen saver like Inset Systems' "Hijaak"!

Installation itself is quite simple; the 3.5 inch disk I received contained an IN-STALL program, a README.TXT file with last-minute updates that didn't make it into the printed manual, a small utility program for Tandy users, a copy of Central Point's MI.COM program which displays information about programs currently loaded into your machine's RAM, and a compressed file containing the Central Point Backup program files. Typing INSTALL and pressing <ENTER> brings up the first installation screen, in which you choose to perform a first-time installation, modify a previous installation, or install Central Point Backup on a network server. At any installation screen, you can press <ESC> to exit if you wish to start over.

One important note for those of you who have used the earlier version to perform incremental backups (regular backups of files that have changed); Central Point has made enough changes in Central Point Backup that it is strongly recommended that you perform a new full backup with this version before resuming your incremental backup schedule. Further, I suggest you keep your last backup set until you've had the chance to both backup AND restore your files with Central Point Backup. This isn't a "version 1" piece of software, but whenever substantial changes are made to a product, and your files are potentially at risk, prudence is the best course!

During installation, you describe your video monitor, choose a directory for the Central Point Backup program files, identify the location of your AUTOEXEC.BAT file (in case you choose to modify it later to facilitate use of Central Point Backup), and decide if you want to install the files and configure the program for your machine rather than doing a simple file copy. By default, the files are copied into a subdirectory called C:\CPBACKUP. You can opt to have the scheduler installed as a TSR (Terminate and Stay Resident) program on boot-up, so that you can schedule backups that will occur at any date and time you specify. The CPBSCHED program will be inserted in your AUTOEXEC.BAT file after the PCCACHE and before the PCSHELL statements, if they exist. Finally, you can choose to protect Central Point Backup with a password to guard against unauthorized use, and you can set User Level to Beginner, Intermediate, or Advanced, depending on how many commands you want (or want your network users) to be
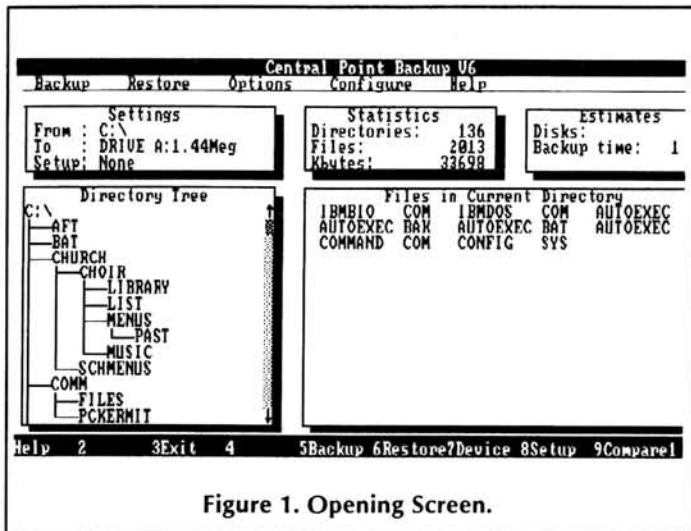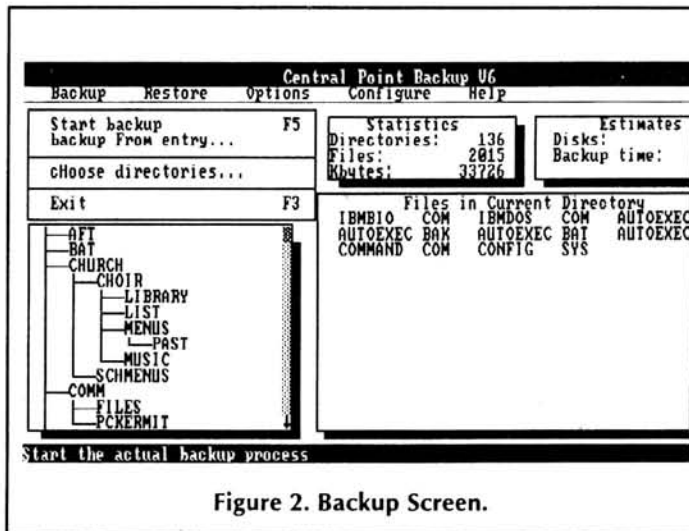

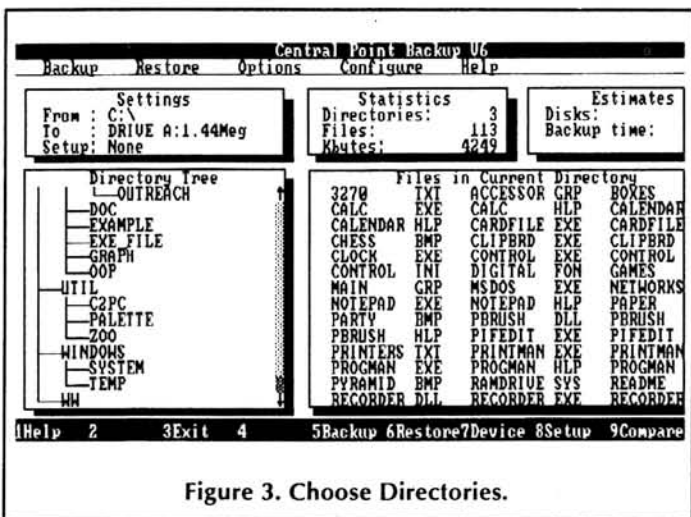Figure 1. Opening Screen.

Figure 2. Backup Screen.

Figure 3. Choose Directories.

Figure 4. 23% Done.

Figure 5. Backup Complete.

Backup and Restore Time Trials
Using CP Backup, Version 6

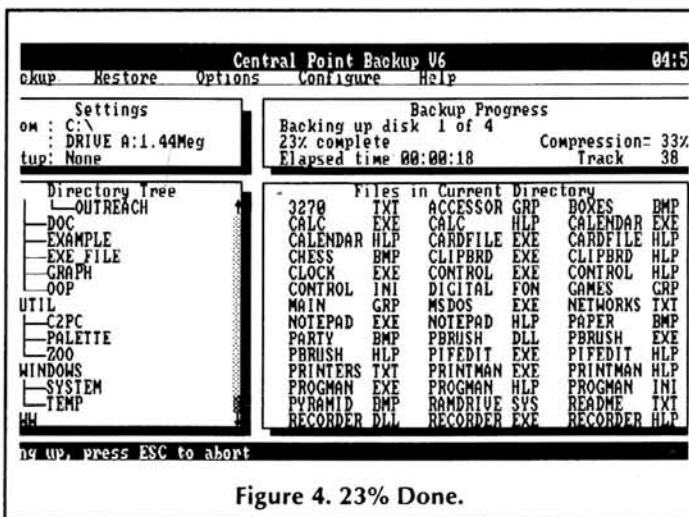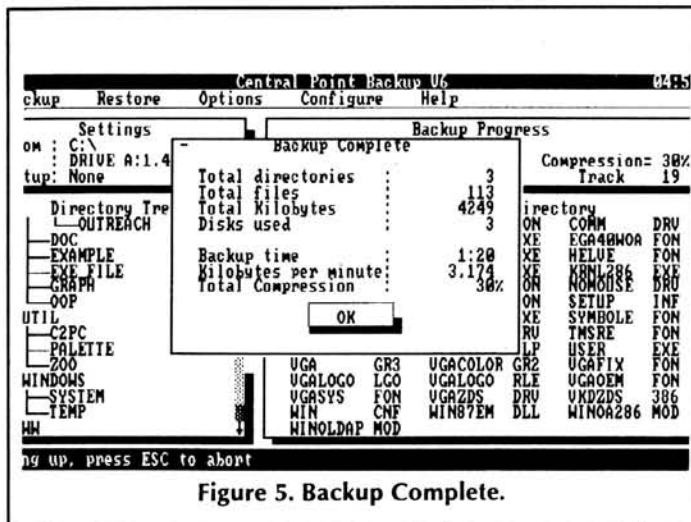| Machine	Files	Size (MB)	Action	Time	(MB/min)
Z248/12	181	3.3	Backup	1:02	2.81
Z248/12	181	3.3	Restore	2:00	1.64
Z248/12	113	4.2	Backup	1:20	3.17
Z248/12	79	2	Backup	0:40	3.00
SSprt-SX	118	6.9	Backup	4:59	1.38
SSprt-SX	118	6.9	Restore	3:30	2.00

Figure 6

able to use. After you respond to these queries, the program files are copied to the specified directory, and any changes you have authorized will be made to your AUTOEXEC.BAT file.

At this point, I started the program (type CPBACKUP and press <ENTER>), and encountered a Welcome screen that reminded me that I had not yet configured the program for my computer. First, I confirmed the type of backup drives displayed for my machine; a 3.5 inch A: drive and a 5.25 inch B: drive. If my machine had a tape drive installed, I would have been prompted to confirm that choice as well. On the next screen, I chose the backup media I use in each drive; 1.44 MB diskettes for drive A: and 1.2MB diskettes for drive B:. At this point, I was offered a chance to have a "backup confidence" test performed on the backup floppy/tape drive of choice and a spare diskette or tape. If your hardware passes this test without error, backup speed will be set to High by default. If you choose to skip this test, your default speed will be left at the Medium setting. My machine, which runs Zenith's version of MS-DOS 3.3 Plus and uses BIOS version 3.30.05, drew a warning that it "might not handle high speed DMA data transfer". I chose the High speed option, anyway, and resolved to check my first few backups for corrupt data or difficulties in restoring files. When DMA (Direct Memory Access) mode is used, data can be written to your backup media at the same time subsequent data are being read, which can save a substantial amount of backup time.

As it turned out, my machine passed the Backup Speed Test with no problems. This isn't the first time a Central Point Backup utility doubted the compatibility of the DMA hardware Zenith uses (see pp. 19-20 of the March, 1990 issue of REMark magazine for an example with PC Tools version 5 and a Z159!), but I never have had any problems as a result. At this point, I re-booted my machine; it was . . .

Test Drive Time

When you first start Central Point Backup, the program analyzes your hard disk, and then displays the screen you see in Figure 1. There are five "pull-down" menus at the top which can be activated either by a mouse or by the ALT-[firstletter] combination. The sections showing your Settings, Statistics, Estimates, Directory Tree and Files in Current Directory are displayed in drop-shadow boxes using a new, brighter color scheme than previous PC Tools screens. At the bottom of the screen is the familiar message bar, which describes the function of any "short-cut" function keys, and also displays a one-line description every time you activate one of the pull-down menus. Note that all files and all directories are selected for backup by default, and that if you could see the full screen image represented by Figure 1, you would note that it would take about 25 1.44 MB diskettes and about 18 minutes to back this all up!

The first step to running any backup other than a full disk backup is to choose the directories of interest. Central Point Backup allows you to customize backups in any way imaginable, so that you can define a specific set of files or file types to back up and save that setup for future use. In my tests, I performed simple backups of entire subdirectories, but be aware that this does not fairly represent the power of this backup program.

Everyone needs a "backup philosophy." I work with some people whose philosophy is "I'll start regular backups the day after I lose something critical." Others with an unlimited supply of diskettes or a tape drive simply back everything up. The most practical approach for me is to back up only my work, plus any configuration files created when I install software. I rely on the backup disks I make of new software when it arrives to restore any program files that get lost or corrupted. In fact, I confess to a recent laziness (frugality??) of mine encouraged by some software manu-

facturers who use the famous "dual disk set." If my software arrives on both 3.5 inch and 5.25 inch disks, and all media are readable, I write-protect all disks and rely on the dual set to preserve the program files. Still, there's no one to protect your own work but you; it's important to remember that!

When you select the "Backup" pull-down menu, the screen resembles Figure 2. By highlighting the "cHoose directories..." with the arrow key, and pressing return, you can prepare a simple subdirectory backup. I chose to test this program by backing up the three subdirectories created when we installed Windows 3 (C:\WINDOWS, C:\WINDOWS\SYSTEM, AND C:\WINDOWS\TEMP). Once selected, the Statistics window changes to indicate that 3 subdirectories, containing about 4,249,000 bytes of data in 113 files have been selected (see Figure 3). The Estimates changed as well: Central Point Backup figured that I would need four diskettes and about a minute to complete this backup. At this point, I pressed <F5> to start the backup.

Central Point Backup tracks the progress of the procedure for you, as you can see in Figure 4. This screen shot was taken 18 seconds into the first backup disk, when the entire backup was about 23% complete. Compression of files during backup is the default option, and the screen indicated that current compression was able to reduce the size of the backed-up files by about one-third. The current file being backed up is highlighted in the Files window (which isn't evident from Figure 4), and you can stop the process in mid-stream by pressing <ESC>. There's little to do now but get a cup of coffee or flip through the well-made manual to pick up other usage hints.

Results

Every 30 seconds or so I was reminded to place a new diskette into the backup drive (A:) that I had selected. Though

the "Estimates" box had prepared me to provide four 3.5 inch diskettes, the actual backup only required 3 (see Figure 5). After one minute, 20 seconds of actual backup time (add about 30 seconds for my fumbling with diskettes!), the program finished. The speed was a respectable 3.17 megabytes per minute, and the files used allowed an overall compression rate of about 30%. I ran a few other trials, backing up collections of large and small files on my Z-248, and even ran a test on a Zenith SupersPort SX (386SX laptop) machine we were using at work. The results are summarized in Figure 6.

In general, Central Point Backup moved about 3 MB/minute to 3.5 inch diskettes, and was fairly accurate in estimating backup time. The diskette estimate was one too many in most cases, but a slight error on the conservative side leaves you better prepared than one which leaves you scrambling for vacant diskettes in mid-backup! The one record I can locate of a timed restore using PC Backup version 5 on my machine took 3 minutes, 35 seconds to restore 79 WordPerfect files (4.4 MB), a noticeably slower 1.23 MB/minute. So Central Point has indeed improved both speed and capability in the current version. Guess it's time to start saving for a tape drive so I can reap the benefits of late-night full incremental backups.

The Bottom Line

If you already own PC Tools Version 6, it's worth the time and minor expense to download the improved Central Point Backup program files from Central Point's electronic Bulletin Board. You can use the Telecommunications module of Desktop; it already has the BBS number loaded and ready to dial if you have a modem inside or connected to your machine.

If your situation makes it impossible for you to purchase the entire PC Tools package but you need backup software, I suggest you purchase Central Point Backup and begin a regular file backup routine. In the future, you can leverage this investment (plus $50) into a purchase of the full package, which has got to rate as one of the all-time great buys in the software world (right up there with "$99 and a page from your Lotus manual" for Borland's Quattro Pro!).

Finally, if you need fast, reliable, and flexible backup software, and have no other useful collection of software tools, buy the entire PC Tools package up front. Once you become familiar with Central Point's view of what a user interface should look like, you'll find it difficult to endure any other. Why else would Windows 3 users like me prefer running PC Shell as a non-Windows application to using the built-in Windows File Manager??
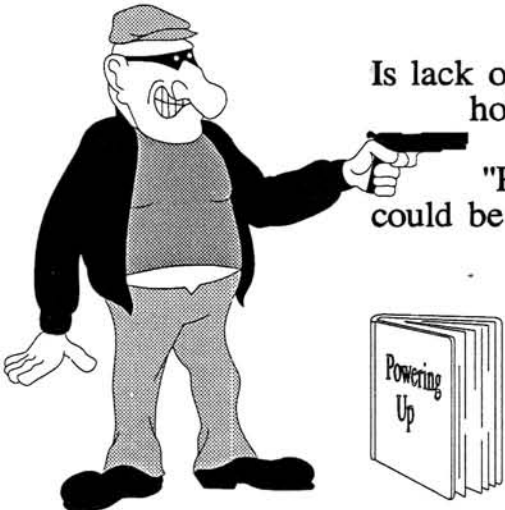
Final Words

If you have any questions about using Central Point Backup, feel free to send them directly to me or to Buggin' HUG. After all, that's what User's Groups are for! Now if you'll excuse me, it appears that there's this snoring baby in a snowsuit under my computer table, and if I don't see to her soon, she'll probably roll over on top of my power str...

Items Discussed

Central Point Backup	$ 99
PC Tools Version 6	$149

Central Point Software
15220 N. W. Greenbrier Parkway
Suite 200
Beaverton, OR 97006
(503) 690-2236   ✳

Dan Sadler
Technical Writer
Zenith Data Systems
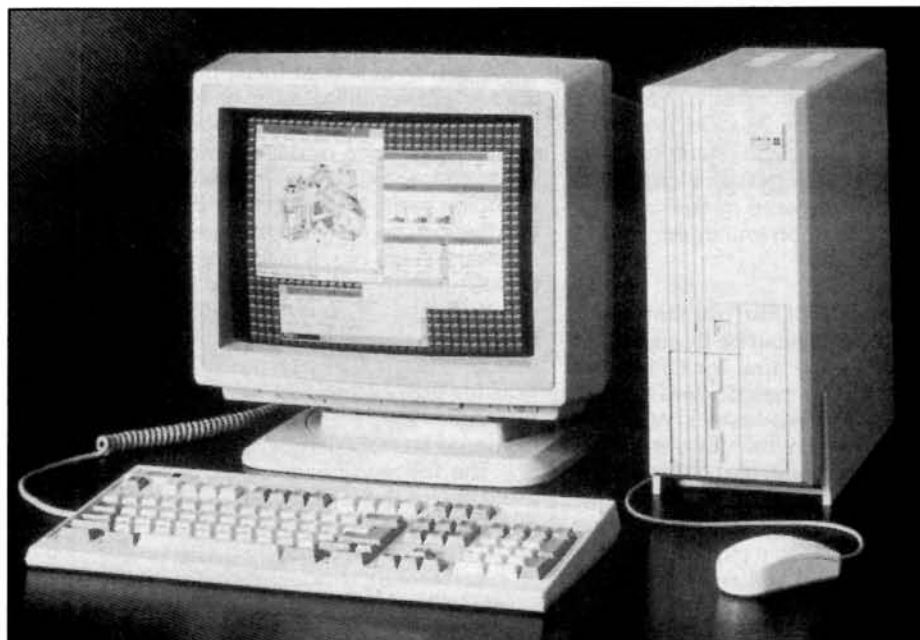
An Introduction to the Z-486/25E

It's been said that a mans best friend is his dog. Depending on the man's profession, today's dog may have some competition. But wait, a computer is a computer, and not often thought of as a *friend*, right? Perhaps, but I suspect present and future (personal, as well as professional) computer users may disagree.

An Overview

Meet the Z-486/25E, Zenith Data Systems' first 486-based EISA (Extended Industry Standard Architecture) desktop. Based on Intel's 25 MHz 80486 microprocessor, this machine is definitely no slouch.

The 80486 is much more than your average microprocessor. In addition to a 32-bit wide address and data bus, this CPU internally incorporates the following:
- 8 kilobytes of cache memory
- a 4-bit even parity generator/checker
- an 80387-equivalent numeric coprocessor
- a write queue.

Some of these devices are not really new and have appeared in previous Zenith Data Systems computers, but all were individual components requiring valuable physical board space.

The Z-486/25E weighs approximately 18 lbs., and the base model ships with the following:
- a 101-key keyboard with AT/XT switch
- one 3.5-inch, 1.4M floppy disk drive
- one 3.5-inch, 80M hard disk drive
- 4M RAM on memory board (direct

hardware support for 63M EMS)
- a VGA video card (supports EGA, CGA, and MDA formats)
- EEPROM for user password
- a lithium backup battery.

A Private Stash Called Cache

The cache memory scheme itself is old-hat, but I am going to discuss it in hopes to clarify any doubts or questions. To better understand cache memory, try to visualize the processors cache RAM as a refrigerator, and main or system memory as the grocery store. For most people a trip to the grocery store involves getting in their car, driving to the store, dealing with other traffic, finding a parking space, purchasing the items, driving home, and putting away these items.

The CPU follows a similar route called a "read cycle". When it needs to read in a new string of data, it must finish the present task, flush or empty the write queue (discussed later in this article), take control of the bus, pass address and control information to the memory controller, and once delivered, store it in cache memory.

The 8K cache RAM within the CPU is capable of storing 128 individual data strings. Each data string consists of four dwords (double-words), each dword being four bytes (or 32 bits) long, for a total string length of 16 bytes. This provides the CPU with its own storage area similar to our refrigerator.

Information Anticipation

I know what you're thinking; "Shouldn't the cache data strings be 4 bytes (1 dword) in length instead of 16 bytes (4 dwords), as the microprocessor is a 32-bit device?" Good question.

Enter the memory controller. First, one must understand the function of a memory controller is to decode processor address bits, determining what specific area of main memory the CPU wants data from, and provide the necessary memory read or write commands, along with other control signals required by DRAM devices.

With Zenith Data Systems' *burst-mode* memory controller, the CPU is provided with its requested (32-bit) dword from main memory, as well as three other dwords in that area of memory (hence the 16-byte vs. 4-byte data string). These additional three dwords can be thought of as the "next most-likely accessed" block of information (like buying a case instead of a six-pack). By storing them in cache memory, we reduce our trips to the grocery store, and improve overall efficiency.

For example, if the CPU requested data from 1000H, the memory controller would transfer that dword, along with dwords from 1004H, 1008H, and 100CH. The idea here being that we are anticipating CPU data requests. By the same token, if the CPU requested a dword from 1008H, it would receive dwords from 1000H, 1004H, 1008H, and 1000CH.

Zenith Data Systems recommends that

SIMMs (single in-line memory modules) be installed in pairs of equal-density modules. This allows for a 64-bit data bus width configuration required for fast-burst transfers.

Great, hopefully now you understand a little bit more than you did regarding cache memory and the burst-mode memory controller. Now let's take a look at the functionality of the write queue, another interesting contraption within the Z-486/25E.

Write Queue... What Do You Do?

The write queue can be thought of as a temporary storage area for CPU write cycles, where cache memory was defined as a storage area for read cycles. When the CPU writes to main memory, it simply loads the address, data, and control information into the write queue. At the same time, the CPU checks its cache memory to see if it has a copy of that particular area of main memory being written to. If it does, cache is updated, and the CPU views the cycle as complete. By posting write cycles, the CPU becomes available to process other tasks and calculations earlier than it would in computers without a write queue.

While the CPU loads write cycles into one side of the queue, the memory controller is pulling them out at a slightly slower rate on the other side. The reason the memory controller cannot keep up with the CPU is that other devices on the memory controllers side of the queue also take over the bus. Can you say "Memory refresh"?

If some other device (such as the DMA controller, or another bus master in one of the EISA expansion slots) writes to main memory, the memory controller must check and see if that string of data being overwritten is also present in the CPUs cache memory. If so, the memory controller informs the CPU that one of the data strings within its cache RAM no longer matches main memory. The CPU marks that data string as invalid, and overwrites it the next time a new data string is brought in on a read cycle.

The queue within the CPU can store up to four write cycles, and the external queue is capable of storing up to eight write cycles. When the queue is full, the CPU must wait until one of the previously-posted write cycles has been executed before loading another cycle. If the CPU initiates a read cycle, it must wait until ALL posted cycles have been executed by the memory controller. This process is known as "flushing" the queue.

Parity, a Parody?

Unlike previous Zenith Data Systems computers using odd parity detection, the Z-486/25E uses an even parity generation and detection system. The main reason for the change from odd to even is because that's the way Intel designed the internal parity circuits within the 80486.

A parity bit is generated by the CPU (also by the EISA bus buffers) and transferred with every data byte that is written. When this byte is read back, the internal parity checker verifies that even high-bit (or logic 1) parity has been maintained. For example, if a byte of data contained bits 00110011, the CPU would pull the corresponding parity bit for that particular byte **low** (logic 0), as there already are an EVEN number of high bits. If the same byte read 00000111, the parity bit would be **high**.

If even parity has not been maintained, the SIMM location will be displayed on the monitor, usually followed by the computer "locking-up", as the CPU stops execution.

Optional Hardware

The following hardware options are available for the Z-486/25E:
- Weitek 4167 coprocessor.
- TIGA video card (requires 1024 x 768 pixel resolution monitor). Some models of Z-486 ship with this video system.
- Support for 12M of ISA-compatible (16-bit) RAM.
- Support for 4G of EISA-compatible (32-bit) RAM. Total of 4G RAM including the 12M of ISA-compatible RAM.
- One 5.25-inch (720K or 1.4M) floppy disk drive, mounted with adapter bracket in the top drive bay position.
- One SCSI magnetic tape-backup unit, mounted in the top drive bay position.
- Support for up to 7 external SCSI hard disk drives.
- One 64K OEM (expansion) ROM socket.

And Now the Bad News...

As with everything, a certain amount of bad accompanies the good. This may be personal opinion, but I believe the following aspects of the Z-486/25E are shortcomings:
- The EISA configuration utility,
- small chassis,
- lack of EISA expansion slots,
- small power supply.

Although there are no hardware jumpers or switches within the computer that must be reconfigured when adding optional hardware, there is no firmware-based Setup program either. Many Zenith Data Systems computer users are accustomed to pressing CTRL-ALT-INS, followed by typing setup to access this program when verifying or changing hardware configuration.

Instead, a disk is shipped with the computer containing the EISA configuration utility. Say goodbye to the days of quick configuration changes, and hello to another piece of software, and 29 pages within the owner's manual explaining it. I miss the Setup program.

I'm sure some will disagree, but it seems to me that the expandability con-

cept was a driving force behind the EISA design. This computer is housed in a convertible mini-tower chassis, with a small backplane board offering only 3 available card slots (at approx. 2A each), and a 150 watt power supply. Seems to defeat the purpose of expandability...

But then again, at least it doesn't ship with that 3 boards-in-one, 2 card slot consuming, 5-amp miscarriage of a SCSI/ESDI/ floppy controller card known as *ALF*, as was the case with the Z-386/33. Maybe that's the reason why there is a 50 watt difference in power supplies between Zenith Data Systems' 386-based and 486-based EISA computers.

In conclusion, the two features of this computer that impress me the most are the CPU's built-in features (numeric coprocessor, cache memory), and the burst-mode memory controller. Priced at $8599 suggested retail price for the base model, it's built for speed. ✳
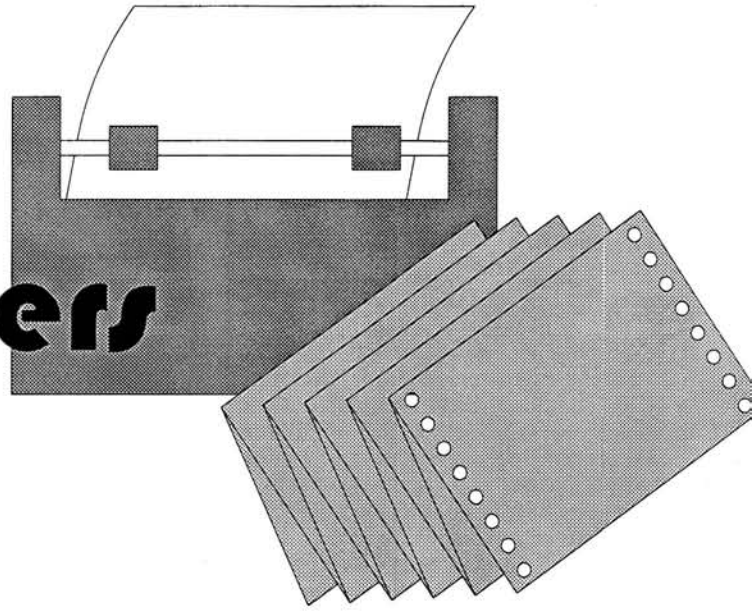

We are still trying to get caught up on our issues.

We would like you to know how much we appreciate your patience. Thank You!

Dis an Dat About Printers

Laszlo Vesei
420 N. Philip Road
Niles, MI 49120

Word processors have a rather limited value without a printer. I found this out the hard way in '81 when I acquired my first computer, an H-89. It depleted my treasury somewhat; therefore, it took awhile until the printer, an H-25 followed. In those days, it was still Heath Company's custom to bundle software with the hardware so, among others, Magic Wand came into my possession. In the following years, I learned to use the printer and the Wand and developed a love/hate relationship with them.

No doubt, the H-25 printer is very rugged, even (or perhaps especially) with today's standards. However, it is NOISIS-SIMUS and at that time it was strictly limited to English text. Just as I became sick and tired to manually apply the Umlauts (diacritical marks) to foreign correspondence, a clever gentleman, Mr. Fina came out with a new ROM chip set to replace the originals. It was now easy to print the high ASCII characters — if only the word processor were capable of handling them. (I reported it in the July 1987 issue of REMark.)

Alas, Magic Wand could do it only with heroic effort. Briefly, I experimented with the old CP/M WordStar, but it was s-l-o-w-e-r than molasses in January and expensive to boot.

About this time, some money came my way, so I obtained a H-161 which was bundled (among others) with the successor of Magic Wand, called PeachText. Shortly thereafter, a Gorilla-Banana (ugh) printer came into my possession. The latter was gotten rid of rather fast and replaced by a Citizen 120D.

Now, this is an excellent 9-pin printer, better than most. It is relatively quiet, has a small footprint, capable of many things and inexpensive to boot. If the pull-only paper-feed and the somewhat awkward loading of the paper is of no concern, I can heartily recommend it. I have found only one thing to gripe about: the capital letters Ü and Ö are at the wrong spot, where ü and ö should be and versa-vice. Of course, if you learn to live with this idiosyncrasy or you limit yourself to the king's English, then this should be no problem to you.

The Citizen printer can emulate either an IBM Proprinter or 3 different Epson models. In Epson mode, it refuses to print the accented characters, but is able to print Italics. In IBM mode, it prints the full extended ASCII characters and also the slashed zero, but alas, no Italics.

In IBM mode, it works fine with the usual word processors, as long as the native Courier font, Pica or Elite suffices. However, if one wants to take advantage of all the capabilities of this little jewel, one needs a word processor which uses embedded printer commands, such as Peach-Text does. GW-BASIC will do it, too. With the appropriate commands, this printer can print double-tall, double-wide, both of these, proportional, negative (white letters on black background), not to mention the various letter and row spacings. In graphic mode it will print photographs and AutoCAD drawings, too.

At the beginning there was a problem getting ribbons for this printer. I learned to revitalize the ribbon by means of WD-40. It works, but it is somewhat messy. Also it necessitates carefully prying apart the cartridge. Once it is done one might as well reload the cartridge with an easily obtainable ribbon of the same width and quality, intended originally for some other printer. Manual dexterity required, also a good handcleaner.

Another use of WD-40 is to free stuck pins on the printer head. On this machine the head is easy to replace without tools, but pricey. WD-40 can not do any damage and may do some good. Incidentally, the ink in a good ribbon is supposed to contain some lubricant for the pins.

Eventually, the need arose to produce better quality print than a 9-pin printer is capable of. My modus vivendi (tempus fugit=retired) included a laser printout, so a rather intensive study of the market followed. As the result of the study, considering the recommendations of friends and several magazines and last, but not least, finances, I purchased a Panasonic KX-P1124.

Now, this is one fine machine, but it took a month and several phone calls to learn to live with its idiosyncrasies. In my experience, I used one time or another at least 30 different printers, they all have subtle (or sometimes not so subtle) differences. The Great Stumbling Stone is the Operating Instructions booklet. I am sure it was written originally in Orientalese, then translated into English, back into Orientalese, and into English again. The translators, no doubt, were PhD-s in languages, but innocent of anything technical, in general, and computers, in particular. Several phone calls were needed to throw some light on the more arcane points of the instructions.

These phone calls deserve special mention. Theoretically, the technical information service of Panasonic in Secaucus, N.J. is open from 8 AM to 5 PM EST, 5 days a week. If you call one second before 8 AM or one second after 5 PM, you get an answering machine which does not take messages. Any other time, the number is busy. But if you have a Hayes-compatible modem and Procomm software (or a

Demon Dialer), you can dial their number every 30 seconds automatically, until someone answers. Once you get that far, you are in clover. The people I talked to were very nice and knowledgeable. But getting thru to them?! Yecch! (Lately, they saw the light and service calls are answered now by a human operator, who takes your name and phone number. A technician will call you back, usually within 24 hours.)

Most self-respecting printers have a small buffer built in and the Panasonic KX-P1124 is no exception. It has an 8K buffer. Furthermore, it has arrangements inside to add a serial interface if desired, (I did not) and an additional buffer chip (I did). If graphics are considered, the additional buffer is a conditio sine qua non, however, it is somewhat pricey. Also, do-it-yourselfers need plenty of intestinal fortitude to take apart a perfectly good printer for this surgical procedure. The DIP buffer chip has the orienting notch on one side, rather than on one end, so the inserting is pure guesswork, despite the otherwise clear instructions that came with the chip. (The pricey chip did not get smoked, but it was pure luck...)

After using this printer for several months I fell in love with it. (My previous Citizen printer is now the prize possession of a college kid.) Now, let's see some of the features:

When the new printer awakens the first time it is in a "factory set" mode. It can be reset to either LQ-2500 or IBM X24 mode if so desired, but why bother if word processing follows. For any other purpose, see the best setting later. In any case, the permanent memory inside will remember these settings indefinitely. (Static RAMs?)

There is a rather longish list of printer commands for those using BASIC or a word processor with embeddable commands. Me, I am using TEXTRA as the word processor. Textra lets you use 15 different type faces, each having 7 fonts, provided your word processor is set for Panasonic KX-P1124. In LQ-2500 or IBM mode, you are restricted to 7 typefaces (still with 7 fonts.)

An explanation of type faces: There are 5 resident in the machine: Courier, SansSerif, Prestige, Script, and of course, Draft. These can be had in Pica (10 pitch), Elite (12 pitch), wide widths, narrow widths, double heights, proportional and in various combinations of these. As for fonts, there is Bold, Underline, Italic, Strike thru, Double underline, Superscript and Subscript. (With TEXTRA you can easily edit the fonts. TEXTRA also can remember the setting of 4 different printers at the same time.)

Paper can be fed from the top rear or the bottom rear. I miss the bottom feed, but the ease of loading the paper more than makes up for it. There are 3 paper

feed modes: Friction, pull tractor and push tractor. Friction feed is for single sheets, the tractor feed is for perforated edges. My favorite is the push feed: it does not waste the first sheet, also it has a "parking" mode. In this mode, it retracts the paper so as not to interfere with subsequent friction operation. There is a full width door in the front, where small forms (tags, envelopes, etc.) can be introduced. After this operation, it takes a push of a button to automatically return the tractor-fed paper to its previous position. In 5 months of average usage (and 2 big packages of laser-cut paper), I still have to experience my first paper jam.

The ribbon needs mention too. The printer came with two ribbon cartridges. After a couple of months the first one faded out. It is always smart to look up the instructions before doing things. So, I did just that before opening the second package. Lucky me! There is a pad in the back of the cartridge. The book said to push it with a pencil tip until it clicks and put the spent cartridge back into the printer. Lo and behold, I heard the click and sure enough, the print gradually became black again, to the point that it became necessary to back the head away from the paper (there is a lever for this). Now, after 3 additional months, I am still on the first, revitalized cartridge.

As an afterthought, I want to mention, that I used the printer with a CAD program (See Figure 1) as a plotter. It works fine, although the plotter does a better job.

After saying all this I owe the reader several things:

First, some samples of the printers capabilities (See Figure 2).

Then a BASIC program (See Figure 3) to print out the extended ASCII table (See Figure 4) with the Panasonic KX-P1124. A word of caution: The program shown does not work with any other printer without substantial modifications!

To use this program, or any other BASIC program for that matter, that must print out reasonably good looking strings, the printer must be set to the set-up conditions as follows:

Emulation mode: Proprinter X24 G2
Default font: Courier
Int'l character set: USA
Data length: 8 bits

All other setup conditions are optional.

The accompanying ASCII printout program shows an unusual way to address the printer. The printer is set up as a random access file. The BASIC command "LPRINT" sometimes results in unexpected goings on with the extended ASCII set. (Probatum est.)
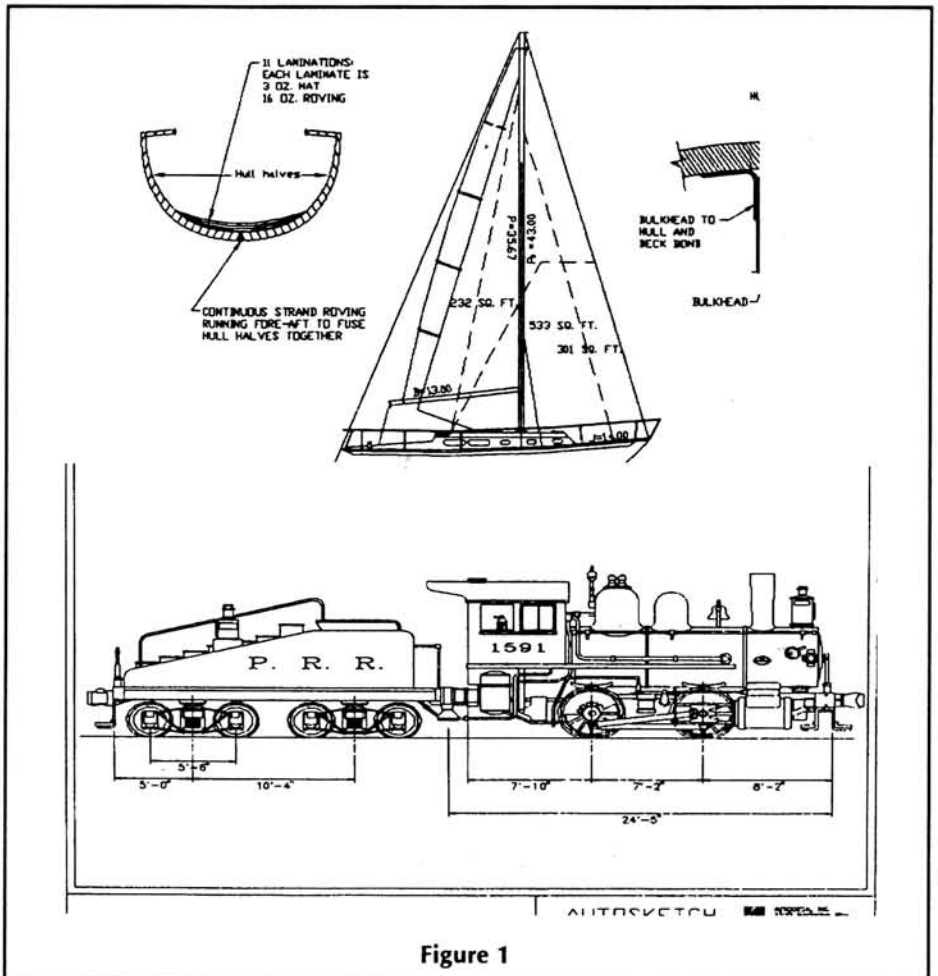
For word processing, the printer setup

Figure 1

Figure 2

FONT TEST:

This <u>line</u> is in font 1. Pica Courier
This <u>line</u> is in font 2. Elite Courier
This <u>line</u> is in font 3. Wide 5 CPI
This <u>line</u> is in font 4. Pica Prestige
This <u>line</u> is in font 5. Compressed 17 CPI
This <u>line</u> is in font 6. Script Pica
This <u>line</u> is in font 7. Wide 8.5 CPI
This <u>line</u> is in font 8. Proportional

This <u>line</u> is in font 9. Tall Proportional

This <u>line</u> is in font 10. Tall Wide
This <u>line</u> is in font 11. Tall Wide 6 CPI
This <u>line</u> is in font 12. Pica Sans Serif
This <u>line</u> is in font 13. Tall Pica
This <u>line</u> is in font 14. Tall Elite
This <u>line</u> is in font 15. Compressed Tall 17 CPI

STYLE TEST:

This line is bold.
<u>This line is underlined.</u>
This line is italic.
~~This line is strike-through.~~
<u><u>This line is double underlined.</u></u>
This line is superscript.
This line is subscript.

This <u>line</u> has all-seven styles included.

<u>This line is both bolded and underlined.</u>

MICRO-JUSTIFICATION TEST

This paragraph is being microjustified. This means an equal amount of space is being placed between each word to give the justification a smooth look. This line has a few styles like <u>underline</u>, <u><u>double underline</u></u>, and ~~strike-through~~ in it.

This paragraph is being microjustified. This means an equal amount of space is being placed between each word to give the justification a smooth look. This line has a few styles like <u>underline</u>, <u><u>double underline</u></u>, and ~~strike-through~~ in it.

This paragraph is being microjustified. This means an equal amount of space is being placed between each word to give the justification a smooth look. This line has a few styles like <u>underline</u>, <u><u>double underline</u></u>, and ~~strike-through~~ in it.

Figure 2

Figure 3

```
10 CLS:PRINT:PRINT
20 PRINT TAB(15)"Select font:":PRINT
30 PRINT TAB(20)"Courier - - - - - - - - 0"'
40 PRINT TAB(20)"Sans Serif - - - - - - - 1"'
50 PRINT TAB(20)"Courier - - - - - - - - 2"'
60 PRINT TAB(20)"Prestige- - - - - - - - 3"'
70 PRINT TAB(20)"Script- - - - - - - - - 4"'
80 PRINT TAB(20)"Bold PS - - - - - - - - 6":PRINT'
90 LINE INPUT"Your selection - - - - ";F$
100 IF F$="" OR F$="5" THEN 10
110 FF=VAL(F$)
120 IF FF>54 THEN 10
130 B$=CHR$(27)+CHR$(94)
140 LPRINT CHR$(27)+CHR$(107)+F$
150 LPRINT TAB(20)"┌"STRING$(41,"-")"┐"
160 LPRINT TAB(20)"│ *** The PANASONIC  ASCII Characters *** │"
170 IF FF=0 THEN S$="-- Courier --"
180 IF FF=1 THEN S$="--SansSerif--"
190 IF FF=2 THEN S$="-- Courier --"
200 IF FF=3 THEN S$="-- Prestige--"
210 IF FF=4 THEN S$="-- Script  --"
220 IF FF=6 THEN S$="-- Bold PS --"
230 LPRINT TAB(20)"│              "S$"              │"
240 LPRINT TAB(20)"└"STRING$(41,"-")"┘":LPRINT
250 WIDTH "lpt1:",255' ------ Set up printer as a
260 OPEN "lpt1:" AS #1'------ random access file
270 FOR X=0 TO 25
280 FOR Y=0 TO 9
290 Z=10*X+Y
300 IF Y=0 THEN V$="" ELSE V$="   "
310 IF Z=33 THEN V$=V$+" "
320 IF Z=256 THEN 380
330 PRINT #1,V$;:PRINT #1,USING"###";Z;
340 IF Z=32 THEN 360
350 PRINT #1," "B$CHR$(Z);
360 NEXT Y
370 LPRINT:LPRINT:NEXT X
380 LPRINT CHR$(12):CLS:END
```
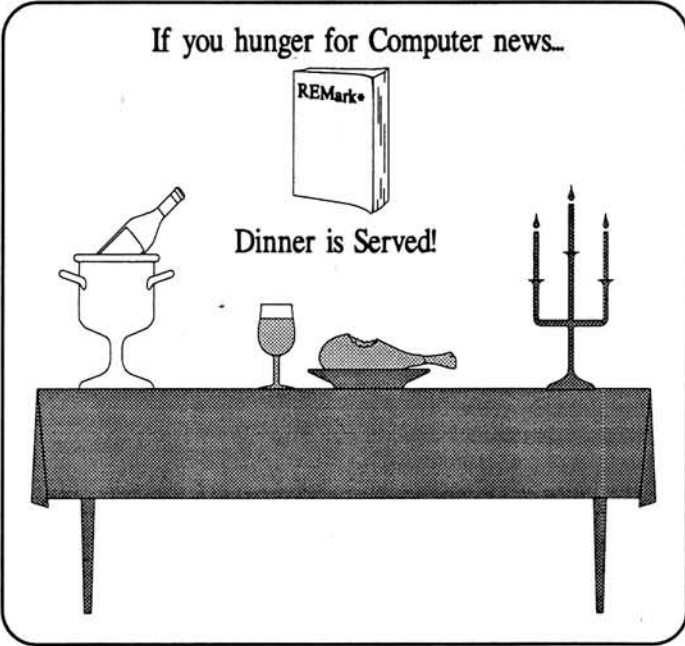
Figure 3

conditions must be reset to suit the word processor, of course. Presumably, there is an automated way to do this, but I did not find it yet. Repeatedly, I pestered both Ann Arbor Software (the makers of TEXTRA) and Panasonic tech service for an automated (software) way to switch the printer from one mode to the other. If they know, they don't tell. (I suspect that TEXTRA knows and Panasonic does not know.)

Despite this small inconvenience, the Panasonic KX-P1124 is an exceptionally good printer, the next best thing to a laser printer. I'll be using this printer for a long, long time. As Jerry Pournelle would say: "Highly recommended". ✷

Figure 4

```
*** The PANASONIC  ASCII Characters ***
             -- Courier --
```

0	1	2	3	4	5	6	7	8	9
0	1 ○	2 ●	3 ♥	4 ♦	5 ♣	6 ♠	7 •	8 ◘	9 ○
10 ◙	11 ♂	12 ♀	13 ♪	14 ♫	15 ☼	16 ►	17 ◄	18 ↕	19 ‼
20 ¶	21 §	22 ▬	23 ↨	24 ↑	25 ↓	26 →	27 ←	28 ∟	29 ↔
30 ▲	31 ▼	32	33 !	34 "	35 #	36 $	37 %	38 &	39 '
40 (	41)	42 *	43 +	44 ,	45 -	46 .	47 /	48 0	49 1
50 2	51 3	52 4	53 5	54 6	55 7	56 8	57 9	58 :	59 ;
60 <	61 =	62 >	63 ?	64 @	65 A	66 B	67 C	68 D	69 E
70 F	71 G	72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W	88 X	89 Y
90 Z	91 [	92 \	93]	94 ^	95 _	96 `	97 a	98 b	99 c
100 d	101 e	102 f	103 g	104 h	105 i	106 j	107 k	108 l	109 m
110 n	111 o	112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 {	124 |	125 }	126 ~	127 ⌂	128 Ç	129 ü
130 é	131 â	132 ä	133 à	134 å	135 ç	136 ê	137 ë	138 è	139 ï
140 î	141 ì	142 Ä	143 Å	144 É	145 æ	146 Æ	147 ô	148 ö	149 ò
150 û	151 ù	152 ÿ	153 Ö	154 Ü	155 ¢	156 £	157 ¥	158 ₧	159 ƒ
160 á	161 í	162 ó	163 ú	164 ñ	165 Ñ	166 ª	167 º	168 ¿	169 ⌐
170 ¬	171 ½	172 ¼	173 ¡	174 «	175 »	176 ░	177 ▒	178 ▓	179 │
180 ┤	181 ╡	182 ╢	183 ╖	184 ╕	185 ╣	186 ║	187 ╗	188 ╝	189 ╜
190 ╛	191 ┐	192 └	193 ┴	194 ┬	195 ├	196 ─	197 ┼	198 ╞	199 ╟
200 ╚	201 ╔	202 ╩	203 ╦	204 ╠	205 ═	206 ╬	207 ╧	208 ╨	209 ╤
210 ╥	211 ╙	212 ╘	213 ╒	214 ╓	215 ╫	216 ╪	217 ┘	218 ┌	219 █
220 ▄	221 ▌	222 ▐	223 ▀	224 α	225 ß	226 Γ	227 π	228 Σ	229 σ
230 µ	231 τ	232 Φ	233 Θ	234 Ω	235 δ	236 ∞	237 φ	238 ε	239 ∩
240 ≡	241 ±	242 ≥	243 ≤	244 ⌠	245 ⌡	246 ÷	247 ≈	248 °	249 ∙
250 ·	251 √	252 ⁿ	253 ²	254 ■	255				

Figure 4

What is it, Really???
Part 2

Ed Mosher
ZUG Staff

Here is a list of the ZUG Software written for the H/Z-100 series of non-PC compatible, dual-processor computers, along with a short description of what each offering is and/or does. This list is not intended to give you all of the specifics of the program; just a basic description to help you decide if you need further information. I hope you find this listing helpful. Further sections will be printed in future *REMark* issues.

Listed in alphabetical order. Please consult the latest Software Price List, found on Pages 2 and 3 of *REMark*, for current prices. These programs require either Z-DOS 1.0 (or higher) or CP/M.

H/Z-100 SOFTWARE (NOT PC-COMPATIBLE)

CardCat

This program permits you to organize information in a manner similar to that found in a library's card catalog. Entries are stored on disk and can be edited and searched using built-in routines. Detailed help screens are included. Z-DOS ver 1.0 or MS-DOS 2.0

CheapCalc

CheapCalc is a minimal, but very useful, spreadsheet program that can introduce you to these types of programs with little cost. With CheapCalc your computer screen becomes a window to a large worksheet where you can enter mathematical problems and see almost instant solutions. Requires CP/M-85.

Checkbook Manager

This is a ZBASIC program that has the ability of displaying your checkbook register with up to nine entries at one time, and lets you search and edit individual entries. Requires ZBASIC.

CP/EMulator

CP/EMulator is a program that lets you run standard 8-bit (8080) CP/M-85 programs under Z-DOS. Nearly every program that runs under CP/M-85 will run under CP/EMulator including MBASIC, word processors, etc. This program also lets you do such things as edit Z-DOS files with CP/M editors.

DBZ

DBZ is a menu-driven, Z-DOS Database Management program designed to be quick, interactive, and flexible. It's great for personal use and powerful enough for small business applications. It will handle up to 500 records, with 26 fields and 1199 characters per record. Features extensive help menus. Requires Z-DOS or MS-DOS on H/Z-100.

Dungeon & Dragons (ZBASIC)

This is a ZBASIC version of the Dungeons and Dragons. This version displays graphic representations of the rooms, halls, and doors in the areas you enter. The object of D & D is to locate the lord master of the dungeon on one of the 50 dungeon levels. Comes with versions for color and monochrome monitors. Requires ZBASIC.

EtchDump

ETCHDUMP is the paired Etch-a-Sketch and Screen Dump programs for ZDOS on the H/Z-100. These programs form a complete graphics package that feature easy drawing of complex designs on the screen that can be saved on an MX-80 printer. Requires ZDOS and ZBASIC for ETCH; Screen Dump is written in Assembly.

EZPlot II

This program is a user-friendly, high resolution graphics function plotting program for engineers, scientists, and just about anyone who would like to have their printer plot curves from disk data. It is menu driven, can plot as many as three functions [i.e., $f1(x), f2(x), f3(x)$] on the same set of axes, and can plot contours or path functions. EZPlot will also sort data, if necessary, and can use data from BASIC, FORTRAN, PASCAL, C, or COBOL language data files.

Games (ZBASIC)

This disk contains several games that require ZBASIC. These include air traffic controller, blackjack, word, and a typing practice program. Requires color memory chips for proper video. May not work properly without color memory.

Games Contest Package

This two-disk package contains color graphic games sent in as entries in a *REMark* game contest. Included are fast action games such as Worm and NYAG, and Blackjack, Bowling, Elevator, etc. Some programs require ZBASIC; all require color monitor and color memory.

Games Package II

Another two-disk package that contains six games for the H/Z-100. Includes two versions of Backgammon, two casino-type games, and two card games. Requires MS-DOS version 2.0 or higher, some games require ZBASIC, and all video planes must be installed. Two games use a light pen and require between 256K and 512K of RAM.

Graphic Games

Several more games, modified to use special graphics abilities of ZBASIC, are included on this disk. Games include SINK, HUGDICE, BATTLE, HANOI, etc. Require ZDOS and ZBASIC.

Graphics

This is a ZBASIC application program used to create drawings using ZBASIC's advanced color graphicS capabilities. It is menu-driven from function key line, and position is shown with a "cross-hair" type indicator. Requires ZDOS or MS-DOS with ZBASIC; color memory and monitor also

highly recommended.

Help Screen
This program creates and places in upper video RAM useful, user-generated help messages that can be requested at any time. Requires ZDOS or any version of MS-DOS, 64K video chips, and Monitor ROM version 2.5 or higher.

HUG Background Print Spooler
HBPS lets you use 64K of memory above CP/M-85 as a printer buffer. When loaded, files being sent to the printer are first directed to the allocated memory, which then sends the characters to the printer. This frees your computer for immediate use, instead of having to wait for the printer to finish. Requires CP/M-85 version 2.2.103; does not work with earlier versions.

KEYMAC
Keyboard Macro Processor for the H/Z-100. With KEYMAC, you can program any key on the keyboard to produce up to 100 characters when pressed. Defined macros are stored in files, and you can create them "on the fly."

KEYMAP
KEYMAP lets you designate the responses produced by your computer's function keys. It works like the KEY command in GW-BASIC except that more keys can be defined. In turn, those keys become part of the "system" and can be used with any program or operating system. For example, you can redefine key F1 to produce the command "DIR A:" (including the return!) so that a single keystroke lists the files on the disk in drive A. Requires MS-DOS 1.25 or higher.

KEYMAP CPM-85
Same as KEYMAP (above), except written for CP/M-85, version 2.2 (and above) operating system.

MathFlash
This is a colorful educational program designed for children who are learning the basic mathematical operations of addition, subtraction, multiplication and division. It emulates the typical "flash cards" used at home. Requires ZDOS; a color monitor is highly recommended, but not required.

Orbits
This disk contains a set of programs which show what the possible orbital paths of satellites look like. Written to help students visualize the basic mathematical concepts involved with orbital mechanics. Requires ZDOS/MS-DOS 1.25 or higher, 64K color RAM chips installed on video board. MPI-99 or compatible printer to get hardcopy of screen prints.

Poker Party
These programs bring to your monitor screen the faces and voices of three rustic cow hands out of the old west who invite you to join their friendly poker game. You'll meet Shorty, Ole, and Cisco who play a conservative brand of poker that's hard to beat. Requires ZDOS/MS-DOS, ZBASIC; color monitor recommended.

SciCalc
SCICALC turns your H/Z-100 into a 14-digit scientific calculator. In addition to providing six math operators, it supports a large variety of transcendental functions, trig, statistics, metric/US conversions, and key math and physical constants. It's also pre-programmed to solve plane and solid geometry problems, and octal and hex math. Requires ZDOS and ZBASIC version 1.25.

Skyviews
Did you ever wonder just what stars were overhead on a given night? With SKYVIEWS, you can plot the positions of major stars (those with 4.0 or greater magnitude), as well as those of the sun, moon, planets and constellations. This astronomer's helper provides right ascension-declination, azimuth-elevation, and other information about the individual celestial bodies, and marks major constellations on the screen to aid in locating them. Requires ZDOS version 1.25 and ZBASIC.

Small-C Compiler
This set of disks forms a complete compiler for the SMALL-C language, which is a true subset of the UNIX C language. The compiler converts a file of SMALL-C statements into a file of MS-DOS Assembly Language code statements. It includes a runtime support library. Full source code is also provided so the user can modify and reassemble the compiler at will. Requires MS-DOS Version 2.0 and above, and two floppy drives.

Spell5
This spelling checker is designed to be quick and easy to use. Dictionary size is limited only by disk space, while the document size is limited by memory. Includes a version for checking spelling in FRENCH, also. Oui! Requires any version of MS-DOS.

Spreadsheet Contest Package
This disk contains five entries from the *REMark* Spreadsheet contest. Programs include Personal Expense, Payroll, Real Estate Investment, Crystal Ball, Taxpro. Four entries require LOTUS 123; the fifth uses Peachtree 5000. All run under ZDOS.

Tree ID
This is a ZBASIC program that aids in identifying various types of trees. It was written to demonstrate a generic menu program for ZBASIC and uses both text and graphics. Requires ZDOS or MS-DOS and ZBASIC, two floppy drives, three banks of 32K or 64K color memory and a color monitor.

Useful Programs
A two-disk collection of utility and application programs for the H/Z-100. Some written in ZBASIC, but compiled so interpreter is not required. Includes screen dump routines, printer controllers, diskette file catalog, etc. Runs on ZDOS or MS-DOS.

Utilities
This disk contains several utilities to aid the user, including two modem programs, a menu executive, printer control programs, a directory display program, and screen dump program. Require H/Z-100 with standard equipment and printer that matches the screen dump routine.

ZPC II
ZPC is a PC emulator program for H/Z-100 series computers. It allows you to run many PC programs on the H/Z-100 without expensive hardware modifications. Supports all video modes, including text and graphics. A list of programs ZPC will run is provided; they include Compiled QuikBASIC and GW-BASIC programs, 123, Word versions 1.1 and 2.0, DBase 3 and 3+ (ver 1), etc. Requires MS-DOS version 2 or 3 and exactly 768K of memory. For best results, your computer should have 32K or 64K of color memory.

H8-H/Z-89/90-H/Z-100 (NON-PC)
Note: All of the following software products require CP/M version 2.2.02 or later and hard-sectored disk (H8;H/Z-89/90), or CP/M-85 and soft-sectored disk (H/Z-100).

Adventure
The 8-bit version of the cave exploration computer game. Adapted for the H8/H-89 with at least 24K of RAM, or the H/Z-100.

HUG Software Catalog Update #1
Describes all of the HUG programs added to our collection during the years 1983 through 1985. A listing of programs added after 1985 was included in each January issue of *REMark*; i.e., the January 1989 *REMark* had all software additions for 1986, 1987, and 1988.

BASIC-E
An implementation of standard BASIC in a P-code compiler and run-time monitor. A Public Domain compiler written by author of CBASIC. Requires at least 32K RAM; however, programs written under BASIC-E may require 48K or more.

Casino Games

A collection of three Casino-type gambling games: KENO, POKER, and VEGAS (Blackjack). Requires 48K RAM H8/H-19 or H/Z-89, or H/Z-100.

CheapCalc

CheapCalc is a minimal, but very useful, spreadsheet program that can introduce you to these types of programs with little cost. With CheapCalc your computer screen becomes a window to a large worksheet where you can enter mathematical problems and see almost instant solutions. Requires 64K RAM on H8, H/Z-89/90.

Checkoff

A Household utility to maintain a balanced personal checking account. Menu driven program permits editing, sorting, merging, and searching of records. Requires 48K RAM and two drives, and Microsoft BASIC 5.21 or higher.

CopyDOS

A program that can copy files from Z-DOS or any 8-sector, 5.25" MS-DOS disk to CP/M disks. Runs on any H/Z CP/M system under CP/M-85 ver 2.2.100 or CP/M 2.2.03 or 2.2.04.

Disk Dump & Edit Utility

DDEU is a versatile disk dump and edit utility for CP/M that is compatible with virtually all H/Z disk systems. Lets user examine and edit disk tracks, sectors, etc. Completely menu driven. Requires 32K RAM and ver 2.2.02 of CP/M or higher.

Dungeons & Dragons

This is a ZBASIC version of the Dungeons and Dragons. This version displays graphic representations of the rooms, halls, and doors in the areas you enter. The object of D & D is to locate the lord master of the dungeon on one of the 50 dungeon levels. Comes with versions for color and monochrome monitors. Requires ZBASIC.

Fast Action Games

A collection of games that tests the users reaction speed. Modified versions of games available for HDOS. Includes Snake, ACK-ACK, Break19, Bob, and Ski. Requires 48K RAM and H-19 terminal or H/Z-100.

Fun Disk I

Five games written in MBASIC. Includes Escape, Dracula, Merlin, Concentration, Mastermind. Requires MBASIC ver 5.X, 64K of RAM, H8/H19, H/Z-89, or H/Z-100.

Fun Disk II

More of the best of HUG games. Contains eight games on four disks. Includes CRIBBAGE, BERSERK, MONOPOLY, KRYPTO, EXRAY, plus others. Some require MBASIC 5.2X; all require 64K RAM and one drive.

Games Disk

Collection of MBASIC games, including: SLOTS, WUMPUS, HANOI, NIM, HANGMAN, OTHELLO, DORNBACK, TICTACTOE, BATTLE, and QUBIC. Requires 48K of RAM, one drive and BASIC-80 ver 5.2 or later.

Grade

A program that lets teachers enter grades into a book in random order. Provides printouts suitable for permanent copies. Fully menu driven with maximum of 100 students and 60 scores per student. Requires 64K RAM, one drive, and MBASIC.

HRUN

This is a CP/M program that emulates the Heath Disk Operating System (HDOS). Allows virtually any non-hardware dependent HDOS program to be run on a CP/M compatible system. Requires CP/M version 2.0 or higher with at least 32K of RAM. Two drives recommended.

HUG File Manager & Utilities

A CP/M version of HFM for PC compatibles. Displays files, lets you copy, delete, etc. Also includes utility programs SYSINFO that details system memory addresses; LONGREL, a file that contains the listings from the *REMark* article "Long Relative Addressing for 8080"; and SEARCH, a routine that adds a search command to the DDT debugging tool.

HUG Software Catalog Update #1

Updated catalog that lists software releases for years 1983-1985.

KEYMAP CP/M-80

KEYMAP lets you designate the responses produced by your computer's function keys. It works like the KEY command in GW-BASIC except that more keys can be defined. In turn, those keys become part of the "system" and can be used with any program or operating system. For example, you can redefine key F1 to produce the command "DIR A:" (including the return!) so that a single keystroke lists the files on the disk in drive A. Requires CP/M.

MBASIC Payroll

Payroll system for small business. Handles 30 employees with single drive system, or 100 with dual drives. Calculates deductions using Fed Tax tables, FICA tax rate, state and city tax rates by percentage. Requires 64K, printer, one drive (two recommended), and MBASIC ver 5.21· or higher.

NavProgSeven

Database management system for pilots who fly cross-country frequently. Built around a latitude/longitude referenced navigation program designed to prepare flight logs that are ready for cockpit use. Stores aircraft performance data, nav data for each checkpoint, airport or navaid. Requires H-19/H-89, 64K RAM, two drives, printer, and MBASIC 5.21 or higher.

Sea Battle

A video action game that uses the graphics features of the H-19 terminal. The player commands a destroyer force in opposition to an aircraft carrier and escorts, all of which attack the players ship. Requires 32K RAM on H8 and H/Z-89/90 (hard-sectored) and H-19 terminal, or H/Z-100 (soft-sectored).

Utilities By PS

A collection of CP/M utilities written by Pat Swayne, the HUG software guru. Includes directory listing, printer control, and disk drive testing routines. Requires 32K RAM for H8; H/Z-89/90.

Utilities

A group of CP/M utilities that includes programs for both H8-H/Z-89 and H/Z-100. Includes SEE, SCR89, UNSCR, SCRNCLK, and more. SEE is a text file listing routine, SCR is a screen dump utility for text files only, SCRNCLK is a real-time clock that can be displayed on screen.

X-Reference Utility for MBASIC

Contains two programs that will provide the Microsoft BASIC programmer with tools to aid in program development: VARIABLE cross reference utility and BRANCH statement cross reference utility. Requires 32K RAM. Tested only on MBASIC 5.2 and above.

ZTERM

A modem package for the H8, H/Z-89/90, and H/Z-100 that lets computers talk to each other. A modification of the CP/M TERM package for the H8 and H/Z-89 series. Requires 32K RAM; H8 must have 4-port serial card (H8-4) or WH8-47 board. ✻

Getting the Most From Your Computer

Part 2

John P. Lewis
6 Sexton Cove Road
Key Largo, FL 33037

In Part 1 of this series, we looked at a way in which the computer user could enhance the usefulness of his/her computer through learning to program in Turbo Pascal. That might sound like an ambitious project, but as you will soon see, it is not nearly as forbidding as you might imagine. I'm going to make the learning process as painless as possible by presenting code examples that perform some useful tasks while providing a vehicle for an exercise in logic.

For those of you who enjoy a challenge, this is a good place to "get your feet wet". I'm sure you will find, as I have, that the deeper you get into programming, the more enjoyment you will derive from it. With some individuals it becomes almost a passion. It is not necessary to conjure up some code examples that would baffle most of us in order to create programs that you will find useful. The code included in this article is certainly a good example of the above.

Before we get started with the new code, we will perform a little surgery on that presented in Part 1. Actually, we will transform it from a program into a Pascal unit, providing a means of enhancing portability and modularization. Once a code segment, or even an entire program has been changed into a unit, it can be accessed by another program by including it in the "uses" clause following the program name in most Turbo Pascal programs. The conversion process is quite simple and logical, so put on your programmer's hat, we're going to embark on a little "Turbo Tour".

A Turbo Pascal unit has two sections or divisions, the interface section and the implementation portion. The interface provides a means of making "public" those portions of the program which must be available to calling programs. In other words, if one of the procedures in the unit

must be accessed by the program in which this module resides, it must be listed (declared) in the interface portion of the unit. If it is to be private (inaccessible), then it must not be declared. Variables are treated in the same manner, declaring a variable in the interface portion of a unit makes the variable global in nature. Moving the same variable to the implementation section of the unit makes it "private", inaccessible to the rest of the parent program.

The implementation section of a Pascal unit is self-descriptive. It contains the code that carries out (implements) the various tasks provided by the module. Just below the implementation statement the "uses" clause lists the various units used by the unit under discussion. Yes, some units depend on others for the definition of functions contained therein. For instance: nearly any program or unit that utilizes cursor control in its screen output will need to access the gotoxy(col, row); procedure contained in the crt unit. Next, the procedures and functions that perform the work are defined within the body of the implementation portion of the unit. Last, if the unit contains code that must be initialized before being called, a begin statement similar to that contained in a regular program will cause the execution of any code required for initialization.

Let's make an example of the code contained in Part 1. We will construct a unit that can be included in future programs for window creation. Never fear, we won't be getting involved in writing the code for true windows that pop up and disappear, just a somewhat less sophisticated emulation of the real thing. If you would like to get involved in writing code for true pop-up and pull-down windows, stick around, we are going to get deeper as we go.

Examine the listing for "Boxunit" very

carefully, it is quite similar to the listing contained in Part 1 of this series, "DRAWBOX.PAS". If you have copied this code and it now resides on your disk, then your job is going to be quite easy. Make a COPY of it under another name (BOXUNIT.PAS), and then work on the new version, leaving the old code for reference and a possible back up in case of "pilot error". Perhaps the easiest way to accomplish this would be, while "DRAWBOX.PAS" is loaded into the Turbo compiler, to invoke "Files" (Alt F), choose "Write to" from the pull-down menu, and then supply "BOXUNIT" for the new name when asked by the compiler.

Now you can work on the new version with relative impunity. Modify the code so that it appears just as it does in this month's listing. Compile this latest version and you will notice instead of producing a file with an "exe" extension, the compiler will generate a file named "Boxunit.tpu". Success!

Notice that I have left out some of the code from Part 1 in the new version. This was done in an effort to make the unit as generic as possible. Adding code to enhance the appearance of the box will, at the same time, limit its usefulness in other applications.

Before proceeding, let's look at one of Pascal's "loop control" features used extensively in the above example. The "for" statement uses an integer (i) for control. The "do" must be followed by "begin" in the case of a compound statement which, of course, must be terminated by an "end". If only one statement is to be executed by the "for" loop, then no "begin" is necessary. You will find examples of both applications in "BOXUNIT.PAS".

Another point of interest is the use of untyped constants for the actual parts of the box. An untyped constant retains its value throughout the program, resisting

BOXUNIT.PAS

```
Unit Boxunit;
interface
Procedure Drawbox(Uplftx, Uplfty, Lowrtx, Lowrty:integer);

   const                      { Alter the following values to move the menu box }
                              { around on the screen }
     Top     = 4;             { location of top line }
     Left    = 5;             { location of left column }
     Right   = 75;            { location of right column }
     Bottom  = 22;            { location of bottom line }

implementation
uses crt, dos;

Procedure drawbox(UplftX,UplftY,LowrtX,lowrtY:integer);
   Var i : Integer;

   const
     TopLft = 201;            { These characters may be changed to alter the }
     Hor    = 205;            { appearance of the box }
     Vert   = 186;
     BtLft  = 200;
     TopRt  = 187;
     BotRt  = 188;

   begin
    clrscr;
    gotoxy(UplftX,UplftY);write(chr(TopLft));        { draw top left corner }
    for i:= (uplftX+1) to (LowrtX-1) do  { top row }
    begin
      write(chr(Hor));                    { draw the top horizontal char }
    end;
    write(chr(TopRt));                     { draw top right corner }
    for i := (UplftY+1) to (lowrtY-1) do
    begin
      gotoXY(UplftX,i);write(chr(Vert));   { draw both vertical columns }
      gotoXY(LowrtX,i);write(chr(Vert));
    end;
    gotoXY(UplftX,LowrtY);write(chr(BtLft));          { draw bottom left corner }
    for i:= (UplftX+1) to (LowrtX-1) do
    write(chr(Hor));write(chr(BotRt));     { draw bottom horizontal char }
                                           { followed by the bottom right char }
    end;
   end.
```

any effort to change it, unlike typed constants which can take the form: Hor : Integer = 205;

In Part 1 of this series, I promised you a computer calculator, one with the ability to "back up" (a desirable feature for those of us with ten thumbs). Another feature not mentioned, but one which will become increasingly obvious, is the fact that you will be able to customize this calculator to your individual needs. It comes with only four functions, but can be easily modified to handle square roots and numbers to a power or more sophisticated uses if you so desire. I encourage you to experiment with it and modify it to satisfy your curiosity, but modify a copy, not the original. You can have as many copies of the same source code as you want in the same directory, as long as they carry unique names. Of course, you can create different directories for each copy and retain the same name, but that is an activity that deserves a lot of thought towards careful organizing. I'm assuming you have a hard disk. If not, then the task of keeping everything where it is easy to find becomes quite different. In any event, this is an entirely different field of expertise. If you wish to learn more about MS-DOS, then I would encourage you to read William Adney's articles in this magazine. I have learned much from him, particularly in the area of memory management.

Alright, let's look at the code for our calculator, CALC.PAS.

You will notice that we have already put "BOXUNIT" to work since it is the first file named in the "uses" statement of "CALC.PAS". This is an example of how really useful, powerful programs can be built a step at a time and Turbo Pascal is an ideal tool for construction.

"CALC" calls on "BOXUNIT" twice, once to draw the menu box and again to draw a box for the calculator to fit within. We use two procedures to provide a multi-color screen to enhance the appearance of our utility, "Esthetics" and "Setup" are called, first to provide a light gray background with a black box within it, and then setup is called to provide a blue background with yellow letters for the actual calculator. If you are using a monochrome monitor, then substitute black for the background color and white for the text color. You can just delete the code relative to the color commands if you want, but that would necessitate adding them at a later date if the program is given to a friend or if you acquire a color monitor. "Setup" first creates a window within our box, thereby preventing any accidental calamities from destroying our frame. The *window(col, row, col, row)* command creates a logical window on the screen and changes all cursor position references to a relative position within the new logical "window". For instance, if we use the command *window(20,10,40,20);* we create a logical window on the screen that starts at column 20, row 10 and ends at column 40, row 20. If we then give the command: *gotoxy(1,1);* the cursor will appear at absolute screen coordinates col-

CALC.PAS

```
Program Calc;
uses boxunit, crt;

Var
    accum, val1  : Real;
    row, col, i  : Byte;
    Option, Len  : Integer;
    ch, chc, op  : Char;
    s, st : string;

Function Get_result(s : String;p : char):Real;
    Var v : real;            { provide real storage for value of s }
    code : Integer;
    begin
    Val(s, v, code);         { get numeric value of string s, store in v }
    case p of
        '+':begin accum:=Val1+V;end;
        '-':begin accum:=Val1-V;end;        { compute result, store in accum }
        '/':begin accum:=Val1/V;end;
        '*':begin accum:=Val1*V;end;
        else accum:=V;
        end;
    Val1:=Accum;             { store accum in val1 }
    Get_result:=Val1;        { Function must be defined }
    end;

Procedure docalc;
    begin
        col:=2;row:=14;st:='';
        Accum:=0;ch:=chr(0);Val1:=0;      { initialize variables }
        gotoxy(col,row);
        while ch <> 'q' do                { provide means of escape }
          begin
            ch:=readkey;
```

```pascal
             if ch <> chr(8) then st:=st+ch;   { enable backspace }
             if ch = chr(8) then
                begin
                  if col > 2 then
                     begin
                     col:=col-1;gotoxy(col,row);   { update display after backspace }
                     write(' ');
                     end;
                  len:=length(st);
                  st:=copy(st,1,len-1);          { remove last char }
                end;
             write(ch);
             if ch <> #8 then
             col:=col+1;
             gotoxy(col,row);
             if ch = '=' then
                begin
                  s:='';op:=chr(0);
                  for i:=1 to length(st) do    { parse the input string }
                     begin
                       chc:=st[i];
                       case chc of
                       '0'..'9':begin s:=s+chc;end;
                       '.'     :begin s:=s+chc;end;
                       '*'..'+':begin st:=s;accum:=get_result(st,op);
                                s:='';op:=chc;end;
                       '-'     :begin st:=s;accum:=get_result(st,op);
                                s:='';op:=chc;end;
                       '/'     :begin st:=s;accum:=get_result(st,op);
                                s:='';op:=chc;end;
                       '='     :begin st:=s;accum:=get_result(st,op);
                                s:='';op:=chr(0);end;
                       end;
                     end;
                  gotoxy(col,row);write(' ',accum:4:4);{ format & display result }
                  gotoxy(10,row+2);
                  write('Enter "q" to quit, any other key to continue ');
                  ch:=readkey;if ch <> 'q' then col:=2;gotoxy(col,row);clreol;
gotoxy(col,row+2);clreol;accum:=0;val1:=0;st:='';gotoxy(col,row);
                end;
             end;
          end;

Procedure Esthetics;
   begin
      textcolor(black);textbackground(lightgray);
   end;

Procedure SetUp;
   Begin
      window(Left+1,Top+1,Right-1,Bottom-1);
      TextBackground(Blue);TextColor(Yellow);clrscr;
   end;

begin
   Esthetics;clrscr;
   drawbox(left, top, right, bottom);   { top=4, left=5, right=75, bottom=22 }
   setup;
   gotoxy(11,2);write('Please enter...');
   gotoxy(20,4);write('1. Calculator');
   gotoxy(20,6);write('2. Function not yet implemented.');
   gotoxy(20,8);write('3. Function not yet implemented.');
   gotoxy(20,10);write('4. Function not yet implemented.');
   gotoxy(20,12);write('5. Function not yet implemented.');
   gotoxy(20,14);write('6. Exit to operating system');
   gotoxy(18,16);write('the number of your choice ');
   readln(option);window(1,1,80,25);clrscr;
   case option of
   1:begin Esthetics;clrscr;
         drawbox(left, top, right, bottom);gotoxy(35,top);
         write(' Calculator ');setup;gotoxy(2,2);
         write('This calculator provides four functions using the following');
         gotoxy(2,4);
         write('keys: + (addition), - (subtraction), * (multiplication)');
         gotoxy(2,6);
         write(' / (division).');
         gotoxy(2,8);
         write('Problems should be entered in a number, operator, number format.');
         gotoxy(2,10);
         write('Pressing the equals key ( = ) will cause display of the answer.');
         gotoxy(2,12);write('Enter your problem: ');
         docalc;
      end;
   end;
end.                          CALC.PAS (Cont'd.)
```

umn 21, row 11. Any attempt to write past the boundaries of the logical window will cause the text to "wrap".

Our next chore is to create a menu for the user. Since we are going to add to this basic menu screen, we will go ahead and plan its final (?) appearance now. Notice the plethora of cursor location commands followed by write commands within the program main body. Here we are attempting to build a nice looking screen and Turbo Pascal is definitely up to the job, making the task relatively easy and providing almost instantaneous screen updating. A "case option of" statement is provided below the block of menu content statements which enables multiple choices of the options provided. Only one is implemented at this time for obvious reasons. Using the Pascal "case" statement provides an easy, compact method of screening the user's input for any one of many possibilities.

Instructions for the use of our calculator are printed within the screen window while leaving room near the bottom for problem entry. Only one job remains, that of providing the logic for problem entry and the computation/display of answers.

The procedure, "docalc" is the "work horse" of our project calculator. First, it initializes the screen column and row variables, as well as the variables used in the math calculations. Notice that we retrieve a character at a time employing *ch:=readkey*. "Ch" is then compared to a backspace (*chr(8)*) which, if present, causes the string to be truncated one character through the use of Pascal's "copy" function. If ch is not equal to a backspace, the character is appended to our input string (*st:=st+ch*) and the cursor is advanced one column. Each input character is then compared to the "=" operator. If the equals character is detected (signaling the end of character input), another case statement (*case chc of*) is called to parse and process the input string.

Using this methodology provides a means of filtering the character input for numeric or operator entry, constructing strings from numeric entry while re-initializing "s" (the string variable) for each operator input. Each time an operator is detected, function *Get_result(s : String;p : char)* is summoned to evaluate the expression up to the current position. Note that the current operator is updated (*op:=chc*) AFTER exiting function *get_result*, thus enabling the number, operator, number format.

Before any math can be performed, the input string(s) must be converted to a numeric variable. Turbo Pascal's "Val" procedure is used to evaluate each individual string for its numeric value which is then used within the body of another case statement (*case p of*) to arrive at a running total to be stored in "accum".

I hope you agree, after you have copied the program listing and eliminated the inevitable typos, that the logic and resulting code for this program are easily understood. A large part of this phenomenon is due to the self-documenting nature of Pascal. I have found that I can get a program "up and running" much faster if I code it in Pascal, than if I wrote the same program in "C". Also, the program is much easier to understand when I go back to it at a later date.

Maybe you're thinking that you could get a bite of lunch while your new computer calculator processes the latest keyboard entry consisting of an entire window length chain of numbers and operators. Don't believe it, the answer will appear in the window before you can get your finger away from the equals key. Turbo Pascal is FAST.

O.K., now you have the source code for a calculator which can be incorporated in any of your future programming endeavors. Next issue we are going to add another item to the menu for our programming project and show how to convert CALC.PAS to a Turbo Pascal unit for even greater versatility.

If you have any questions regarding this, or any of the articles which comprise this series, or even Turbo Pascal programming in general, please feel free to write me at the address given at the beginning of this article. Be sure to enclose an S.A.S.E. (Self-addressed, stamped-envelope) if you wish a reply. Feel free to make comments, pro or con, about the series, but if you have any criticism, please make it constructive. If you stipulate that your letter may be included in a future article, that would be helpful. ✽

```
A1:  (F,2)  45                                                    READY

   A B C D E F G H I J K L M N O P Q R S T U V W X Y Z AAABACADAEAFAGAHAIAJAKAL
 1# # +
 2# # +
 3# # +
 4# # +
```
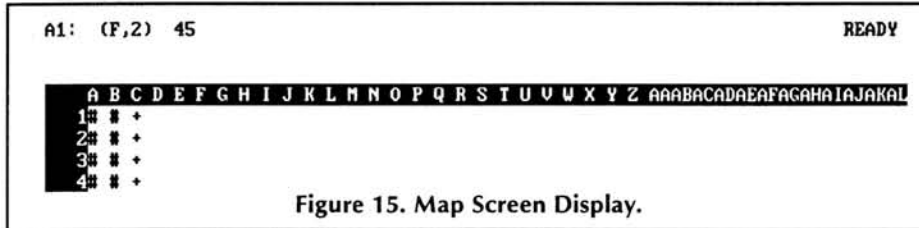
Figure 15. Map Screen Display.

mits you to enter any of the special characters directly in the spreadsheet cells. To insert these characters, press the Alt key down and key in the ASCII number using the keypad, not the numbers across the top of the keyboard. All of these characters are shown in Appendix H of the ENABLE Spreadsheet/Graphics manual.

ENABLE OA has improved the Audit functions. If you select the Audit function from the Worksheet menu, you have three options to select from, LIST, HIGHLIGHT, and MAP.

The first option is List. If you select this option, you can select Status, Formulas, or Range-Names. Selecting Status will show you the current values of Global setting. See Figure 13 for the display.

The last option, MAP, is unique in that it will show you in one display the makeup of all cells. The display uses " for labels, # for values and + for formulas. The cells are shown only two units wide so a lot of cells are shown on the screen.

When you are finished with the spreadsheet, press F10 S(ave). ENABLE will respond with the file name. Pressing <Enter> will cause ENABLE to prompt "Cancel" or "Replace" if the file exists. This gives one additional chance before over writing a current file. Like the word processing files, the spreadsheet files can be PASSWORD protected. The procedure is the same as outlined in the word processing article.

This completes an overview of the ENABLE OA spreadsheet functions. I don't expect you to run out and buy ENABLE, but I hope that this short series will make you want to look at the capabilities of an integrated software package. ENABLE is very powerful and full-featured. Each release provides more capability to the user.

For the next two articles in this series, see the February 1991 issue for ENABLE Revisited — Part 4, and the March 1991 issue for ENABLE Revisited — Part 5. We're sorry for any inconvenience this may have caused. ✽
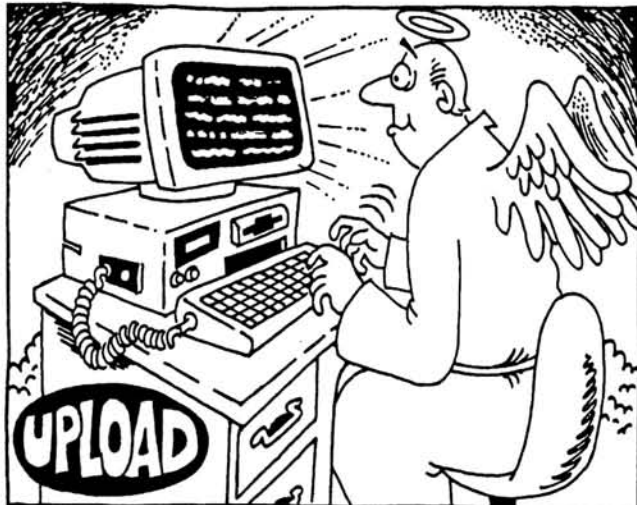
Fear of Frying

Nathan E. Baker
Technical Writer
Zenith Data Systems

Since my last article (Real World Computing, Oct. 1989), in which I taunted the virtues of the MinisPort (floppy disk drive version), I've upgraded my portable computing with a MinisPort-HD, which includes a 20M hard disk drive and 2M of RAM. Someday I'll step into the real world and blow a couple thousand dollars on an 80286 or 80386SX portable, but until then...

When I bought the hard drive version of the MinisPort, the only reservation I had was with the dependability of the hard disk drive. Those of you who have been in the computer industry for a while might remember problems that early hard disk drives had when they were implemented in a portable computer design. I do. I'm happy to report that those problems have disappeared.

I've always considered ZDS products to be very reliable, but when I put the MinisPort-HD through a typical camping trip lately, I discovered exactly how sturdy they are.

The Minisport-HD is not a toy-store computer, or a cheap down-sized XT-clone. Anything that can put up with the amount of abuse I've heaped upon this computer deserves some respect.

Nate's Vacation
(or, Portable Computer Goes to Nifelhiem)

As in my last literary foray, I took my portable computer camping last week. I usually take a tent, and I do a lot of fishing and hiking, usually alone (which I prefer, in most cases). This time, as before, I took a number of batteries and two charging units with me, so I had the equivalent of about twenty hours of constant on-time at my disposal. (Why anyone besides myself would torture themselves like this, I don't know, but be aware that I don't consider my personal writing as "work" — and hey, keep an eye out for "Fishhooks for Franklin", due in better bookstores in the next year or so.)

Arriving at one of my favorite back woods campground (which is within easy driving distance of essential services, such as taverns and party stores), I set up my tent in a nice location, which was easy, since there was only one other occupied camp site. The old couple there kept a close eye on me (read that, "stared a lot") and my tent, so I wasn't too worried about security. Plugging my battery chargers into the nearest power tap, I left two batteries behind to charge while I took off to go fishing.

Last year I managed to dunk my portable into a river. It came out of the depths wetter and dead, but after I disassembled it, cleaned it, and dried it thoroughly, the unit powered-up with no problems. The floppy disk drive even worked, though the 2-inch floppy disk that was in it was trashed. This time I'd learned from my mistake, and I avoided dropping it in the river. I dropped it on the dock instead.

With about 12M of text (about 11.5M backed up before I left on vacation) on it, I thought I was done for. With my heart in my throat — and while I was away from anyone who could hear me scream — I turned it on. Everything worked, though I wouldn't recommend this type of drop test, or the dunk test, to be performed by the unqualified (Ha!).

The days progressed without anything else bad happening to my hardware. Then the weekend arrived, and with it, new neighbors. Noisy neighbors with dogs and — oh no! — motor homes with children. I'm pretty much immune to these irritations, since I'm gone for most of the day and I usually don't get back to my tent until the A.M. hours (a nearby tavern is essential), but strange things happened that weekend.

A little boy ran through my camp on Saturday morning, and I learned to hate the spring-activated latch on my AC adapter. (You know, the black input cord that latches into the battery pack with a death grip, unless you pull the plastic shroud back?) The whole computer slam-danced across my camp site and tried to follow the kid, who'd somehow wrapped one of his legs around the power cord. He stopped a few yards away and started crying. I don't know why he was crying. I really wouldn't have killed him. Sure, the fillet knife was in my hands, but hey, that's just circumstantial evidence. Still, his parents made sure he stayed away from my camp sight.

As bad as that accident sounds, it really wasn't as bad as dropping the thing on the dock. The computer still worked, though it was quite dusty and the keyboard was full of pine needles. Good thing I learned to carry some computer repair tools (itty bitty phillips screwdriver, pliers, and a rock).

Saturday seemed against me. After the incident of the rambling kid, I decided the best thing to do was to go fishing (and escape from possible long distance retaliation from someone's parents — hey, it's not my fault that their motor home's power plugs kept falling out of the tap).

I returned to the campground around sunset, not because I had to get anything, but because I wanted to check my camp site. Too many people were showing up, and the old couple that stared at me all week (sorry — watched me all week) were gone.

I knew something was amiss when a mangy beagle-like dog high-tailed it from my camp site when I showed up. A suspicious wet mark graced the side of my tent.

With all the trees around, I didn't know why this brain damaged mutt had to pick my tent, but needless to say, I was not a happy camper. Not wanting to make any more enemies that I didn't have to make, I decided to let it go, and after checking the tent to make sure everything was alright (if somewhat besmirched), I left.

Later that night, it rained. No, rained is the wrong word. Deluge? Flood from the sky? The wrath of Poseidon visited upon the land? Whatever. By the time I returned to my tent (which I'd wisely pitched above the water table), I went directly to sleep. Sometime later, I was struck by the call of nature.

Fortunately, I'd picked a sight close to the restrooms (another essential). Slogging through the mud and the new maze of support ropes, tent stakes, and dangerous looking electrical cords, I was successful with my quest. On my return trip, I noticed something. My battery chargers were still plugged in!

Yow!

The ventilation holes gaped skyward like rectangular funnels. Hmmmm, I thought, this could be a problem.

By now, I was pretty wet. The campground was wet. Everything was wet. The only thing I could think of using was a branch to pull out the plugs. Not wanting to fry, I looked around for something better... Where's someone else's kid when you need them? Then I thought of my fishing pole. A little while later, I'd managed to successfully land two electrified chargers — with batteries attached! They did flop around a bit, but hey — that motor home stopped 'em. Soaked, but not electrocuted, I went back to my tent.

In the morning, I cleaned my catch (pulled the circuit boards out) and hung them out to dry. It didn't rain and the sun was bright and warm, and in a few hours they looked pretty dry, so I decided to try one, knowing it probably wouldn't work. After all, it wasn't rated for outdoor use, and I'm sure it matched or exceeded its maximum operating humidity.

Digging a discreet hole and covering it with my tackle box, I plugged it in. It worked! The other charger worked, too. I'm still using them today (though they look a bit battered — hey, they put up quite a fight! Good thing that motor home was in the way...)

There was one other abuse I could have put the chargers through, and I almost did, but I couldn't figure out a way to fool the beagle-mutt into thinking my battery charger was my tent. Well, there's always next year...

The Moral of This Story

I hope my experience is of some use to you, if not just for amusement. The moral is, of course, that ZDS portables can "take a licking, and keep on ticking."

Happy porta-puting! (Hey! That's not your frisbee in my campfire — I always cook like that!) ✳

Port-O-Call

COM1

Laura White
759 Polfus
Benton Harbor, MI 49022

Your 9600 Baud Modem:
Getting the Most Out of It

One of the latest advances in technology is the 9600 baud modem. However, as with anything new, there are a few adjustments that might need to be made in order to get the most out of your new, faster modem.

On COM1, one of our subscribers was inquiring about switching to a dual standard 9600 baud modem. He called in and questioned how he could change from a 16450 chip to 16550 chip in order to get the speed necessary for his new modem. It turns out that he received a complete course on the speed of modems and how it affects the computer.

The first piece of information he received was that a 9600 baud modem is not that fast and that the National Semi-Conductor series of UART chips, back to the original 8250, has always been fast.

He was then informed of some tests that were run comparing the 16450 and the 16550 chip.

Apparently the 16550 is a direct replacement for the 16450/8250B. The 550 is not necessarily any "faster", but has the FIFO mode which interrupts the computer one time for every 16 bytes of data either transmitted or received. The 450 generates an interrupt every time a byte is either received or sent.

The computer's software needs to be aware of the 16550 and you need to know how to program it before receiving any benefit. If you are getting a V.42 bit comparable modem, the actual data rate could top out at 36 kilobytes per second which *could* put a strain on the machine. There is

an OOP Communication source code library for Turbo Pascal 5.5/6 called Object.Com.

As far as testing goes, the original personal computer, with the 8250B UART, maxed out at 36 kbps. The same machine with a 550 UART could receive 115 kbps with no errors as long as you were doing something simple like writing to the screen. Incoming data was accurate without losing anything at this speed.

Our caller continued with his advice by stating that if you have a 286/386 class machine with a 16450 UART, it should be adequate enough. The only instance where the 550 chip might prove to be more useful is if you were running a multi-line BBS or something of that nature.

After reading the advice our original caller received, it can be interpreted that even though technology has supplied us with a 9600 baud modem, there does not seem to be a wide variety of applications. It could also be interpreted that the more powerful computers seem to have what it takes to get "enough" out of the 9600 baud modem. There is, however, another side to the scenario.

One zuggie called in and offered another bit of advice which completely negates everything we have already been told. But remember, our callers up until now have been offering advice on how to get "enough" power from the fast modem.

Our final caller disagrees with what everyone had discussed. He in turn saw a very real need for the 16550 AFN buffered UART on 8 and 12 MHz machines even without multi-tasking.

He uses a US Robotics HST and dual

standard modem with the ports locked at 38,400 baud. Without the 16550 chip, he had constant problems with extraneous and missing characters. Apparently the characters would appear during a BBS session, especially when he touched a key during a screen update.

After spending some time trying to figure out what the problem could have been, our final caller tracked it down to his 16450 UARTS. By installing a 16550, his problem was solved. He also informed our original caller that he had better beware of the Western Digital chip. According to him, this chip has problems operating at 2400 baud and below.

For those of you looking into a 9600 baud modem, there are many considerations which need to be looked at. Do you really need the extra power especially when there does not seem to be a great number of application possibilities. Second, if you do feel you need the most powerful modem on the market, be aware that it is new and there are still some bugs that need to be worked out.

I am aware of others who are having problems with their 9600 baud modem with simple receiving and transmitting. As a matter of fact I know of a case where a zuggie had one and decided he was having too many problems that were attributed to the modem. He in turn opted against it and went back to his 2400 baud modem.

It would appear from other zuggies that they are not completely "sold" on the 9600 baud modem either for the same reasons. The choice is obviously up to you. Good luck and keep fellow zuggies informed of your findings by dialing COM1.

This month in Port-O-Call: COM1 we are going to look at some ways to append your files by using a series of simple commands to detect a virus, protect against installation programs tampering with your AUTOEXEC.BAT file and, using the ATTRIB command to protect your files by giving them the READ ONLY attribute.

Detecting a Virus

A virus can show up anywhere at anytime causing your system to do any number of unexpected things — hide files, misread characters, even completely take the system down. There are, however, ways of detecting a virus.

You can acquire programs, either commercially or through shareware, which are designed to detect any changes in files. There is, however, a virtually free method of detecting the presence of something that has changed even a single bit in any file.

All PC/blue (DOS) and SIG/M (CP/M) software disks are distributed with cyclic redundancy check utilities (CRC). These were originally intended to detect disk and file copying flaws.

There is a file of these disks named CATALOG.NNN. This file contains an ASCII list of files and the disk along with the original CRC utility. Apparently when the user logs onto that disk and runs the CRC utility, it will recompute the CRC for each file listed in the CATALOG.NNN file. It will then compare it against the original file, and a single bit change anywhere in the file will result in a different CRC that will be reported. CRC67.ZIP is available on the COM1.BBS and can be used with any version of DOS.

The CRC can be used with your batch file called AUTOEXEC.BAT to check the files in any directory. All that is needed is a known benchmark list of CRC's which the CRC can create itself. An example used on the bulletin board was as follows:

Suppose you want to check your root and bin directories every time you boot the system. Assuming the CRC.EXE is in your path, do this:

```
C:\> CRC *.* f (build a crcklist.CRC file
            in the root directory...)
```

Done.

```
C:\>cd \bin
C:\>bin.crc*.* f  (build another in the
            bin subdirectory...)
```

Now use your editor to build CRCBOOT.BAT file:

```
CRC [>>\CRCBOOT.LOG]
cd \bin
CRC [>>\CRCBOOT.LOG]
cd \
```

Then in the AUTOEXEC.BAT file add this line:

```
CALL CRCBOOT
```

Note: The optional creation of a log file records the results. Without this log, the results will simply appear on the screen and scroll off into oblivion.

You can modify the CRCBOOT.BAT to check any directories you wish. Each must contain a CRCKLIST.CRC file previously created with "CRC *.* F".

Protecting Files
Using the ATTRIB Command

Looking at the files on your hard disk, there are some that change regularly and those that do not. For example, your budget from last year will not change, and most application programs such as MS WORD will not change either.

Because you might want to prevent changes happening to some of these files, you might want to consider the advantages of the ATTRIB command when it comes to protecting them against unwanted changes.

In order to utilize the ATTRIB command, you must first decide if a file is read-only. To view the status of the files in a directory do the following:

```
C:\WORD>attrib *.*
```

DOS will display a listing similar to this:

```
A   R  C:\WORD\PROGRAM4-3.doc
A      C:\WORD\BUDGET2-90.DOC
```

The letter "R" in front of the file tells you that its read-only attribute is turned on and cannot be deleted or changed.

To make a file read-only in order to protect it from changes, issue the following commmand:

```
attrib +r FILENAME
```

where FILENAME is the name of the file you want to protect.

Once you have used the ATTRIB command to turn on a file's read-only status, any attempts to delete or erase that file will deny you access.

You can also turn off the ATTRIB command by typing the foloving command:

```
C:\>attrib -r FILENAME
```

The ATTRIB command is not the perfect fool-proof method of saving your files from changes. It does however, help protect your files from accidental changes. (And we all know how often that can happen!)

Now on to protecting your AUTOEXEC.BAT file.

Protecting Your AUTOEXEC.BAT File

Actually this section is designed to let you in on the hows and whys regarding protecting your file from installation programs. If you have ever noticed, your AUTOEXEC.BAT file is constantly experiencing changes whenever you install a new program.

Before we actually get into the protection of your file, I thought it might be helpful for you to know what installation programs do. They usually make two types of changes: a) insert a directory into the PATH statement or, b) append new commands to programs that load and stay in memory until you reboot.

How do these changes affect the performance of your machine? By placing an infrequently used directory in your PATH statement — especially at the beginning, your machine is not running as efficiently as possible. You will want to place that new directory at the end of the list.

In addition, since most installation programs have a tendency to take whatever space is available, this can cause chaos in your memory-resident programs such as mouse drivers. To avoid this, you need to change the order in which these lines appear, or remove them altogether.

The way to deal with this struggle for space is quick and easy. First, place the startup commands you would normally place in AUTOEXEC.BAT in another batch file named BEGIN.BAT. Then create an AUTOEXEC.BAT file that contains only the following two commands:
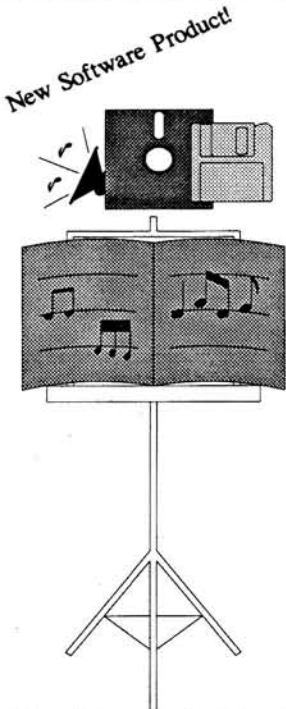
```
@echo off
c:\begin.bat
```

By taking this simple step, you can go back and examine the changes that were made by an installation program and make the changes you deem necessary.

Conclusion

Protecting your files is a very important factor when it comes to your computer. Here are a few things you can do to keep your system running efficiently and safely. Many thanks to Ray Brant, Craig Stevenson, Pat Swayne and Rick Terrill; and Robert Savage for his information on virus detecting. As usual, there would be no Port-O-Call without COM1 users. Until next time, thanks for visiting our Port-O-Call: COM1. ✻

Introduction to C++
Eighth Installment

Lynwood H. Wilson
2160 James Canyon
Boulder, CO 80302

I've gotten a fair start on my Artificial Life project this past month. Thus far, it consists of tiny creatures which move around a screen with food, obstacles, and mostly empty space. When one of them runs out of fuel he dies, when one of them eats a food character he gets more fuel, and if his fuel level rises high enough, he reproduces and becomes two. Next step will be a genetic algorithm so they can evolve more complex behavior. Soon to come, bisexual reproduction. We're all looking forward to that one.

This program is working out very nicely in C++. The Object Oriented paradigm seems to lend itself quite well to exploratory programming. What I think I mean by that is that you can reconsider a decision and change something fundamental without sending a swarm of bugs and changes propagating throughout the system. You can try something without the investment, without committing yourself to a lot of work if you change your mind and want to try something else. I like it a lot.

It has been said that LISP, the main Artificial Intelligence language, has been optimized over the years for use by programmers who never quite know what they are doing and that is why it is so easy to make major changes to a LISP program. This is not meant to slight the AI community, but rather to say that they are working at the forward edge where all sorts of things are not generally known in advance.

This resulted in an extremely flexible language, which allows you to change things and try different approaches with little penalty.

It seems to me that C++ offers the same kind of freedom, the same flexibility. The more I use it, the better I like it. I would hate to get a contract which had to be done in plain C.

C, in part, grew out of a lot of new ideas about how programmers should write programs. C++ is more in line with the way programmers really work, rather than fulfilling some theoretical requirement.

Resources

I'd read somewhere that Bruce Eckel was co-author of a new book on OOP called *The Tao of Objects*, and I've been looking forward to it. When I finally got a copy, it turned out that Mr. Eckel had been too busy and left the book (except for a long and good introduction) to his co-author, Gary Entsminger.

The Tao of Objects has examples in both C++ and the new Turbo Pascal with explanations of the differences between the two languages. Few programmers are going to want both, so why not publish two versions of the book as others have done? Much of the code is in pretty small fragments which makes it harder to understand, particularly if you are just learning. Some of the examples are complete programs, which I found more useful. Some of the examples don't work, and some of the explanations seem to go slightly awry.

It's not a bad book, but I think it would be better if Mr. Eckel had had more time. I still think his book *Using C++* is one of the very best in spite of being based on Version 1.

I have a new contract to develop the user interface for an expert system and, along with the job came a copy of the Zinc interface library for C++. It does very fancy windows in either text or graphics mode.. So far I can only report that the books are hard to read and their demo program has a few bugs in it. The main problem with the documentation is insufficient examples. Long descriptions, a few code fragments, very few complete examples. This is true of almost all software manuals, in spite of long, loud complaints from users and reviewers. More on Zinc as I learn to use it.

C++ Techniques and Applications by Scott Robert Ladd is a good book. It's for C programmers learning C++, and it's full of good information on the differences between the languages and the ways they are best used. Lots of sample code in complete programs, and a disk. I recommend it, in spite of too many typos.

I have another new book which I will report on if I have time to read it before I finish this.

Pointers

Pointers are variables which hold the addresses of data instead of the data itself. This may not sound like much if you have not encountered it before, but it is a very powerful concept. Pointers are used frequently in C and C++ both to do things which cannot be done any other way and to produce more compact and readable code.

Pointers can also be used to produce unreadable code. As always, with power must come responsibility.

Memory

The memory in your computer is organized as a linear series of bytes, each of which has its own address. When your program defines a variable with a statement like char ch; the program finds some unused memory and allocates one byte of it as a home for the variable ch. From that point forward whenever ch is referred to the program goes to that place in memory, that same byte. If you execute this statement: ch = 'Q'; the program goes to the memory location which holds x and puts the number 7 (the ASCII value which represents 'Q') in that byte.

If the variable ch no longer exists because the block within which it was declared ends, its memory is freed and can be reallocated for another variable.

Different data types need different amounts of memory. A character variable uses only one byte, an int uses two bytes, a float uses four, and so forth. These numbers also can vary from one machine

to another.

Addresses

Each byte of memory has a unique address. In the IBM PC and its relatives, the memory addresses are numbers between 0 and 655,360. Each of these addresses can be thought of as a pointer to a single byte, but until variables are created and memory allocated for them by a program, they are not pointers to variables.

When a program creates a variable and allocates memory for it, the address of that memory is stored by the program as the address of the variable. In the case of data types which require more than one byte (all types but character), the address of the variable is the address of the first of its bytes.

Definition

A pointer is defined almost like other kinds of variables. The difference is that the compiler needs to know what kind of data it will point to. To do that, we need to use a new operator. It is called the indirection operator and is represented by the asterisk '*'. This is the same symbol that is used for multiplication, but the usages are different enough that there is no confusion. I think the best way to read the indirection symbol is "That which is pointed to by", although it is a bit windy.

Here is a pointer definition: `int *some_ptr;`. Recall that when you define an ordinary variable, you first tell the compiler what data type it will be. Here the asterisk tells the compiler that the variable being defined will be a pointer and the word `int` indicates that the pointer will point to integers. I read that definition like this: I here define a variable named `some_ptr` which is a pointer to integers. But there is another way to look at it.

The definition of an integer variable, `int x;`, could be read like this: This variable x is an int. In the same sense, the above definition of a pointer could be read like this: This variable `some_ptr` is a pointer to int. It could also be read like this (starting with the asterisk): That which is pointed to by the variable `some_ptr` is an int. I like that one best myself.

Initialization

When a pointer variable is declared, memory is allocated for it just like any other variable. And it needs to be given a value just like any other variable. This is done with another new operator called the address operator which is symbolized by the ampersand (&). Once again, this symbol is used to mean other things in other contexts, but I don't think you will find that a problem. The address operator is a unary operator (an operator which takes a single parameter) like the indirection operator and returns the address of the symbol that follows it. Thus &x returns the address of x,

the actual number of the memory location at which the first byte of x is stored. This value can be assigned to a pointer.

Putting it All Together

Here is a simple example of the use of pointers.

```
#include <iostream.h>
void main(void)
{
    int x = 5, y = 23;
    int *int_ptr;
    int_ptr = &x;

    cout << *int_ptr << '\n';
    *int_ptr += 7;
    cout << x << '\n';
    int_ptr = &y;
    cout << *int_ptr << '\n';
}
```

The output from this program is:

```
5
12
23
```

First we define an integer variable called x, and initialize it to the value 5. Then we define a pointer variable which points to integers, called int_ptr. This line can be read "That which int_ptr points to is an integer." So far, just like an uninitialized variable of any kind, int_ptr has junk in it. It points to something, but we don't know what. So before we can use it, we must initialize it, we must make it point to something in particular. The next line assigns int_ptr the address of x. &x evaluates to the address of x, and it is assigned to int_ptr. As you might guess, this can also be done in one line like this: int *int_ptr = &x;. In either case x must be defined first. Note that you are putting the address of x (&x) into the pointer variable int_ptr, not into the place int_ptr points to (*int_ptr).

Now that int_ptr points to x (has the value of the address of x), we can use it to access the value of x. The next line of the program prints the value of *int_ptr to the screen. Remember to think of * as "that which is pointed to by", so that *int_ptr becomes "that which is pointed to by int_ptr" which is x. Thus, when we print *int_ptr, the value that is printed is the value of x and when we add 7 to *int_ptr we are adding 7 to x. *int_ptr is exactly equivalent to x as long as it holds the address of x. The only difference is that x always refers to the same thing, the same variable, but int_ptr is a variable and can be made to point to other integers.

A pointer may be made to point to a different variable (of the proper data type) by assigning to it a different address. In the program above, the statement int_ptr = &y; causes int_ptr to point to y, and thus, when we print *int_ptr to the screen the second time the value printed is 23, the value of y.

So far we see that a pointer to a variable can be used for everything the variable itself can be used for, and it can be shifted from one variable to another.

Caution

```
#include <iostreams.h>
void main(void)      // WRONG
{
    int *int_ptr;

    *int_ptr = 9;
    cout << *int_ptr << '\n';
}
```

What is wrong with this picture? Where is the 9 stored? You don't know? Neither do I. In the previous program, we initialized int_ptr to point to x, to the memory space allocated to the variable x. In this one, we did not give int_ptr a value, and so it could point anywhere. So always initialize pointers before using them.

This kind of error can cause extremely elusive bugs. Depending on the number that happens to be in the memory space allocated to the pointer, the 9 could overwrite any memory in the system, another variable, program code, operating system space, anything at all and cause any kind of problem. Or it could go in a safe spot and the program would run just fine. What is worse, the program could run fine most of the time and crash mysteriously once in a great while.

The Turbo C++ compiler protects you from some of these errors by initializing the pointer to 0 (NULL) for you. As near as I can tell, this is not part of the language standard, just something Turbo does. It is nice, but I sure wouldn't get in the habit of trusting it.

As a result of this, the program above runs fine, prints 9, and then prints a run-time error message which says "NULL pointer assignment". I will talk about it more later, but this is a good place to mention that the value 0 for a pointer is special. Note that I mean the pointer is pointing to memory location 0, not that the pointer is pointing to memory which holds the value 0. The system assures us that no useful information is ever stored at memory location 0 and so a pointer value of 0 (NULL) will never point to valid data. This value is often used to indicate an error condition.

Pointers as Function Parameters

Back when we first met functions, we saw that you could only pass one variable back from a function. This can sometimes be a problem. What if the function produces more than one result? Here is a way to solve the problem.

```
#include <iostream.h>
void switch_em(int *a, int *b);
void main(void)
{
    int x = 17, y = 23;

    cout << "\nX = " << x;
    cout << "\nY = " << y;
    switch_em(&x, &y);
    cout << "\nX = " << x;
    cout << "\nY = " << y;
}

void switch_em(int *a, int *b)
```

```
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
```

Note that if you call the above function `switch`, you will get a lot of error messages that don't seem to have much to do with the real problem. `switch` is a reserved word which means something quite different. Don't ask how I happened to think of that.

In the main function, we defined two integer variables. Memory was allocated for them, and values put into that memory. Then we called the function and passed to it the addresses of the two variables by using the ampersand (&) operator.

In the function, the parameters were declared as pointers to integers. Memory was allocated in the function for the pointers, but not for the integers they point to.

By passing the addresses into the function, where they filled variables declared as pointers to integers, we have caused the pointers in the function to point to the integer variables in `main`. Unlike the normal case of passing parameters into a function by value, here we are passing parameters by reference. There is only one integer variable named `x` and both `main` and the function `switch_em` have access to it.

In the function, since we have a pointer to `x` rather than `x` itself, we must refer to it as `*a`, read, "That which is pointed to by `a`". But we know that means `x`, because we know that `a` points to `x`.

One result of all this is that whatever we do to change `*a` shows up in `x` back in `main`. Because they are the same thing. Thus, we can swap the two values between `*a` and `*b` in the function and return to `main` to find that they are exchanged in `main` also.

Overloading the Function

This is a good time for a brief digression on overloaded functions. In old C, if we wanted to be able to switch the values of two character variables too, we would have to call the function something different like `switch_chars` or some other awkwardness. In C++, we can call the function by the same name as long as it differs in its parameter list. Otherwise, the compiler wouldn't know which one to call.

```
#include <iostream.h>
void switch_em(int *a, int *b);
void switch_em(char *a, char *b);
void main(void)
{
    int x = 17, y = 23;
    char w = 'Q', z = 'U';

    cout << "\nX = " << x;
    cout << "\nY = " << y;
    switch_em(&x, &y);
    cout << "\nX = " << x;
    cout << "\nY = " << y;

    cout << "\nW = " << w;
```

```
    cout << "\nZ = " << z;
    switch_em(&w, &z);
    cout << "\nW = " << w;
    cout << "\nZ = " << z; }
void switch_em(int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

void switch_em(char *a, char *b)
{
    char temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
```

The compiler uses the function whose parameter list matches the parameters in the function call in each case.

As always, we must provide the compiler with prototypes for the functions above `main`. Actually, it is not the case that the compiler requires prototypes for functions, but that the compiler must know about each function before it is used. This can also be accomplished by putting `main` last in your source file and ordering the other functions so that they are defined before they are called. I sometimes do this during development, but I always rewrite the files with `main` first and put in the necessary prototypes. I think the program is easier to follow if it progresses from the highest level to the lowest, rather than the other way 'round.

Pointers and Arrays

Pointers are occasionally useful to return data from a function, as shown above, but their real power comes in dealing with arrays and other complex data structures.

```
#include <iostream.h> void main(void)
{
    char char_a[] = "This is a string.";

    cout << *char_a;
}
```

Since `char_a` points to the first element in the array, when we print that which `char_a` points to, `*char_a`, we find a T on the screen. What then do you suppose we will get if we print that which is pointed to by `char_a + 1`, `*(char_a + 1)`? Right, it will print the h that follows the T. For if `char_a` points to the first element in the array, surely `char_a + 1` points to the second element and `char_a + 2` to the third. Thus, we can print the whole string like this:

```
#include <iostream.h> void main(void)
{
    char char_a[18] = "This is a string.";

    for(int i = 0; i < 18; i++)
        cout << *(char_a + i);
}
```

The array notation for the `i`th member, `char_a[i]`, is the same as the pointer representation, `*(char_a + i)`.

Remember, since this is a string, we could have also written the above program

like this:
```
#include <iostream.h>
void main(void)
{
    char char_a[18] = "This is a string.";

    for(int i = 0; *(char_a + i); i++)
        cout << *(char_a + i);
}
```

This program uses the fact that all strings end in a NULL (0), and thus `*(char_a + i)` will evaluate to NULL or false when it reaches the end of the string. This saves you from needing to keep track of how long everything is, and so is a very common technique.

Let us look now at a function which uses this idea to count the length of a string.

```
#include <iostream.h>
int slen(char *str_ptr);
void main(void)
{
    char char_a[18] = "This is a string.";

    cout << "The length is " << slen(char_a);
}

int slen(char *str_ptr)
{
    int i = 0;

    while(*(str_ptr + i++))

    return(i);
}
```

The while loop runs until `(str_ptr + i)` points to the NULL at the end of the string, at which time `i` is equal to 18. The function then returns `i` to the calling code. Is that the length of the string? It is surely the length of memory necessary to store the string, including the NULL, but most authorities would say that the length of the string was 17. The string length function in C, called `strlen`, returns the length without the NULL. If we want to get the length without the NULL, we can move the increment into the body of the loop, so it does not occur the last time. Like this:

```
while(*(str_ptr + i))
    i++;
```

As always, there is another way to do this. Since the character pointer declared in the function is a variable, we can increment it instead of adding `i` to it each time. `i` is used here only to keep track of the number of iterations of the loop.

```
int slen(char *str_ptr) {
    int i = 0;

    while(*(str_ptr++))
        i++;
    return(i); }
```

Note the parentheses around `str_ptr++`. If you leave them out the thing `str_ptr` points to will be incremented instead of `str_ptr`.

This can only be done with pointer variables like `str_ptr`. If we wanted to find the length of the string in the main function instead of calling another function, we could not increment `char_a`. That is because `char_a` is the name of an array, and a pointer to the head of the array, but it is a

pointer constant rather than a variable. If its value could be changed it would no longer point to the head of the array.

Let's try another example. The function in this one will double the value of each element in an array of integers.

```c
#include <iostream.h>
void dbl(int *i_ptr, int len);
void main(void)
{
    int numbers[5] = {76, 93, 7, 128, 11};

    dbl(numbers, 5);
    for(int i = 0; i < 5; i++)
        cout << "\nElement " << i << " is "
                         << numbers[i];
}

void dbl(int *i_ptr, int len)
{
    for(int i = 0; i < len; i++)
        *(i_ptr++) *= 2;
}
```

This works very much like the last program, passing the address of the array to the function along with the length, and stepping through the array by incrementing the pointer. There is one odd thing which you may have noticed. In the last program, we added one to the pointer and it pointed to the next character which seemed reasonable as characters take up one byte each. In this example we increment the pointer and it seems to point to the next integer even though integers take up two bytes.

Pointer Arithmetic

The answer is that the compiler knows what a pointer points to from its declaration, so when we add one to a pointer its value is increased enough to point to the next of whatever it points to. This is also the case in other kinds of pointer arithmetic. For example if we subtract one pointer from another, the answer we get is the number of whatever they point to that can fit between them.

The arithmetic operations legal on pointers are assignment of the value of a pointer to another pointer of the same type, adding or subtracting a pointer and an integer, comparing or subtracting two pointers to the same array or string, and assigning or comparing to zero. We cannot assign values between pointers or do pointer arithmetic unless the pointers involved point to the same kind of thing.

The `void` pointer (declared like `void *ptr`) is an exception. It can point to any kind of data, but cannot participate in any kind of arithmetic. The `void` pointer is used when you do not know in advance what kind of data it will point to. Its value can be assigned to a pointer of another type with the use of a type cast. I will go into some of the more complex uses of pointers next time.

The End

The Zinc interface library didn't get better with further acquaintance. It makes beautiful windows like Windows does, which you can move, resize, scroll, and all, but it is difficult to use, largely due to the documents, and it is inflexible. For example, I needed to build a series of data entry and display screens which would look about the same in graphics mode and text mode. The only way to get ZINC to do a 25 line high screen in graphics mode is to buy the source code and buy a new bit-mapped font from someone else and do what the Zinc tech service person described as "Quite a bit of work" to install the font. And then the same again to get the screen to hold 25 lines in CGA. It might be very useful if you want to do exactly what it does, but I think I'll pass.

Sources

C++ Techniques and Applications
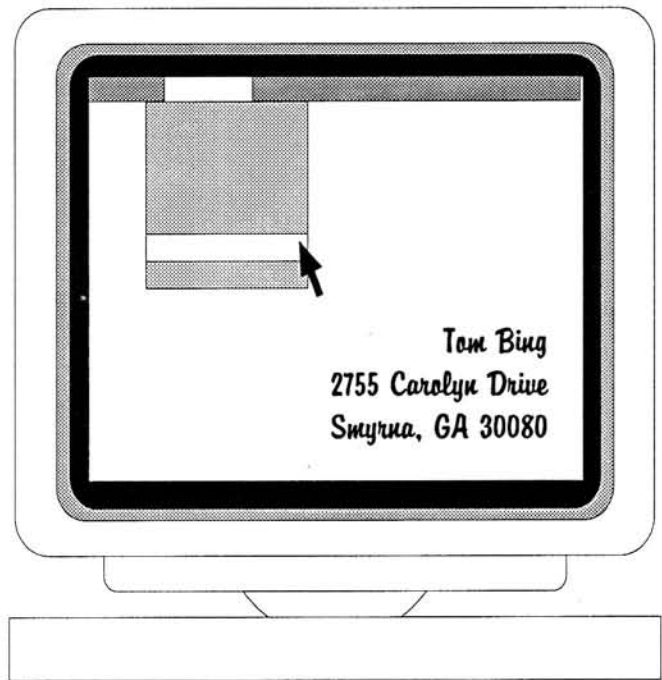by Scott Robert Ladd
M&T Books, 1990
$39.95 with disk

The Tao of Objects
by Gary Entsminger
M&T Books, 1990
$26.95

Zinc Interface Library
Zinc Software Incorporated
405 South 100 East, Suite 210
Pleasant Grove, Utah 84062
(801) 785-8900
$199.95, source $200 ✳
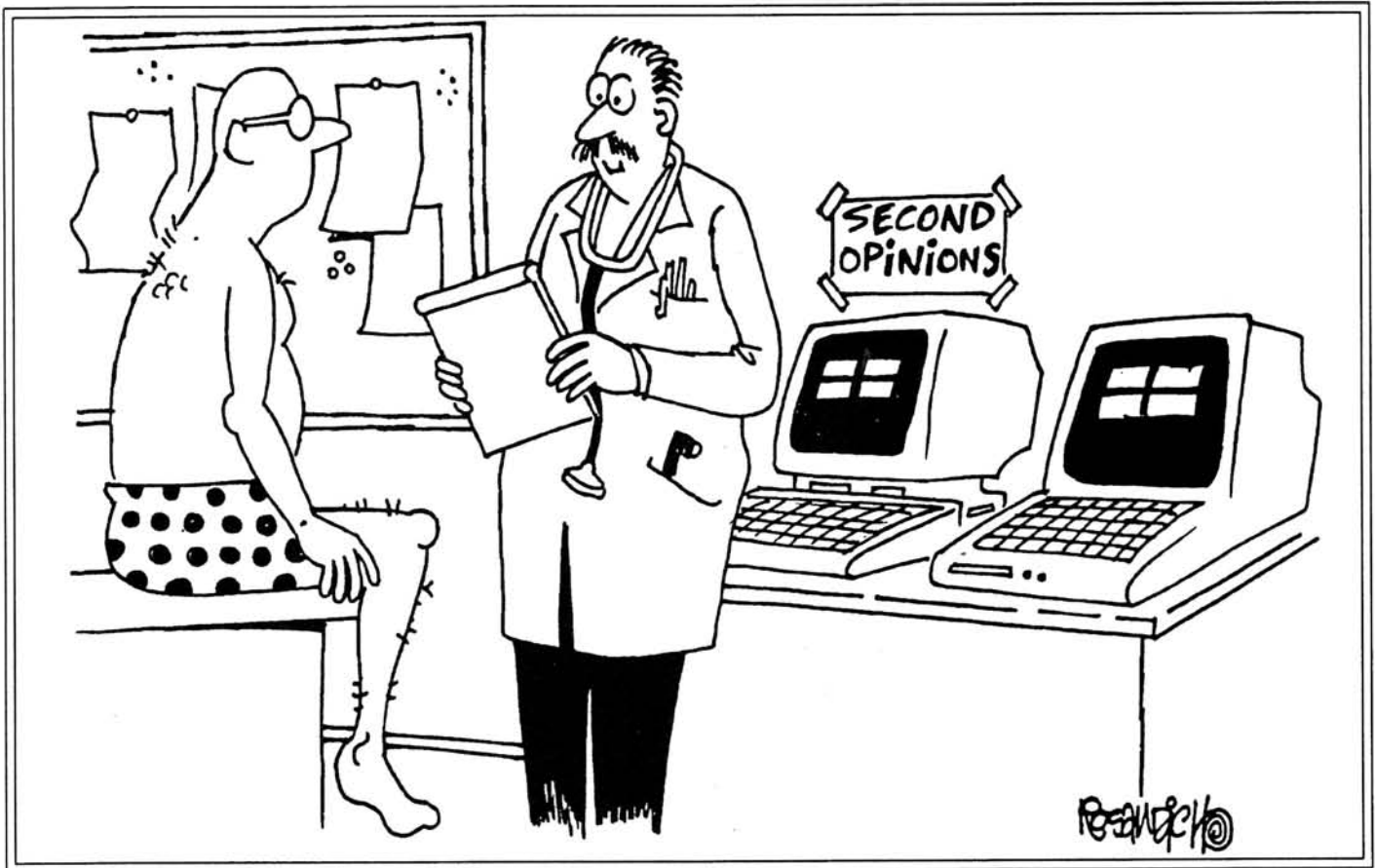
Learning C By Computer

Part 1: Master C

Tom Bing
2755 Carolyn Drive
Smyrna, GA 30080

"There is no royal road to geometry". At least, that's what the teacher Menaechmus reportedly told the young Alexander, who was doubtless anxious to leave off with pictures in the sand and get on with conquering the world. Today, mastery of subjects like C programming still isn't trivial, but there are tools that make the road less rough, if not quite royal. One of these is the "Master C" book and disk combination from the Waite Group.

Let's say you want to learn to program in C. What are your choices? Well, night classes at the community college will cost between $100 and $200 for a one-quarter course, and you have to adjust your work and family life to attend class. You can buy any of several excellent books on the subject, but this approach requires a lot of digging by the student. If you use a classic book like Kernighan and Ritchie's "The C Programming Language", you may also need access to a working C compiler to test your answers to problems. If you can afford the air fare and the $1400 tuition, you may want to take AT&T's one-week course in C programming.

By now you're thinking, "There must be a better way!" I agree. We ought to be able to use the personal computer itself to guide us through the complexities of C, without having to endure the mind-numbing simplicity of some program that just automates a multiple-choice test. The folks at the Waite Group must think so too; they have developed the "Master C" book and disk combination, priced at $44.95. I have used this product to brush up on C, and I wish I'd had it years ago when I wrote my first C programs. Although I can't evaluate it as a C beginner, I think it ought to be fairly easy to follow, even if the longest program you've written is a 15-line BASIC number-

guessing game. After working all the way through it, here are my major findings:
1. Master C is menu-oriented, but not stifling.
2. You'll need a hard disk with two megabytes free. A color monitor is very nice to have, but not essential.
3. You can complete the course without a full-scale C compiler; the lessons show the outputs from the example programs.
4. The book is a good reference for further study, but you can work your way through the course without reading it. You will need to refer to it for installation.
5. The tutorial program does an amazing

job of simulating one-on-one sessions with an expert teacher. The student is warned, however, that while it tolerates sloppy English in some answers, it makes stringent checks on the C program statements that you write. (But that's the point, isn't it?)
6. You don't have to slog through it in one sitting. Master C will "bookmark" your progress and return you to the proper sub-menu when you re-run the program.
7. The mix of "lecture" and hands-on is about right for a beginning C programmer, I think. There are a lot of cute tricks in the program (for instance, it calls you

```
THOMAS                          Master C

                         Master C

        DONE       1. C and Programming
        DONE       2. Introducing C
        DONE       3. Data and C
        DONE       4. Character Strings: Input/Output
        DONE       5. Operators/Expressions/Statements
        DONE       6. C Control Statements: Looping
        DONE       7. C Control Statements: Branching
        DONE       8. Character Input/Output
        REVIEWED   9. Functions
        REVIEWED  10. Arrays and Pointers
        DONE      11. Character Strings/Functions
          ▪       12. File Input/Output
          ▪       13. Storage Classes
          ▪       14. Structures
          ▪       15. C Preprocessor/C Library

                   Q. Quit

Choose Option Below by its First Letter or Choose Menu Number:
Achievement Glossary Calc Write Help Refresh Obj Jump Tutor Quit
```

Figure 1. Master C Main Menu with "A" Option Selected.

by name in critiquing your answers) to keep you interested.

Installation

Book-and-disk combinations are popping up more often in the stores nowadays. However, Master C is the first I've seen that really required a hard disk — 2.2 megabytes, in fact. You will also need one 5-1/4 inch floppy drive, at least 384K of RAM and an IBM-compatible PC with DOS 3.0 or later. You don't need a color monitor, but color certainly lends emphasis and enhances interest and understanding. If you want to run a TSR program like Sidekick or PC Tools while you use Master C, more RAM may be needed. You must follow the installation procedure "by the book"; the files on the four 360K disks are compressed, and the Master C installation program is needed to expand them. Master C records your name and program serial number the first time the program is installed. The program may be removed and re-installed later; later installations don't require you to enter your name and serial number again. The disks aren't copy protected, but under the terms of the license, Master C can only be used legally on one computer at a time. Your scores and completion of units are recorded in a file in the C:\MASTERC\REGISTER directory as work progresses. This allows you to pick up where you left off when you re-run the program.

Here's a minor point, but it may save you some irritation; when the program asks for your name during installation, put in your nickname if that's what you like to be called. I put in my given name, and it was a little distracting to have the program print, "Good work, THOMAS!" after correct answers; it sounded too formal.

Course Organization

The main menu lets you select from among 15 "chapters" from "C and Programming" through "C Preprocessor/C Library" (Figure 1). In Figure 1, the "A" (Achievement) option has been selected, causing the completion status of each chapter to be displayed. When a chapter number is selected, a sub-menu with lessons for that chapter is displayed (Figure 2). Making a sub-menu selection will cause a series of lessons and exercises on that topic to be displayed. On some chapters there is a third menu level for some of the topics.

The 15 chapters of Master C closely parallel the structure of two Waite Group books: *New C Primer Plus* and *C Step-By-Step*. In fact, the only book chapter omitted from the course is the one on bitwise operations. Both the books and the program conclude with a discussion of the C preprocessor.

The two books and Master C all cover the same subject matter. What makes us prefer one over the others? Do they com-

```
THOMAS                          Master C
─────────────────────────────────────────────────────────
                     12: File Input/Output

        100%    1. Communicating with Files
        100%    2. Low-Level I/O
        100%    3. Standard I/O
        100%    4. A Simple File-Condensing Program
        100%    5. File I/O: More Functions
        100%    6. Random Access: fseek()/ftell()
        DONE    7. Behind the Scenes with Standard I/O
         ▪      8. Other Standard I/O Functions
         ▪      9. Review

                D. Done


────────────────────────────────────────────────────────
Choose Option Below by its First Letter or Choose Menu Number:
Achievement Glossary Calc Write Help Refresh Obj Done Jump Tutor Quit
```

Figure 2. Chapter 12 Submenu.

plement each other? My choice, if I could have (or afford) only one, would be Master C. Using it is like having personal tutoring from a very knowledgeable C wizard — one who doesn't get tired or cranky, who is ready whenever you are, and supremely accurate. I was really impressed by the range and depth of knowledge that went into the program.

The 230-page book that comes with Master C is designed to get you into the program quickly and painlessly. Only 46 pages are needed to introduce you to the Master C program; the rest (Part II) is a reference manual, explaining C's "vocabulary" — specific operators, keywords, functions, and common macro definitions. Clearly, the Waite Group intended the program itself to be the dominant teaching tool. In my view, a good way to use the book and program together is this: skim through Part I of the book to install the program and learn how it's organized. Then work through the first three chapters of the program. At this point, you can return to the book and read through Part I again. The second reading will improve your grasp of program features before you tackle the heavy stuff in the later chapters.

The other Waite Group books are also useful, though not essential to completing the course. The *New C Primer Plus* is an encyclopedic (730 page) volume for personal study or reference use. *C Step-by-Step* is more of a classroom text, with review questions and exercises rounding out each chapter. The latter book also has numbered paragraphs that correspond to Master C's chapters and sub-chapters, ex-

cept for Master C's omission of bitwise operations, as I mentioned before. While both books are well done, I prefer Master C for three reasons. First, you can't fake it — you really have to interact with the program and answer questions intelligently to pass each lesson. Second, Master C will give you a basic knowledge of C quickly; just don't try to complete all 31 hours' worth of material in one sitting. Third, Master C will let you know how the ANSI C standard has changed some programming practices. These changes are things you might overlook even if you had access to a full C compiler. Master C was two years in the making, and the careful preparation shows in the quality and depth of the lessons.

When you're working through Master C, you're doing one of three things — reading explanatory text, responding to questions, or using the "option bar" to navigate the program or to get help. Options can be selected by typing the first letter of the option name. Option names are:

Forward: Move to next lesson screen.

Back: Return to previous lesson screen.

Note: Display supporting information for the current screen.

Example: Display program runs or specific code examples.

Achievement: Display completion status and scores for items on current menu.

Glossary: Define terms and keywords entered by user.

Calc: Pop-up calculator for math

Write:	Display address and telephone number of Waite Group.
Help:	Display context-sensitive help message.
Refresh:	Repaint the screen, removing any Help, Note, or Example windows.
Obj:	Display objectives of selections on the current menu.
Jump:	Go to specific chapter and section requested by user.
Tutor:	Display a series of lessons on using Master C.
Done:	Exit without saving achievement information.
Quit:	Save scores and "bookmark" to allow resuming at the correct item, then exit.

problems.

Not all these options appear on every menu or lesson screen. Whenever the "Note" or "Example" selection occurs among the options, it will be in flashing letters so you won't overlook it. When these options are selected, they will display screens giving further detail about the current lesson. The Glossary option is versatile, allowing you not only to find definitions for key terms, but also to jump directly to lessons related to glossary entries. The Jump option also allows you to go directly to the numbered lesson section which you specify. The Achievement option will display the completion status of each lesson listed on the current menu (DONE, BEGUN, 90%, etc.). It's very reinforcing to select "A" and see good scores for the parts you've completed.

In several cases, Master C will coach

```
QUESTION: The fgets() function takes three arguments. Name them in their
correct order.

What answer best fits?
pointer to char, integer limit on bytes to be read, pointer to file
```

```
 FEEDBACK 
 No. It should be: It takes a pointer to char where input is to be stored, an
 integer representing the maximum number of characters, and the file pointer.

 You are having difficulty and can review some related material.
 The Done option offers an early exit from the review.
```

```
Choose Option Below by its First Letter:
Forward Calc Write Obj
```

**Figure 4. An Exception to Master C's
Tolerance of Loosely Worded Answers.**

the student in writing a short program or function by "blackboarding". In this technique, the student is given a basic description of the desired program or function. He or she is then asked to supply C statements to implement the program. As each statement is supplied, Master C will either accept correct answers or provide review and feedback for incorrect ones. Either way, the accumulated correct answers are shown on the right side of the screen (Figure 3). By this method, beginners get used to writing simple programs very quickly.

Study Techniques

What are some ways to extract the maximum benefit from Master C? First, when you're reading explanations in lessons, have a note pad handy to jot down key points, such as definitions and highlighted terms. You can be sure these will show up in questions. If you answer a question "I don't know", or with some other definitely incorrect answer, you will be shown a few screens of review material dealing with the question. After that, you'll have another shot at the same question, or a similar one. You can't move forward or back through the lesson when you are being prompted for an answer. Jotting down important terms or definitions will enable you to answer correctly the first time.

If you have a desktop program like Metro or Sidekick, you can save screens displayed by Master C to a file, without stopping the operation of the program. A utility like the SNIPPER program from PC Magazine, available through the Zenith Users Group's COM1 BBS, will also save screen images. If you run across a lesson that is especially important to you, or maybe an example that you have questions about, capturing a screen display in a file will provide a useful reference and will enable you to easily explain your questions to another C programmer or a member of the Waite Group staff. All that is required is that SNIPPER or the utility you are using be loaded as a TSR program before you run Master C. I used SNIPPER to capture the

```
QUESTION: Imagine that you have
the following code. Now you want
to combine the first and last
names in full using the following
format. If you enter: Jim Hill
then full will contain: Hill, Jim

What is the first step.

What answer best fits?
strcpy(full,last);
```

```
 FEEDBACK 
 Well Done.
```

```
 ACCUMULATED ANSWERS 
#include <stdio.h>
main()
{
    char first[10], last[10], full[21];

    printf("Enter first and last name:");
    scanf("%s %s",first,last);
```

```
Choose Option Below by its First Letter:
Forward Back Glossary Calc Write Help Refresh Obj Done Tutor Quit
```

**Figure 3. The Blackboarding Method
of Writing Programs.**

screen displays in this article. However, I still think it's vital to make handwritten notes. It's my opinion that the act of writing down information makes you more likely to remember it than just capturing it in a file.

When you're answering questions by supplying definitions or missing words, Master C is remarkably flexible in recognizing a correct answer. For instance, in one question, a function contains a local automatic variable with no initial assigned value ("int lines;"). The student is asked what value will be assigned automatically to "lines". My answer was "The value is indeterminate", which is true. This was scored as correct, even though the "standard" answer was "The value is undefined". Be warned, however; you can't be sloppy in answering questions that require you to type in C statements. While Master C will coax you along if you have a partial C statement that's correct as far as it goes, be sure to balance parentheses and brackets and use semicolons correctly. Master C is very helpful about explaining why a C statement is wrong, but in the end, we have to enter answers that are exactly correct, because that's all a C compiler will accept. As the Master C book says, "when questions deal with the details of C programming code, be sure to include all the elements".

Praises And Complaints

I think the greatest strengths of Master C are that it's highly interactive and seems to adapt itself to the situation, offering reviews in those areas where the student is having trouble. I especially like the clear and detailed explanation of arrays, pointers, structures, and storage classes. If C is going to give you trouble, it's likely to be in one of these four areas, and this course covers them quite well.

I don't like the way the "Done" and "Quit" options are used. The "Done" or "D" option is used to exit the section you are in (or the entire program) without saving the score you made on that section. In order to save score information and then exit the Master C program, you must select "Quit" or "Q". I see the need for both options, but the choice of option names is counter-intuitive to me. I think of something "Done" as being completed, while "Quit" means to leave unfinished. However, if you'll just remember to use "Q" to exit when you want to save your work, you'll do fine.

There were two or three misspellings and typos I saw in the program, but the vast majority of screens are accurate. I had one case where Master C's tolerant interpretation of "right answer" did not seem to hold (Figure 4). Apparently, there are limits to the way the same idea can be recognized in different words. All these concerns were minor to me, in view of the overall effectiveness of the program.
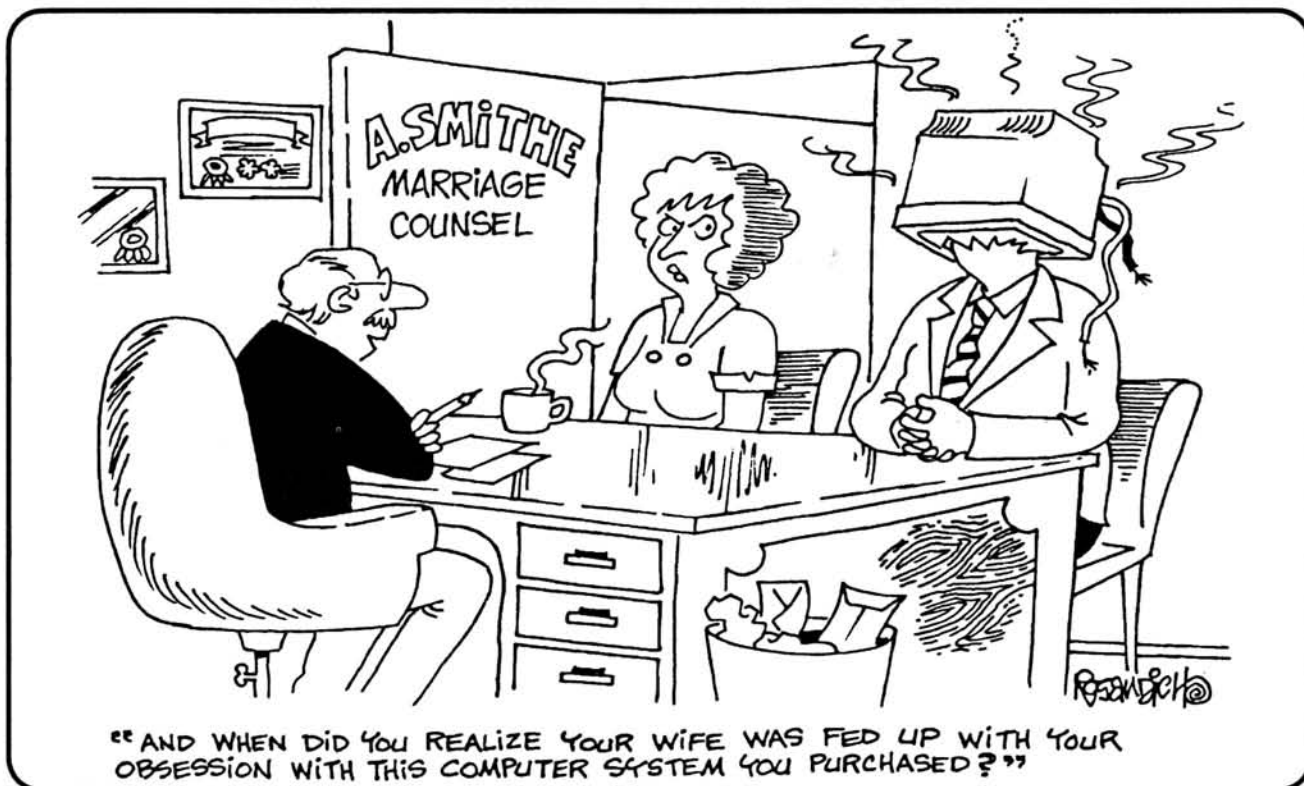
Conclusion

I am an intermediate-level C programmer, neither a wizard nor a beginner. I will venture the opinion that beginners will find Master C an aid to learning C fundamentals quickly and thoroughly. As an expe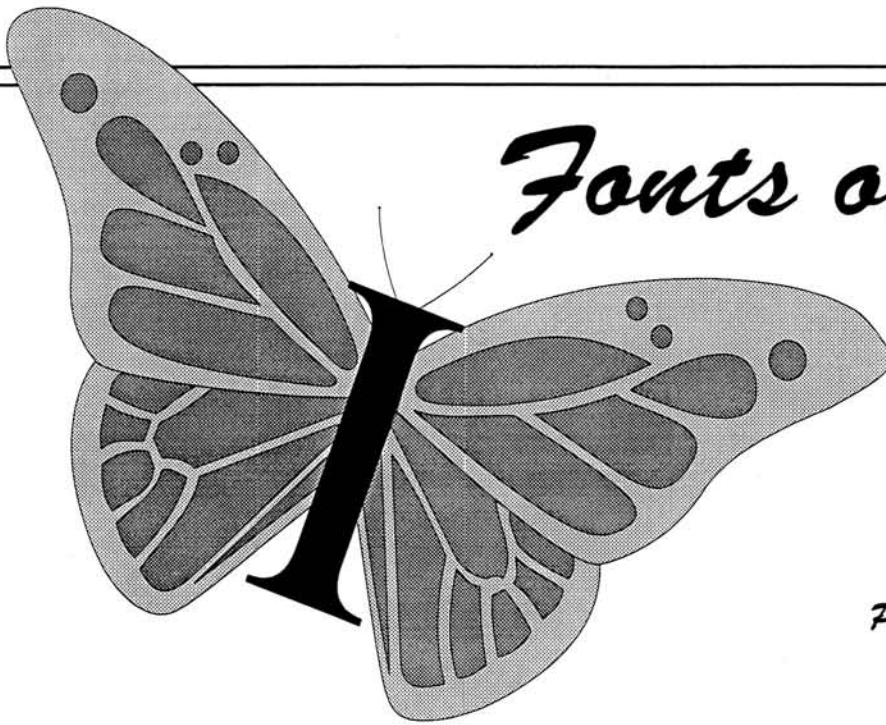rienced programmer, I was impressed by Master C's approach to the finer points of the language; I thought the presentation of topics such as pointers was made as simple and clear as possible. Master C is flexible and usually fair in evaluating student answers. The glossary and "jump" options allow a user to browse through any part of the material which may require further study. It's not a full-blown compiler, but you can install it as a menu selection accessible from Microsoft Quick C. After you finish the course, you probably won't be ready to write complex code (like device drivers or operating systems), but you should be able to tackle simple programming tasks with confidence.

Master C should be available at most major bookstores that carry a full line of computer titles. Its current U.S. price is $44.95. In future articles in this series, we'll look at two other computerized C tutorials: "Learn C Now" from Microsoft and Osborne's "Turbo C++ Disk Tutor". ✳

" AND WHEN DID YOU REALIZE YOUR WIFE WAS FED UP WITH YOUR OBSESSION WITH THIS COMPUTER SYSTEM YOU PURCHASED ? "

Fonts on the Fly

Alan Neibauer
11138 Hendrix Street
Philadelphia, PA 19116

One of the important specifications of any printer is the number of fonts that it contains. Most printers today come with several fonts. Dot matrix printers usually have a default upright and italic font, as well as draft and near letter quality modes. Ink jet and laser printers have the widest variety.

But except for Postscript printers, the size of fonts is limited to those available internally, on optional cartridges, or downloadable as softfonts on disk.

Collecting cartridges and softfonts can be an expensive proposition. So many of us are envious of Postscript owners, and those lucky enough to have a Hewlett Packard LaserJet III. These printers can create fonts on the fly; that is, they generate almost any size font that you need at the time of printing. You do not have to store individually sized fonts on disk or cartridge. In fact, many of the outline fonts used to generate fonts on the fly are built into the printer's ROM so they are available immediately without downloading or insertion.

Fortunately, fonts on the fly, or scalable fonts, are now available to the rest of us.

William G. Nabor did an excellent job of explaining one scalable font package in his article, *Adding LaserJet Soft Fonts to WordPerfect 5.X with Bitstream's Facelift*, in the February 1991 Remark. He explained how to install Facelift into WordPerfect, and some basics about fonts.

In my own articles, *An Introduction to Postscript* (REMark, August 1990) and *Implementing Postscript* (REMark, September 1990), I explain about Postscript.

In this article, I'll look in more technical detail at font generation and some alternative ways to take advantage of it with your dot matrix and laser printer.

Outline Fonts

To understand how fonts on the fly are generated, first visualize a pre-sized font, such as a softfont that you can download to a laser printer. The shape of each character is stored as a bitmap pattern showing where each dot of toner must be applied to form the image. Figure 1, for example, shows the bitmap pattern for the lowercase letter b. Each 1 represents where a dot of toner would be applied to paper, each 0 a blank area.

Notice that this bitmap pattern will only form a character in one size, 32 dots high and 24 dots wide. With a resolution of 300 dots per inch, the character is roughly from an eight point font. If you wanted to print the character in any other size, you'd need an entirely new bitmap pattern on disk.

A downloadable font stored on disk contains a bitmap pattern for every character in the font. To print characters in two sizes, you need two font files, each with the bitmap pattern for every character in the font. The larger the font, the more bits used for each character, and thus the larger the size of the font file.

A single 24-point font in the PC symbol set (which includes all of the characters you can display on the screen) can take up 90K bytes of space or more.

That's why William's complete set of fonts used 27 MB of disk space.

Postscript and other printers that generate fonts on the fly use font outlines rather than bitmaps. An outline font contains a description of the typeface, rather than the dot-by-dot instructions for printing the font in a single size. Each character in the font, as illustrated in Figure 2, is com-

| Row	Bit Map Pattern			
01	00111100	00000000	00000000	00000000
02	00111100	00000000	00000000	00000000
03	00111100	00000000	00000000	00000000
04	00111100	00000000	00000000	00000000
05	00111100	00000000	00000000	00000000
06	00111100	00000000	00000000	00000000
07	00111100	00000000	00000000	00000000
08	00111100	00000000	00000000	00000000
09	00111100	00000000	00000000	00000000
10	00111100	00000000	00000000	00000000
11	00111100	00000000	00000000	00000000
12	00111100	00000000	00000000	00000000
13	00111100	00000000	00000000	00000000
14	00111100	00000000	00000000	00000000
15	00111100	00000111	11111110	00000000
16	00111100	00011111	11111111	00000000
17	00111100	00111111	11111111	11000000
18	00111111	11110000	00000000	11111100
19	00111111	11100000	00000000	00011110
20	00111100	00000000	00000000	00001111
21	00111100	00000000	00000000	00001111
22	00111100	00000000	00000000	00001111
23	00111100	00000000	00000000	00001111
24	00111100	00000000	00000000	00001111
25	00111100	00000000	00000000	00001111
26	00111100	00000000	00000000	00001111
27	00111111	11000000	00000000	00111110
28	00111111	11110000	00000000	01111100
29	00111100	01111000	00000001	11111000
30	00111100	00111111	11111111	10000000
31	00111100	00011111	11111111	00000000
32	00111100	00001111	11111110	00000000

Figure 1
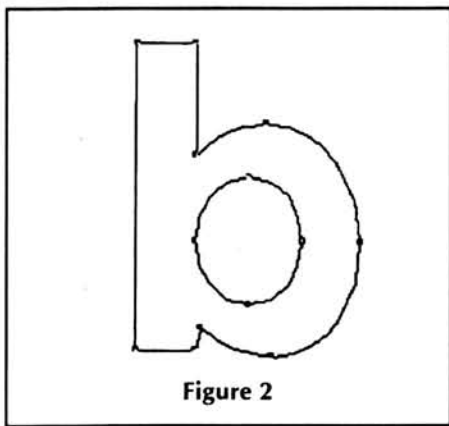
Figure 2

posed of a number of points connected by straight lines and segments of arcs. The lines and arcs describe the shape of the characters, not its size for any single font.

In addition to the lines and arcs, most outline fonts also contain "hints" that tell the generation program how to expand the character to any size with the best results. The program uses the lines, arcs, and hints in the outline to generate the individual characters in the sizes called for when the document is printed.

While the algorithms that actually create the fonts are rather complex, the size of the printed character is determined by the length of the lines and arcs connecting each of the points. The longer the lines and arcs, the larger the character.

To create italic characters, the program angles each of the segments in relation to the baseline, or the line on which the character prints on the page.

By modifying length, angles, and the relationship of the horizontal and vertical axis, the character can be shaped in many different ways. For example, expanding the vertical segments more than the horizontal segments creates a taller, narrower font. If you expand the horizontal segments more than the vertical ones, you create a compressed, wider font.

Outline fonts are built into Postscript and LaserJet III devices. The printers also include their own control language that converts outline characters into fully-formed characters of the proper size and shape. Postscript printers use a Postscript Interpreter, LaserJet III printers use PCL Level V.

Applications with a compatible printer driver tell the printer what characters to print and in what size. The printer's language then generates the fonts from the outlines.

Font Generation

In the past, those of us without Postscript or LaserJet III printers had to seek alternates to fonts on the fly. Purchasing and storing individually sized bit-map fonts could be expensive and take up quite a bit of disk space.

So Bitstream, Hewlett-Packard, VS, and other companies developed a compro-

mise to fonts on the fly in font generation programs. These did not produce fonts on the fly. Instead, they included outline fonts on disk with which you could generate almost any size bit-map font. In addition, they installed the fonts in your application's printer driver, so the driver could download it to the printer when needed. Of course, you had to create the entire bitmap font and store it on your disk before you needed it. The files required the same amount of disk space as traditional bitmaps, but at least the system gave the non-Postscript user a wider variety of fonts and sizes than before.

Bitstream, to their credit, made font generation inexpensive by offering it free.
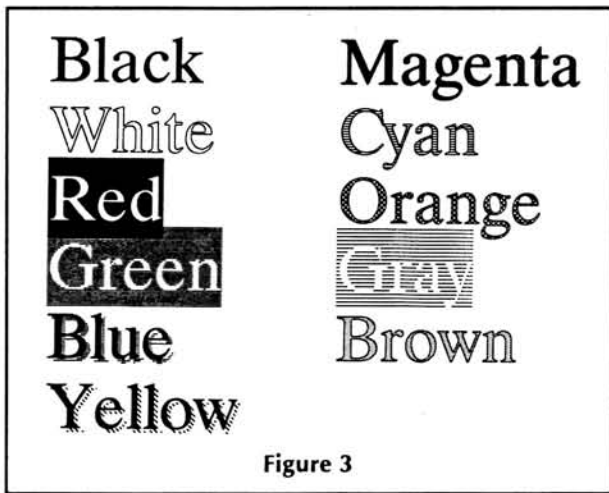
Figure 3

Manufacturers, such as WordPerfect and Microsoft, could include a sample of Bitstream outline fonts and Fontware (the generation program) with their applications. The other font generation vendors made their programs available at minimal cost. Of course, all of them hoped to generate income through sales of additional outlines.

In summary, the process involved these steps:
1. Generate the entire font bit-map for the sizes you wanted to use.
2. Install the fonts in the program's standard driver.
3. Download the entire font as needed.

Once the fonts are created, the generation program is no longer part of the process. The application driver takes control just as if the font were purchased from any source.

Fonts on the fly for the rest of us was only a matter of reapplying these same outlines in a different way. Rather than generate the entire bitmap before hand, the information for the font outline is added to the driver, in the same way the driver contains details on Postscript fonts. You then start the application by running a loader program. This installs a small TSR program that starts the application, but which intercepts all printer requests. Print jobs are transferred to the font on the fly generation program, which creates from the outline the characters needed by the document and transmits them to the printer. The process is now:
1. Install information about the font outline in the program's driver.
2. Intercept print jobs.
3. Generate and download the characters.

In his article, William Nabor explained that after installing Facelift with WordPerfect, you are asked to enter a font size when you select a font to use. This is the same prompt that appears with WordPerfect's Postscript driver.

Scalable fonts no longer required the user to first generate and store individual font files on disk. Like Postscript, you only needed the outline file on disk and, like Postscript, you had fonts on the fly in almost any size.

Special Effects

Glyphix, from SWFTE International, is another example of fonts on the fly software using scalable outlines. Glyphix is installed much like Facelift. It modifies the printer driver and installs a TSR program that intercepts the print request. The driver

0	Normal (0)	H	40% Shade (17)
1	Outline (1)	I	60% Shade (18)
2	Reverse (2)	J	80% Shade (19)
3	33% Reverse (3)	K	15% Striped Background (20)
4	15% Reverse (4)	L	25% Striped Background (21)
5	Fat Stripes (5)	M	Shadow Horizontal Stripes (22)
6	15% Stripes (6)	N	Shadow Vertical Stripes (23)
7	25% Stripes (7)	O	Shadow Diagonal Left (24)
8	33% Stripes (8)	P	Shadow Diagonal Right (25)
9	50% Stripes (9)	Q	Shadow Squares (26)
A	Vertical Stripes (10)	R	Shadow Cross Hatch (27)
B	Diagonal Left (11)	S	Shadow 20% (28)
C	Diagonal Right (12)	T	Shadow 40% (29)
D	Squares (13)	U	Shadow 60% (30)
E	Cross hatching (14)	V	Invisible Print (31)
F	5% Shade (15)		
G	20% Shade (16)		

Figure 4

can create outline and shadow fonts if you select these options on the appearance menu. But it also assigns each of WordPerfect's color options to a special effect.

By pressing Ctrl-F8 C, you can select from 11 standard colors. While changing colors has no effect with a non-color printer, the Glyphix driver converts the color codes to special effect fonts, as shown in Figure 3. Selecting Red, for example, prints the current font in reverse, selecting orange prints in a basketweave filled pattern.

For added convenience, the program comes with a WordPerfect macro that assigns 32 effects (Figure 4). The macro selects effects using the Other option in the color menu, and assigning percentages of red, green, and blue. The driver translates the WordPerfect code to the appropriate effect.

Like all font on the fly systems, Glyphix doesn't create and download the entire font, just the characters that have to be printed. So, the generation process is remarkably fast. Printing a large headline using Glyphix is much faster than using a standard bit-map font which requires the entire font to be downloaded first.

Even the special effects are created quickly from the same outline font information used to generate standard characters. For example, compare the standard and special characters shown in Figure 5. The
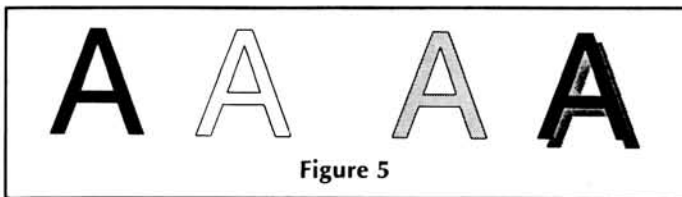
Figure 5

outline character is created by "drawing" the strokes of the character without filling in the dots between them. Striped, hatched, and other effects are created by filling in the spaces between strokes with a special pattern. Shadow characters are formed by printing a duplicate of the character at a slight offset. A wide variety of effects are formed by changing the pattern and offset of the duplicate.

Glyphix outline fonts come in two varieties; Standard and Foundry. One standard outline is used to create normal, bold, and italic fonts of the same typeface, in any size. The foundry outlines produce a higher quality image because separate normal, bold, and italic outlines are provided. Like Facelift, Glyphix also has an optional dingbat font (called Zingbats) that also requires you to enter codes using the Ctrl-V or Ctrl-2 command.

Prescript

In my article, *Implementing Postscript,* I reviewed two Postscript interpreters — Goscript and Ultrascript. These are software alternatives to purchasing a Postscript printer and they work much the same way as Bitstream and Glyphix. The major difference is that they utilize your application's current Postscript printer driver.

You setup your application as if you have a Postscript printer. The TSR intercepts the print job and transmits it to the generation program which interprets the standard Postscript commands and creates the fonts from its own outlines. The outlines are width compatible with those from Postscript, so as far as your application is concerned, you have a Postscript printer. You do not have to install any fonts in your application's driver and can use the fonts with any program that works with Postscript printers.

Since I wrote that article, I've had a chance to review another Postscript interpreter, Prescript version 1.2, from Pan Overseas Computing.

Like other interpreters, Prescript comes in two versions; Standard and Deluxe. The standard version includes 28 fonts, the deluxe version, 50 fonts, 15 in addition to the 35 supported by most Postscript printers.

Both versions of the program are identical except for the number of fonts supplied. Both also include two different operating modes, on-line and off-line.

In the on-line mode, Prescript intercepts standard Postscript print jobs, generates the fonts, and transmits them with the document to the printer. You do not have to exit your word processor, desktop publishing or other application that's configured for a Postscript printer. In on-line, printing is remarkably fast, up to five times faster than the laser printer's default mode.

The trade off for speed, however, is hardware requirements. To print on-line, you'll need an 80286, 80386, or 80486 computer with at least 1MB of extended memory. If you don't have the RAM, you can use Prescript's off-line mode. In this mode, you save the Postscript output to a disk file, exit the application, then print the document from the DOS prompt. Off-line mode requires just 640K of memory, and once initiated, prints faster than on-line.

For even faster printing, you can install Prescript with optional interface cards, a combination hardware and software Postscript alternative. One is a half-slot card that fits inside your computer, the other card fits into the option slot in the Hewlett Packard LaserJet II. The two are connected with phone-type RJ-11 cable.

The cards bypass the slow parallel port usually used to connect computer and printer. Data is transmitted directly to the printer's video circuitry.

One added feature is that the cards enable you to print a full page of high resolution graphics with only 512K of printer memory. Normally, you'd need to install extra memory to print full-page graphics at high resolution.

Selecting An Option

In light of the new scalable font packages (such as Bitstream and Glyphix), do you need postscript emulation like Prescript, Ultrascript, or Goscript? Both offer fonts on the fly at any size, and speed improvements over downloading entire fonts, particularly large sizes.

The Postscript interpreters offer several advantages. First, they can be used with any program that supports Postscript. You do not need special drivers or installation programs. With programs such as Facelift and Glyphix, you need an installation kit for each application for which you want to use. The installation kit for WordPerfect will not work with WORD or WordStar. You might also have a program for which no installation program exists.

Second, while Prescript works only with laser printers, GoScript and Ultrascript work with most popular dot matrix and inkjet printers, although output is certainly better on laser printers.

Still, there are advantages to scalable font programs. Packages such as Facelift and Glyphix cost much less than postscript emulation and require less hardware. If you have an application supported by the installation program, you get the same functions as postscript emulation for less cost.

Also, most application programs do not take full advantage of the benefits of Postscript. They can deal with fonts of all sizes, but not with the special effects that Postscript is capable of. There is no attribute command in WordPerfect, for example, to create fill patterns and various shadow effects with the Postscript driver. Glyphix, on the other hand, offers a number of special effect styles selected from the keyboard.

After using both types of programs, I see a place for each. When I'm very busy working with WordPerfect, I use my LaserJet's native mode with downloadable fonts. I use Glyphix for special documents when I want large fonts and special effects. While Glyphix actually prints faster than downloading, its background printing sometimes causes noticeable delays in keyboard response. So if I don't need special effects or scalable fonts, I just use the LaserJet driver.

I also use the LaserJet's native mode with other applications when I don't need a large variety or sizes of fonts. But I run the Postscript interpreter with desktop publishing programs and when I want large fonts with applications not supported by Glyphix or Facelift.

Dot Matrix Solutions

If you have a dot matrix printer and a

shoestring budget, you can still print in a variety of fonts even without using a scalable font package.

There are a number of public domain and shareware programs that generate decorative fonts in a selection of sizes on popular printers, such as the Epson FX/IBM Graphic printer and compatibles. The programs print from ASCII text files having embedded commands to select fonts, sizes, and enhancements like boldface and italic. Some of the programs can also print from specially formatted word processing files, such as WordStar and PC-Write.

While the process isn't as automatic as using a postscript emulator, they do provide a low-cost ($5 per disk) alternative for quality printing. Figure 6, for example, illustrates several fonts printed on a nine-pin Star SG-10 dot matrix printer.

You can locate these programs on most bulletin boards or public domain libraries, such as Public Brand Software. Their disk number UP10.1 includes 62 fonts and a font editing system for creating new ones. The program prints from ASCII text and WordStar document files.

Public Brand also has a two-disk set called ImagePrint (disks numbers UP21a.0 and UP21b.0) which contains 25 fonts in sizes up to 48 points. It works with ASCII text and PC-Write files that contain codes for selecting font, print density, spacing, justification, and page layout. ImagePrint also supports a wide variety of dot matrix printers.

Public Brand Software can be reached at 1-800-426-3475. ✻

Sans Serif with Bold

Roman with *Italic*

Script with Underline

Figure 6

On the Leading Edge

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

Computek Systems, Quattro Pro 3.0, Software Bridge, Outside In

This article is dedicated to solving problems of one kind or another. If you are having hardware problems and don't have any idea of where to get repair service, I have a recommended answer for that. Once the hardware problems are solved, we will look at three of my favorite application programs that can also help you solve problems. Let's begin by getting the hardware problems solved.

Computek Systems

Computek Systems is an authorized Zenith Data Systems repair center. That is important because they can get all available parts, including ROMs, for Zenith Data Systems and Heathkit computers and peripherals. Because all of the current Heathkit computers were designed by Zenith Data Systems, Computek Systems can also help you with any computer kit hardware repair problems you are having. But there is a more important reason that I recommend and use Computek Systems for repairing my computers and peripherals.

If you have read my column over the years, you may recall that I have mentioned Chris who was the service technician at the Dallas Heath/Zenith Computer Center. I mentioned in one of last year's columns that the existing Fort Worth and Dallas Heath/Zenith Computer Centers have merged into one store that is on the west side of Dallas. I'm still not sure what the new store is called, but Chris also went to the new store when it was built. And then for one reason or another, he started his own business which is called Computek Systems.

I have known Chris Ziem since 1984, and he is a superb technician. When I have had a choice, I always asked Chris to work on my computers because I thought he was the best. And I think he still is, which is the reason I trust him to work on my computers when things happen that I cannot fix. And as it turns out, Chris has another technician (Ron) who also worked at the Dallas Heath/Zenith Computer Center, so you get two experts for the price of one. Ron has worked on Heathkits for a number of years, and the last time I was in the store, a HERO robot had just been repaired. I have also seen H-89s and Z-100s in the store, so their repair expertise and experience is not limited to current-model computers and peripherals. If you are having some kind of repair problem with just about any Heathkit or Zenith Data Systems product, Chris and Ron can help you with it. Aside from the fact that both are highly competent technicians, there is even better news.

Prices. One of the most frequent complaints I get is that service provided by many of the authorized Zenith Data Systems Service Centers is not very helpful and tends to be quite expensive. Computek Systems has a current price schedule that is much more reasonable, and their service is excellent. For example, there is a flat checkout fee of $25 for any Heath or Zenith Data Systems hardware which can be applied to the repair labor within 30 days. This checkout fee includes a complete estimate of what it will take to restore the equipment (including parts and labor) to proper operation. And the labor cost is quite reasonable and is based on the CPU. The labor cost for an 8088 system (e.g., Z-151, eaZy PC, etc.) is $75. For an 80286 system (e.g., Z-248, Z-286, SupersPort 238 laptop, etc.), the labor cost is $85, and the labor cost is $105 for an 80386 system (e.g., Z-386, TurbosPort 386, etc.). If you are having problems with a Heath or Zenith Data Systems CRT, the labor cost for repairing a monochrome monitor (e.g., Z-120, Z-1220, etc.) is $45, and the labor cost for repairing ANY color monitor (e.g., Z-1380, Z-1490, Z-1492), including VGA, is $65. Please note that these prices only apply to Heathkit and Zenith Data Systems computers and peripherals. Although they may be able to help you with other equipment, depending on the brand and problem, I suggest you call or write to Chris before sending anything else to the store.

There is even a warranty that is difficult to beat: 120 days on labor and 90 days on replaced parts. And if you have any used (working or not) Zenith Data Systems equipment that you want to sell, you might want to check with Chris because Computek Systems may buy it.

Other Parts

I have received a number of letters over the past year about finding two major categories of parts for Zenith Data Systems and Heathkit computers: ROMs and lithium batteries (for the CMOS). For those of you looking for a reliable source at a reasonable price, I highly recommend Computek Systems. For example, the cost for a new ROM set will vary between $15 and $30 because there may be one or two ROM "chips", depending on what system you have.

Batteries seem to be difficult to find for many users. For that reason, I am including a partial list of some of the batteries you can buy from Computek Systems as Figure 1. If the battery you need is not listed, be sure

to check with Chris on availability and price.

In addition to the prices listed, there is also a flat $4.00 UPS shipping charge for the batteries and any ROMs that you order. They will send you the latest available ROM version. Computek Systems is highly recommended.

Quattro Pro 3.0

If you need to solve a problem, the latest version of Quattro Pro is spectacular, and it will definitely give any other spreadsheet a run for its money. Compared to version 2.0, this new version has some really helpful and extremely nice enhancements that should be helpful to nearly anyone. Perhaps it is best to begin with some of the things you can see.

Quattro Pro 3.0 has a new WYSIWYG (pronounced Wizzy-Wig) mode which allows you to see exactly what will print. That is, when you change the size or typeface of the font (e.g., italic), you can see the change on the screen. That makes it much easier to see how the spreadsheet will print although you can always use the preview feature to see how everything will be displayed on a page. It also allows you to see more columns and rows on a screen which may make it easier to work on a spreadsheet. For those of you who have worked with Microsoft Excel, Quattro Pro 3.0 now has the same display capability. But, like Excel, the WYSIWYG mode runs slower because of all the video manipulation involved, although Quattro is not nearly as slow as Excel which runs under Windows. If I don't need to see all of a spreadsheet, I generally run Quattro Pro 3.0 in the "standard" mode for speed, which can use either the EGA or VGA display mode (43 or 50 lines). And for those of you who use Windows, Quattro's INSTALL program will create a Quattro Pro icon in the Program Manager window, and it will also create a PIF file.

The WYSIWYG mode also automatically adjusts row heights to a slightly larger point size than the largest point size in each row; however, you can also manually set the height of a row with a new Style command, just in case a special kind of format is needed. With these kinds of features, you can easily perform a kind of "spreadsheet publishing" (like desktop publishing) that can create really nice spreadsheets. As I write this, neither Super-Calc nor Lotus 1-2-3 has anything like it, but they undoubtedly will as soon as they can catch up.

One thing that has always aggravated me about using a spreadsheet is the fact that I sometimes need to create LOTS of columns, which makes it difficult to judge what size font to use in order to print all columns on a single page. I recently worked on a spreadsheet that I had to reduce to 27 characters per inch (CPI) in order to get all columns to print on my LaserJet III. Even at

Part Number	Fits Models	Price
Lithium Batteries		
418-42	Z-200, Z-241/248, Z-386/16	$7.25
418-53	Z-386/25/33	15.65
418-60	Z-286	13.05
969-995	ZWL-0200-02/04	14.80
969-1763	ZL-1, ZL-2	3.80
969-1847	ZWL-0300-04/10/12	15.00
Batteries and Battery Packs		
ZA-170-1	Z-170/171	95.00
ZA-180-21	Z-181-93/97 (2.0 A/Hr.)	89.00
ZA-180-40	Z-183-92/93 (4.0 A/Hr.)	130.00
ZA-180-45	Z-181-92/93 (2.5 A/Hr.)	110.00
ZA-180-57	SupersPort 184/200 (2.5 A/Hr.)	127.00
ZA-180-65	SupersPort 184/200 (4.0 A/Hr.)	186.00
ZA-180-85	ZWL-0300-04/10/12 (4.0 A/Hr.Rapid charge)	219.00
ZA-3034-HC	TurbosPort 386 (53 W/Hr.)	179.00
150-383-1	ZA-180-51 (2.5 A/Hr.)	79.00
969-618	Z-181-92 (1.5 A/Hr.)	79.00
969-1775	ZL-1, ZL-2 (11.0 W/Hr.)	63.00

Figure 1. Computek Systems' Battery Prices.

that pitch, I ran into the Quattro Pro 2.0 limitation of 254 characters per line, so I could not get all columns printed on that spreadsheet (in landscape mode) even though there was room on the paper. New features have been added to correct both of these problems.

In particular, there is a new choice on the Print menu (Print to fit) which automatically shrinks the text in the defined print block to fit on a single page and then prints it. As if that weren't enough, you can also manually enlarge or reduce the text in the specified print block. That allows you to adjust the printing for the "best look" if you want to prepare something for a presentation. The right margin has also been expanded to print up to 511 characters, so you will not run into the 254-character limitation like I did in the previous version.

Other printing enhancements include a banner printing option, controller shading for LaserJet printers, and the capability to specify how many copies of a print block you want (the default is still 1 copy).

General spreadsheet enhancements include the Save All command which saves ALL of the in-use spreadsheets, and a switchable mouse button. The switchable mouse button feature allows you to change from the default left button to the right button, obviously a nice feature if you are left-handed. Other enhancements include more information about memory usage (conventional and expanded) and a composite status screen showing all of your saved defaults. A new command line option (/E) can be used to define how many expanded memory pages (up to 65,535) you want Quattro Pro to use, and this feature is useful if you want to reserve some expanded memory for memory-resident pro-

grams. And there is also a "progress bar" at the bottom of the spreadsheet screen which shows the time remaining for some operations, such as loading and saving large files.

If you use Quattro Pro to create graphs or slide shows, there are some enhancements that will help you develop very professional-looking graphs and slide shows. For slide shows, you can choose various techniques (called transitions) for changing from one slide to another, such as fade, single- and double-edged horizontal and vertical wipes, spiral, and dissolve with various degrees of disintegration. You can also add sound effects to slide shows and macros by using SND files that are included in the package. For graphs, you can add "sculptured" outlines to graph titles and legends, as well as to the graph itself. For text within a graph, you can add a drop shadow in COLOR to spruce up the presentation.

As I have said before, I think Quattro Pro is the best spreadsheet value around, and all of these new features make it even better than most, especially Lotus 1-2-3. Quattro Pro works with the Lotus 1-2-3 file format and is even compatible if you use macros. Of course, Quattro Pro 3.0 offers some macro features that are better than Lotus 1-2-3, and you can even use the Lotus-compatible menu structure if you are more comfortable with that. I prefer the standard Quattro menu structure because it is better organized and easier to use.

Even the upgrade cost for Quattro is better. The last time I saw a Lotus upgrade form, the cost of the upgrade was $150. When I upgraded from the original Quattro Pro (version 1.0) to version 2.0, the cost was about $99.00. And this most recent upgrade was about $49 apparently be-

cause it was released so soon after the 2.0 upgrade, at least that is what the upgrade notice said. If you want to upgrade from version 1.0 to 3.0, I understand the cost is still only about $99.

Quattro Pro 3.0 requires at least 512 KB of RAM, and 640 KB is recommended. A hard drive with at least 3 MB of free space is required and 4 MB is recommended. Virtually any video card will work from CGA to VGA, as well as an 8514-compatible card. Expanded memory is supported and will work with either EMS 3.2 or 4.0. A mouse is also supported, and I use one occasionally for my work. All kinds of popular printers are supported, and Quattro Pro automatically detects and uses a numeric coprocessor if available. Borland is well known for its high quality software, and Quattro Pro 3.0 is highly recommended.

Converting Files

I mentioned at the beginning of this article that I would discuss software that helps you solve a problem. Because I work with a lot of different computers that are located at many different companies, one problem that I always run into is that of different word processor and spreadsheet programs. And because word processors and spreadsheets use different file formats, I somehow have to bring everything together in a single format when I compile a final report.

Sometimes the problem is relatively simple when I just need to convert a file format from one word processor to another. Other times, the problem is more complex, especially if I want to select part of a spreadsheet to integrate into a report in a word processor. I usually find that I end up with a real hodgepodge of files that are created by some off-beat word processors like Samna IV, Display Write, and PFS Write.

One of my friends maintains that I could solve the problem by using Windows' applications, but he does not understand the problem. That might work fine if I always had a CHOICE, but I don't. I have to use whatever a client wants and has available, and I have not found any corporate user who has spent all the money required to run Windows' applications, including both software and hardware upgrades. So I still need to bring everything together into one file format. I have used one utility program for several years to do that, and I have recently found another that is just as helpful if not more so. As it turns out, both are available from the same vendor: Systems Compatibility Corporation.

Software Bridge 5.0

I have mentioned Software Bridge before in this column, and the latest version has all kinds of new conversions. I originally bought Software Bridge because of the need to easily convert from one word processor file format to another, but this version has all kinds of additional conversions for spreadsheets and data bases.

For documents created with a word processor, Software Bridge will generally convert most, if not all, of the formatting that you have used in the "old" document. Even though some conversions are less than perfect because not all of the formatting can be preserved (because one of the word processors may not have as many formatting features as another), Software Bridge can save lots of formatting time as compared to exporting a file to ASCII from one program and then importing and reformatting it with another.

Document conversions include: DEC WPS Plus, Display Write, First Choice, IBM Writing Assistant, MASS-11, DOS-based Microsoft Word (and RTF), MultiMate, Nota Bene, PFS:Write, Professional Write, Q & A, Samna Word, SmartWare II, Sprint, Volkswriter, Wang PC, WordMARC, Word Perfect, WordStar, WordStar 2000, and XyWrite. You can also convert to: ASCII, DCA/FFT (IBM's Document Content Architecture/Final Form Text), DCA/RFT (IBM's Document Content Architecture/Revisable Form Text), Intelligent ASCII (using spaces to replicate the "look" of centering, indents, margins, and tabs), and the Navy DIF (Document Interchange Format) files. And if you need to convert files created on a Macintosh (using a DOS-formatted disk of course), you can also convert the Mac versions of Microsoft Word, Word Perfect, and MacWrite II.

In most cases, you can convert files either way from any listed file format, but keep in mind that how good the conversion is really depends on what you are converting from and to. If you convert a very large file created with Word Perfect, for example, to Samna, Software Bridge will automatically split the original document into multiple, converted files because Samna has a limit on the number of format lines, pages, and so on. Even though the conversion is quite good, a word processor's limitations can frequently introduce all kinds of interesting situations. In some cases, I have found that it is easier to convert a document to DCA/RFT before importing it to a new program, especially one like Samna that has specific limitations. As you can probably tell, I have worked with the Samna conversion more than a few times. And if your word processor is not listed or cannot export to one of the listed file formats, you probably should consider getting a new word processor.

As you might expect, spreadsheet and data base conversions can only work one way: from the spreadsheet or data base file format to a word processor file format. Spreadsheet conversions include: Enable, Framework, Lotus 1-2-3, Lotus Symphony, Microsoft Excel, Microsoft Works, Mosaic Twin, PFS: Professional Plan, Quattro, SuperCalc 5, SmartWare II, and VP Planner 3D. Data base conversions include: dBase, Data Ease, dBXL, Enable, FoxBase, Framework, Microsoft Works, Paradox, and SmartWare II.

One of Software Bridge's interesting features for both spreadsheet and data base conversions is the capability to convert SELECTED parts of a file. That is, you don't have to convert the entire file (which can be extremely large, especially for a data base). There are three basic selection modes. First, you can specify a number of lines/rows or data base records. Second, you can specify a horizontal or vertical (for columns or tables) block of data. And third, you can specify a non-contiguous range of rows and columns which allows you to pick and choose what you want to convert. You should be able to find an easy way to convert spreadsheet and data base files to your word processor using one or more of these selection techniques.

Another very nice feature is the capability to view a file. That may not sound like much until you try to use the TYPE command to get an idea of what a file's contents are and then find out the TYPE command will not work with some word processor's file formats like Samna and newer versions of Word Perfect. Samna places a CTRL-Z character at the BEGINNING of a file for some reason, and the TYPE command stops when it reaches the CTRL-Z. So, you can't see a Samna file with the TYPE command. From what I can tell, it looks like Word Perfect 5.1 does the same thing. But that does not stop the file viewer supplied with Software Bridge. Then, when you find the file you want, you can translate it to your word processor. This particular feature makes it easy to find the file, and I use it a lot.

Software Bridge requires a minimum of 384 K of RAM and a hard drive. The hardware requirements are minimal as you would expect with this kind of program. If you need to convert files from one format to another, Software Bridge is highly recommended, but there is another option.

Outside In 1.1

Systems Compatibility Corporation has another excellent product that is a nice complement to Software Bridge. Outside In is a memory-resident program that provides the capability to view AND COPY a file directly into your word processor without leaving it. Because Outside In is memory-resident, you can import information from another word processor, spreadsheet or data base file without having to exit and perform the file translation. If you work with as many file formats as I do, that is a real advantage because it really cuts the time required to assemble a document in a single file. And like Software Bridge, you

can view a file and select whatever you want BEFORE you translate it. In fact, the basic Outside In viewing screen is virtually identical to that included with Software Bridge, which was no surprise to me.

This version of Outside In includes all of the conversion capabilities listed for Software Bridge plus Ami and Ami Professional. Spreadsheet and data base conversions are supported too. I have found Outside In is especially useful for viewing files, regardless of what software created them. The biggest difference in the viewing capability between Software Bridge and Outside In is that the viewer for Software Bridge is NOT memory-resident. Outside In IS memory-resident, but it otherwise looks like the same program to me.

Outside In requires 70 KB of RAM, and that's it. It is a small program that comes with one 3.5-inch and one 5.25-inch floppy distribution disk, so it would probably work on just about any PC compatible system. If you need this kind of program, Outside In is highly recommended.

I bought both programs because I received an upgrade card for Software Bridge with a special deal on both. And I have found both are quite useful for their specific applications. Which one I use of course depends on exactly what I am doing, but either one may be sufficient for most users. If I had to choose, I would probably pick Software Bridge first, only because I usually need to translate complete files quickly.

Powering Down

If you have any hardware problems that need to be fixed, I think you will find Computek Systems can help you with virtually all Heath and Zenith Data Systems equipment. They also can help if you want to upgrade your system (e.g., hard drives, video cards, etc.) because they understand Zenith Data Systems computers. Consider writing or giving them a call if you need to get your equipment repaired or need some prices for an upgrade.

I hope the programs I've mentioned will help you solve a few problems you might have. Quattro Pro 3.0 is certainly the best spreadsheet program I've seen, and I've seen and worked with all of the major ones. Software Bridge and Outside In are the best file translation programs I've found, and if you need that kind of application, both are highly recommended.

For help in solving specific computer problems, be sure to include the exact model number of your system (from the back of the unit or series from the Owner's Manual), the ROM version you are using (use CTRL-ALT-INS to find it, except for the eaZy PC), the DOS version you are using (including both version and BIOS numbers from the VER command), and a list of ALL hardware add-ons (including brand and model number) installed in your computer.

The list of hardware add-ons should specifically include memory capacity (either added to an existing board or on any add-on board), all other internal add-on boards (e.g., modem, bus mouse or video card), the brand and model of the CRT monitor you have, and the brand and model of the printer with the type of interface (i.e., serial or parallel) you are using. Also be sure to include a listing of the contents of the AUTOEXEC.BAT and CONFIG.SYS files unless you have thoroughly checked them out for potential problems (e.g., TSR conflicts). If the problem involves any application software, be sure to include the name and version number of the program you are running when the problem appears.

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

Products Discussed

Software
Powering Up (885-4604) $12.00
Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3463 (ZUG Software only)

Quattro Pro 3.0 $247.50
Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066
(800) 255-8008 (Except California)
(800) 742-1133 (California only)
(800) 237-1136 (Canada only)

Software Bridge $149.00
Outside In 99.00
Systems Compatibility Corporation
401 N. Wabash, Suite 600
Chicago, IL 60611
(312) 329-0700 ✄

"MAY I SEE YOUR ARTIFICIAL DISPLAYS?"