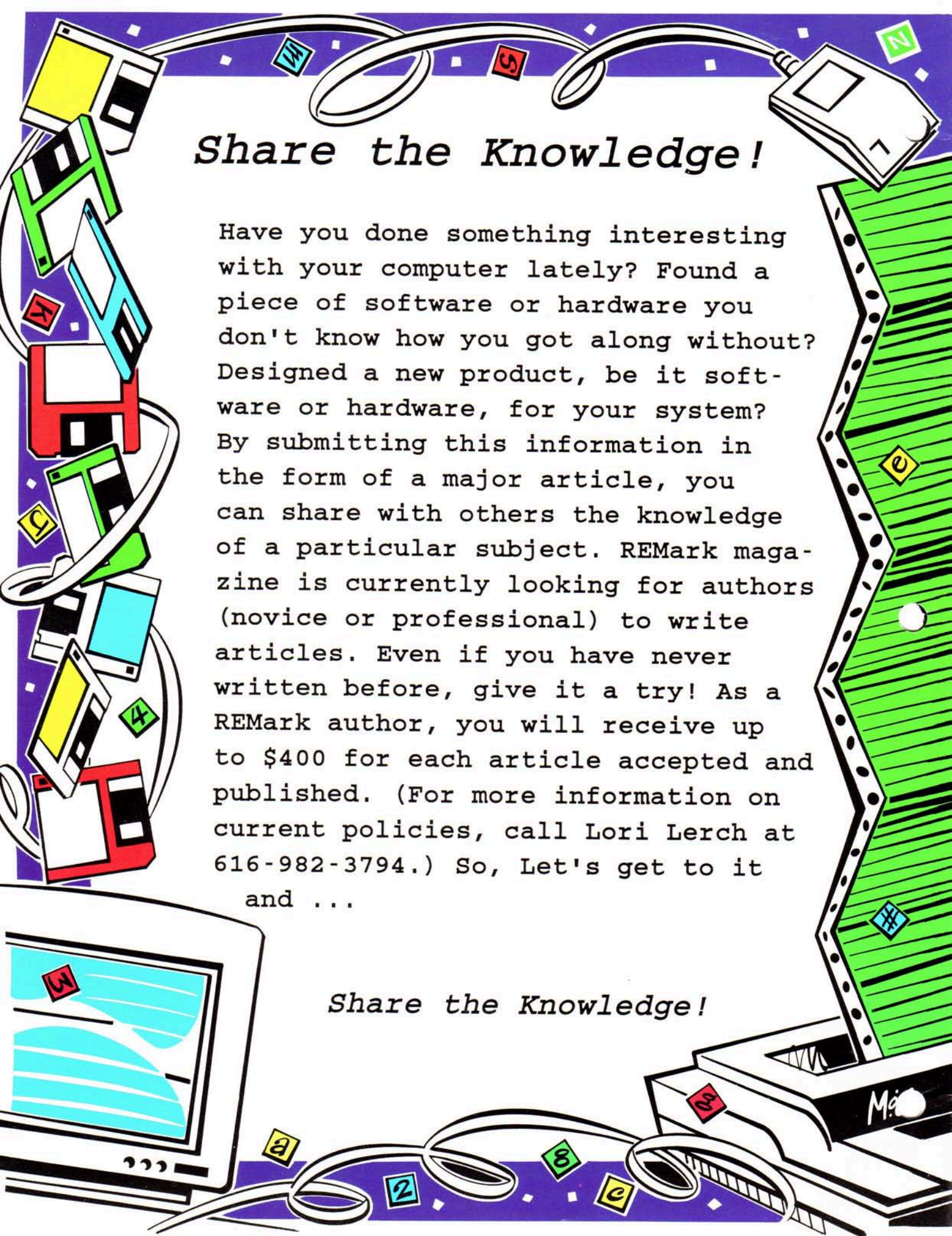
REMark®

May 1992

The Official Zenith Data Systems Computer Users' Magazine

In This Issue ...

Learn About Hard Disk Drives



Share the Knowledge!

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

Share the Knowledge!

May 1992

REMark®

The Official Zenith Data Systems Users Magazine

Advertising

Resources

Software

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

Software Order Form

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

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

Sub-Total ____

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

Total ____

Name ____

Address ____

City, State, Zip ____

Credit Card # ____

Expiration Date ____

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

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

Managing Editor
Jim Buszkiewicz
(616) 982-3837

Software Engineer
Pat Swayne
(616) 982-3463

Production Coordinator
Lori Lerch
(616) 982-3794

Secretary
Lisa Cobb
(616) 982-3463

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

ZUG Software Orders
(616) 982-3463

Contributing Editor
William M. Adney

Printer
Imperial Printing
St. Joseph, MI

Contributing Editor
Robert C. Brenner

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

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

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

* U.S. Funds

Limited back issues are available at $2.50. Check ZUG Product List for availability of bound volumes of past issues. Requests for magazines mailed to foreign countries should specify mailing method and include appropriate, additional cost.

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

ZENITH
data systems
Groupe Bull

REMark Magazine Subscription & ZDS-COM1 Bulletin Board Information

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

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

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

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

Zenith Data Systems Users' Group
P.O. Box 217
Benton Harbor, MI 49023-0217
(616) 982-3463 — Voice
(616) 982-3956 — BBS

Example:
First Name: John
Last Name: Smith, Jr.
Password: ******

Registration on the COM1 BBS is NOT automatic. Once you receive your magazine, check your mailing label for your ID number and call us to request COM1 BBS access (or you may call to receive your ID number and access to the board immediately). All you need to do is enter your first name and last name EXACTLY as they appear on your REMark mailing label (including all spaces and punctuation, see Example), and then enter your ID number as your password.

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

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

REMark Subscription / Renewal Form

New Member: ☐ Yes ☐ No

Credit Card # _____

ID Number: _____

Exp. Date _____

Address Change? ☐

Name: _____

Address: _____

City, State, Zip: _____

Daytime Phone #: ( ) _____

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

BOOTLOG

A Start-Up Trace Utility for MS-DOS

Lawrence W. Conklin
105 Riverglen Road
Liverpool, NY 13090

Have you ever wondered whether someone has been surreptitiously messing around with your computer? Do you worry that someone might be browsing through your files, pirating your software or even installing a virus? I work in a large office and have no practical way to protect my machine. I am not particularly paranoid about someone using it and I do not have a lot of sensitive files on my disk, but I am uncomfortable with the possibility that someone might be using the machine without my permission. On at least one occasion I have found files missing or damaged by persons unknown. From the nature of the damage it was apparent that whoever it was had been copying files.

After that experience, I decided to write a program that would detect unauthorized use of my machine without tipping off the intruder that anything unusual was going on. BOOTLOG maintains a trace file that records the date and time every time the computer is turned on or rebooted. The program expects the first command entered to be a password. Anything else is recorded in the startup trace file as a password violation.

BOOTLOG is designed to run invisibly from the AUTOEXEC.BAT file. Both the program itself and the file it maintains are tagged with the MS-DOS hidden file attribute so that they will not show up in a directory listing. When the program is executed, it displays a dummy command prompt and waits for input. If the expected password is entered, BOOTLOG simply records the system date and time in the trace file and exits. If the entry is not the expected password, the date/time record is annotated with a "Password Violation" message and a copy of the command that was attempted. The command is then passed through to the operating system for normal execution. Nothing happens that reveals that anything unusual is going on. Unless an intruder inspects your AUTOEXEC.BAT file or searches for hidden files, it is unlikely that he will realize that the program exists.

I wrote BOOTLOG in Microsoft QuickC version 2.0, but there is nothing in the program that is likely to cause problems if you have a different compiler. After opening the trace file and allocating space to receive the command entry, the program calls time() to obtain the current date and time. The time() function returns the number of seconds since midnight, January 1, 1970. Not a particularly user friendly format. A call to localtime() converts this value into a structure that contains integer values for the month, day, and year and for the number of hours and minutes since midnight, corrected for your local time zone. BOOTLOG converts the time from 24 hour to 12 hour format and calls asctime() to convert the time to an ASCII text string. The date/time string is displayed on the console and written to the trace file.

The program then displays a dummy prompt and waits for input. On my system I have included a PROMPT command in the AUTOEXEC file that causes the initial prompt to appear as "C:\ ->". You should change the printf() statement to display a prompt that is identical to whatever the operating system displays when you are in your boot up directory. Notice that you have to double the backslash character so that the C compiler will print the string properly. A single backslash is otherwise interpreted as an escape character.

When an entry is made, BOOTLOG compares it with the expected password. You should change the #define PASSWORD statement to whatever you want your password to be. The program does not suppress the output of the password entry on the screen the way passwords are normally handled. Your unwary intruder doesn't know he's entering a password and you don't want to arouse suspicion by failing to echo his entry. You also don't want to compromise your password, so it is wise to avoid letting anyone observe you when you enter it.

If the correct password was entered, a "\n" character is written to the trace file to terminate the log entry, and the program cleans up and exits. If the entry does not match the expected password, the "Password violation" message is appended to the date time string in the trace file, along with a copy of the command that was attempted. The system() function is then called to pass the command that was entered to the operating system for execution.

The scanf() statement uses a rather strange looking format argument for reading the password. If you use the simple %s format, the first white space character terminates the scan. That's ok for reading a simple password, but it creates problems when an invalid command is entered that has one or more arguments. For example, suppose the attempted command is something like CD \BIN, to change to the \BIN directory. The \BIN argument is not read or passed through to the operating system, and the result is an error message instead of the expected directory change. The "%[^\n]" accepts everything entered up to the carriage return that terminates the entry and assigns it to the password string. When the string is passed through as a command, the arguments are included and the command is processed normally.

Once BOOTLOG is included in your AUTOEXEC file, every system boot-up will be recorded in the trace file. I make it a habit to read the trace file every morning and sure enough, I occasionally find that someone has been there. Every week or so it's a good idea to delete the old records from the trace file. Most editors will read a hidden file, although in my case, I have to reset the hidden attribute after I edit the file.

BOOTLOG is not intended to be the ultimate in system security, but it is effective for detecting unauthorized use. My original intent was to write something that would require a password to gain access to the machine, but that is a much more difficult problem. A knowledgeable intruder could simply boot off a floppy disk, bypassing any password scheme run from the AUTOEXEC file on a hard disk. Of course, that would circumvent BOOTLOG too, but because it runs transparently there is nothing to tip off an intruder that there is anything to be wary of. One thing that can be done to hide the program more effectively is to rename it something like SHOWTIME since it displays the date and time on the console.

I have been thinking about a couple of enhancements to this simple program. It would be much more interesting to record a trace of all of the commands that were run following an unauthorized boot-up. That would let you know what the intruder had been up to, and might help you determine who it was. I haven't figured out how this might be done, but it seems as though the program would have to be written as a TSR. Every time a command was run or a program completed execution, control would have to be returned to the TSR to display another prompt and record whatever the next command was.

I didn't make any provision for changing the password, other than by recompiling the program. It would be easy to add a procedure to accept a new password, and it could be stored in the trace file. However, I didn't think it was necessary to change the password very often. After all, your intruder shouldn't be aware that there is a password at all.

Since I installed this program on my office machine, I have occasionally detected unauthorized use, but no malicious intent. I have so far been able to determine who had been using the machine, and it has always been someone with a legitimate reason. If you need really effective security for a system that stores important or private information, BOOTLOG isn't the answer. One possibility would be to try either version 5 of MS-DOS or the new Digital Research DR DOS, version 6. Both operating systems have some support for password protection of your files, and DR DOS provides very complete multi-level password security for your entire system.

```
/*
        BOOTLOG.C - Reads the current date and time from the operating
                    system and records them in a file BOOT.LOG.
                    Intended to be invoked from the AUTOEXEC.BAT file
                    to detect if someone is clandestinely using the
                    machine.  The time is also output on the console.
*/

#include <stdio.h>
#include <time.h>
#include <string.h>
#include <malloc.h>
#include <process.h>

#define PASSWORD "yourpasswrd"

void main(void)
{
    time_t raw_time;
    struct tm *time_now;
    char *am_pm, *password;
    FILE *logfile;


    logfile = fopen("BOOT.LOG", "a");       /* Open log-on trace file  */
    password = (char *) calloc(80, 1);      /* buffer for command line */

    time(&raw_time);
    time_now = localtime(&raw_time);        /* get current time of day */

    if (time_now->tm_hour < 12)
        am_pm = "AM";
    else
        am_pm = "PM";

    if (time_now->tm_hour > 12)             /* convert to 12 hr format */
        time_now->tm_hour -= 12;

    /* display date/time string on console                             */
    printf("\n%.19s %s\n\n", asctime(time_now), am_pm);

    /* record log-on time regardless of whether entry was password */
    fprintf(logfile, "%.19s %s", asctime(time_now), am_pm);

    printf("C:\\ ->");                      /* display dummy prompt    */
    scanf("%[^\n]", password);              /* wait for first entry    */

    if (strcmp(password, PASSWORD))         /* if not password, record */
    {                                       /* the attempted command   */
        fprintf(logfile, "  Password violation - %s\n", password);
        fclose(logfile);
        system(password);           /* then pass thru to DOS   */
        free(password);
    }
    else                                /* if entry was password,  */
    {                                       /* clean-up and exit       */
        fprintf(logfile, "\n");
        fclose(logfile);
        free(password);
    }
}
```

Quibbles 'n Kits 'n Bits 'n Fits

Laszlo Vesei
420 Philip Road
Niles, MI 49120

The palindromic year of 1991 saw three of my articles in REMark: Search for the Perfect Word Processor (May); Dis an' Dat About Printers (July); and All Around the eaZy PC (October). Reading these over again the realization hit me hard (right between the horns) that I have left out quite a lot of stuff! The main purpose of this writing is therefore (as the title so aptly indicates) to fill in these gaps.

As an addict of BBSs in general, and COM1 in particular, I am aware of some of the problems our fellow members have. Sometimes I send messages to answer these. However, not all the ZUG (HUG?) members take advantage of COM1 (and what an advantage it is!) so I include here some notes about stuff not germane to the aforementioned articles.

Methinks it would be best to look at the Word Processor and Printer article together. (Ya know: horse and carriage – love and marriage. I wish I knew which is which).

These two articles generated some correspondence. One of my correspondents just purchased a Big Name word processor and was frustrated and extremely angry at it. On the strength of the REMark article he purchased TEXTRA and was so happy with it that he felt he owed me a letter. He stated that TEXTRA worked on his H-100 without any alteration, function keys and all.

Another correspondent (please, forget the lawyer jokes on COM1) wanted to write a document in small print. He tried the superscript option but the printed text just did not look right. The lines were too far apart. Well, the typefaces can be edited, that is, if your printer is able to accept it. Here is how to do it with TEXTRA: Put TEXTRA into "Shift F7 (Print)" mode. Next select "U" (printer setup), hit "T" (change font) several times, until the desired typeface shows up for your small print. (SansSerif is the most legible of the small fonts, but then legibility may NOT be your objective. Lawyer jokes, you know). When done with the selection, hit <ESC> several times until you are back in your document. If you now hit <F5> and select from the menu either Superscript or Subscript, you will get the small print, but the lines will be too far apart. However, this too can be fixed. Hit <F6> (page preview) then "P", "C" and "S". The parameter you want to change is called "leading" (pronounced ledding). The default is 100%. Change it to something less, say 70%. Hit <ESC> repeatedly until you are back in your document. Insurancemen and lawyers will love it.

In my July '91 article about the PANASONIC KX-P1124 printer there was a Basic program to print ALL the ASCII characters, including the IBM-specific ones. There must be two remarks with regards to this:

1. To print ALL these characters (including the low ones) you have to put the KX-P1124 into a special mode before the BASIC program, as described in the article, will run correctly.
2. Most word processors and printers cannot accept the low ones (below decimal 32), except maybe #21 (§). In fact, some of the expensive ones cannot even accept some of the high ones, as they use *these* for printer commands. (Phooey, Multimate!)

I have written another BASIC program that will print out the printable ones of these characters without the necessity of reconfiguring the printer. Unfortunately, the low ones are unprintable as they are the printer commands in this mode. See Listing 1.

A page, full of these characters often comes in very handy. I keep one in a transparent plastic sleeve at my keyboard.

Undoubtedly, Zenith Data Systems converses with Mother Bull (now, this sounds a bit peculiar, n'est-ce pas?) in Française. I myself write much of my correspondence in German, French or Hungarian. These languages have some special letters, foreign to English. The thing to do is to create macros to produce these letters. With one such macro, for instance, the letter É could be produced by holding down the <ALT> key while hitting <e>, or the letter é by Ctrl <e>, with another the German ß (Ess-Zett) by <ALT><Z>, and so on. Naturally, these, as well as the "framing" characters, can always be produced by holding down <ALT> and entering the *decimal* numeral associated with the desired character *on the keypad* (not on the upper row of numerals!) It is not even necessary to put the keypad into the 'Num Lock' mode, it works either way. This is where the already mentioned cheat-sheet comes in handy. A word of warning: the previously mentioned lower (IBM) characters (lower than decimal 32) can be put on the screen by the <ALT>+numeral method (at least on my H-386/16, I don't know about your machine), but either they will not print, or they will confuse your printer.

The Old Gray Mare

The ol' gray mare ain't what she used to be, thank heavens. It used to be an H-161 and provided yeoman duty for years. One of my arms is still several inches longer than the other one, in way of proof.

It came equipped with two half-height 5¼" drives. To that I later installed a 3½" drive into the cable storage space provided in the lid. Dante Bencivengo's speed-up kit came next, with the reset button, speed selector switch and indicator LED added to the front, just above the sliding door. The keyboard had to be modified to accommodate the sticking-out things, by adding holes and filing openings. The keyboard had a rather stiff and heavy cable. It was promptly discarded and a new one fabricated from a

6-conductor flat telephone cable and two DIN plugs.

Now, at the fast setting, this machine showed a very strange behavior: after about 10-15 minutes it hung. Various tests showed that:

1. The power unit was not adequate. A larger one was installed. Still a ZDS power unit, same physical size but electrically more powerful.
2. The CPU board, which was in the second slot from the power unit, developed excessive heat. The largest offenders were the CPU, the Math coprocessor and the bus controller. The card was moved to the first slot necessitating enlarging the rear opening for the keyboard plug. Here the power unit blower blows directly on it. The Video Logic Board was moved to the second slot. It was extremely difficult to insert it there, because there was no cutout for the rear connectors. It was necessary to bend it into an arc – not a recommended practice! – before it could straighten out and go into the socket. It survived the ordeal, though.

This cured the heat problem.

At this time I wanted to try out Multimate Advantage (a rather expensive but muddleheaded word processor). Unfortunately, the original issue RAM memory was not sufficient. FBE Research (an advertiser in REMark) came to the rescue with a new PAL chip. After replacing the original 64 K RAM chips to 256 K ones, I ended up with 640 K regular memory and 512 K extra, which I used as a virtual disk drive. (It is worth mentioning that FBE Research furnishes a short cable with mechanical clips on the ends to avoid soldering. I choose a short wire soldered in place instead.)

Loading a rather long file into the virtual disk caused a "Parity Error" and everything seemed to quit. Obviously one of the memory chips failed, but which one? In this case the diagnostic program supplied with the machine was useless. I tried to load short ASCII files into the virtual disk, keeping track of the total. After each file was loaded I tried to read it back by the <TYPE> command until the Parity Error manifested itself again. A little mental arithmetic revealed the bank of memory chips where the bad one was residing and, substituting one-by-one, the bad chip was found and replaced. An extension card, having a card-edge connector on one end and connector pads on the other is very handy for servicing these machines. It plugs into the backplane and the suspect card plugs into it, holding it accessible for manipulation.

Since this machine was a (trans)portable one, I thought that addition of a hard disk was not advisable.

According to an old Hungarian proverb "the ewer keeps going to the well until it breaks". My '161 broke too. Specifically, the plastic struts, holding the lid up. Since these were no longer available from any known source, drastic measures had to be taken. No known glue or cement would hold them in one piece for any length of time. As first aid I used to carry a small box (shades of Mark Twain!) as a prop-up. Then I saw someone advertising an empty '151 desktop cabinet. The '161 was moved into this cabinet and became a desktop. Two 3½" drives were installed in the cabinet at this time. The original floppy controller card is capable of handling 4 floppy drives, so the 5¼" drives moved into their own apartment. It was furnished with its own power unit and was connected to the second connector of the disk controller card by a long ribbon cable thru a slot left open in the rear. As expected, the DIP switches on the CPU board had to be set for four floppy drives. Also, the two remote drives had to be set as "drive 0" and "drive 1" on their own circuit board, although the machine recognizes them as drives "C" and "D". Also a cheap color monitor was acquired.

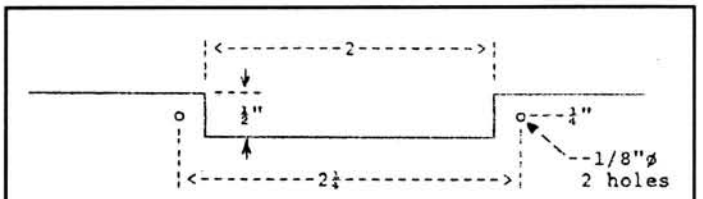


Figure 1
Slot

For good measure a 20 Meg *MFM* hard drive was also installed, along with a Western Digital *RLL* controller card. Contrary to general belief, the hard drive not only worked with this card reliably, but also provided 32 M disk space reliably. After the SPINRITE treatment it was very fast, too. I suspect a cache of some kind on the controller card.

This setup served me well for a couple of years. However, the spirit is restless (as a native of Hungary I may have some Gipsy blood in my veins). A sale of ZCM-1490 monitors on the Bargain Centre seemed irresistible. This monitor did not work with the hardware I had at that time, of course. So a new video card (a Zenith Data Systems Z-449) had to be installed and the old one removed. Since the original issue video card contained the scratchpad memory necessary for proper functioning of the CPU, the latter had to be modified. FBE Research came to the rescue again with a daughter board for the CPU card. At the same time the FBE SmartWatch was also installed under one of the ROM chips. The Old Gray Mare is still in use as my backup should my H-386/16 bite the dust.

The eaZy PC — Revisited

Wonders will never cease. While this is the article that generated most of the correspondence, not a single person queried the hard drive adapter, as shown in the article's illustration. (The print, if you review it, was incomplete.) Maybe these hard drives are extremely difficult to get, or if you finally hunt one down, it comes with the adapter. Be it as it may, if anyone wishes to have a complete print, write to me. I will be happy to provide it.

In my article I mentioned that any eaZy PC having a built-in hard drive can be equipped with an *external* floppy drive. The eaZiest way to do it is to chop a hole in the plastic cover and plug a piece of 34-conductor extension ribbon cable into the floppy #2 end of the floppy cable and stick it thru this hole. The other end of this extension cable goes to the external floppy drive, which might be either a 5¼" or a 3½" drive *with its own power source*. The system ROM does not support high-density drives, though.

Before we progress any further with installing an external floppy drive a word of warning: This thing (or most external things you may add to your computer) is *"polizeiwiedrig"*, meaning the Federal Communications Commission. If you live in a condo or apartment house, or the neighbor's home is too close, this thing may cause problems with his/her radio or TV reception. The FCC takes a dim view of this disturbance. Of course, if you live out in the boondocks it is of no concern.

So far so good. But what about those *les miserables* whose internal floppy cable ends at floppy #1? Do not despair. No situation is without solution. The first thing to do is to take a good look at the ribbon cable leading to floppy #1. There are 2 kinds: one goes straight, the other makes a 180° turn. Observe and write down which one you have.

The next thing to do is to make a new internal ribbon cable. First you will have to get a 34-conductor ribbon cable. The length depends on how you intend to handle the situation. The sloppy way is to stick it thru the slot you produce somewhere in the plastic cover, either at the ventilation slots or at the rear, where the metal pan meets the plastic cover. In these cases your ribbon cable must be long enough to cover the distance from the eaZy PC to your external floppy drive, say about 36". Of course, this way you will have a permanent 'tail' sticking out of your machine. This 'tail' will receive a special treatment, which will be described later. Some people, me included, find such a tail objectionable for esthetic

reasons. Chacun à son goût.

The more elegant way is to file a larger slot into the metal pan and use a 34-pin connector bolted to the pan itself. Figure 1 shows the size of the slot in the metal pan. It is recommended that, to keep the filings out of the electronics, remove everything from the pan. To make sure that everything goes back the right way make sketches and notes. Your (and mine) memory is not always reliable. Me, I am lazy, so I covered up things with rags and carefully shook out the filings afterwards. Chacun à son goût, again. In this case you will need to make up two ribbon cables: one internal in the eaZy PC, the other from the rear connector you installed in your machine to the external floppy drive. Figure 2 shows the ribbon cable internal in the eaZy PC. For this internal cable you will need about 18" of 34-conductor ribbon cable. This cable will receive a special treatment, because Zenith Data chose an unusual way to address the floppies. See Figure 2.

First, the materials:

As mentioned already you will need 36" (3 ft) of 34-wire ribbon cable. If you choose the eaZy way, leave it in one piece. If you do not like the tail hanging out of your machine, cut it into two 18" pieces.

For sources of the materials listed below, please look up my article in the October 1991 issue of REMark. As for the materials, we will split the list into two for the two different methods:

A. The laZy (long tail) method. Parts list:
   1 piece 36" (or a few inches more) 34-conductor ribbon cable.
   One connector for the external floppy drive. It might be either a card edge connector, AMP N°111110-3 or a socket connector, AMP N°499997-8, depending on your external floppy drive.
   2 socket connectors, AMP N°499997-8
   For the assembly of this cable see Figure 2.
B. The 'elegant' method. Parts list:
   2 18" pieces of 34-conductor ribbon cable.
   1 connector for the external floppy drive, as above.
   3 socket connectors, AMP N°499997-8
   1 straight long tail wire wrap connector, AMP N°102158-8

If you happen to have a complete internal (2 floppies) cable then you are lucky, because you will need only one 18" cable. Also, you can disregard the next paragraphs describing the construction of this cable, since you have it already.

The first cable, (or in the laZy method the only cable) receives a socket connector at one end. The second socket connector will be located at the same distance from this end as the old internal floppy cable was. You will have to fold this part of the cable the same way too.

The other end of the cable is a bit tricky. The following will describe this special treatment, which will be necessary for both, the internal cable and the 'long tail' cable as well. See Figure 2.

Start at the colored edge of the cable and count the wires. Take a sharp knife and make a slit between wires 2 and 3 about 4" long. Likewise between 3 and 4. Also between 9 and 10 and between 12 and 13. The last two slits will be located between 15 and 16, also 16 and 17. The next step will be easier if you stick a piece of double-scotch tape (sticky at both sides) to your tabletop. Now stick the slit end of the cable to the tape. Take the 3-wire part (wires # 10, 11 and 12) and turn it upside down, so that # 10 and 12 changed places. Stick it to the tape this way. Now move wire # 2 to the # 16 spot and wire # 16 to the #2 place. Stick them to the tape this way. Secure the cable end with a piece of ordinary masking tape across the cable before you lift it from the table. Now secure the other side with masking tape also. If necessary cut the end of the cable even with a pair of sturdy scissors. Lastly, crimp a socket connector (or for the laZy method if your external floppy drive so requires, a card edge connector) on this end of the cable. A picture is worth 1000 words, take a good look at Figure 2.

The other 18" cable remains straight. One end gets the socket connector, the other end the connector for the floppy drive.

Now a word for those, who already have a two-floppy cable and are of laZy disposition: for the 18" cable substitute a pin connector AMP N°746492-8 for one of the socket connectors. This will be plugged into the end of the existing floppy cable. The other end of the 18" cable (in this case it might be longer, too) must still receive the proper connector for your external floppy.

For those who chose the 'elegant' way: now is the time to install the long-tail connector. First you will have to produce the slot in the bottom pan of your eaZy PC, see Figure 1. The location of this slot is not too important. If you have a cooling fan, you can remove it. No ill effects were reported. This is a good location then for the slot. If you wish to retain the fan the slot must go above the "MOUSE" port. In this case – assuming you wish to reuse the original two-floppy cable that came with the machine – you are out of luck. It will be too short. So it is either remove the fan or construct a new, longer (18") cable.

The long-tail pin connector has 34 pins in two 17-pin rows. One row of these pins will be all connected together and eventually connected to the ground. It is easier to solder the necessary wire to

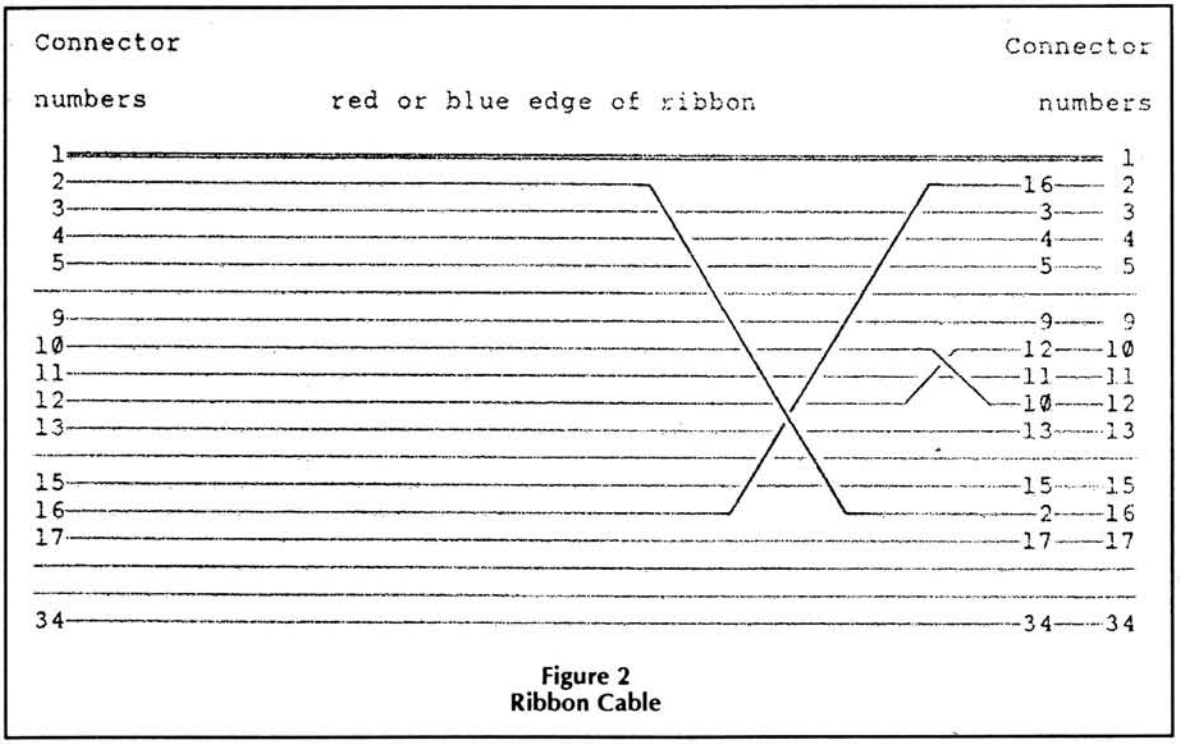


Figure 2
Ribbon Cable

these pins before the connector is fastened to the case. Now install the connector into the slot so that the longer pins will be inside the pan and the grounded pins at the bottom. Fasten the connector to the pan with two #4 screws and nuts. Connect the grounded pins to the ground at any convenient location inside the pan.

Next plug the socket connector of the internal floppy cable onto the long pins of this connector so that only the (ungrounded) top row of pins is used, the colored side of the cable is toward the main circuit board and the bottom row of the socket connector is plugged into the top row of the pin connector. It looks odd that only one half of the socket connector is plugged in, but believe me it is the only way it will work. If you use the original floppy cable, it is barely long enough to plug into the long tail connector. A short tail connector would not reach. The only thing remains to apply a paint dot outside the pan to indicate the colored side of the cables. Magic marker will do fine.

Before you reassemble the machine do not forget to remove the jumper from the W-6 position. The absence of this jumper will tell the eaZy PC that you have two floppies now.

If your external drive is another 3½" drive, all is well. It will be recognized as Drive B, while the hard drive remains Drive C. If it happens to be a 5¼" drive you will have to run DSKSETUP to notify your machine about this fact.

By the way, in my previous article I recommended MS-DOS (Zenith Data Systems) 3.21 as operating system. I have had bad luck with newer ones. However, Zenith Data Systems MS-DOS 4.01 will work fine if you delete all "SHELL" files and modify the AUTOEXEC.BAT file accordingly.

Ordinary etiquette requires: if you write to me and expect an answer include return postage, s'il vous plaît! I am after all an unsanitary retired male citizen (dirty old man to you) on a rather limited income.

This program, written in GW Basica, will print all the printable ASCII characters on a 8½ x 11" sheet, using a Panasonic KX-P1124 printer. The printer is left in "factory new" mode. This program usually will not work correctly with any other printer.

```
10 WIDTH "lpt1:",255'  Set up printer as a
20 OPEN "lpt1:" AS #1'- random access file
30 PRINT #1,CHR$(27)+"t"+"1":PRINT #1,CHR$(27)+"6"
40 PRINT #1,CHR$(27)+"x"+"1"
50 CLS:PRINT:PRINT
60 PRINT TAB(15)"Select font:":PRINT
70 PRINT TAB(20)"Courier- - - - - - - - - 0"'
80 PRINT TAB(20)"Sans Serif - - - - - - - 1"'
90 PRINT TAB(20)"Courier  - - - - - - - - 2"'      Select
100 PRINT TAB(20)"Prestige- - - - - - - - 3"'      typeface
110 PRINT TAB(20)"Script- - - - - - - - - 4"'
120 PRINT TAB(20)"Bold PS - - - - - - - - 6":PRINT'
130 LINE INPUT"Your selection - - - - ";F$
140 IF F$="" OR F$="5" THEN 50
150 FF=VAL(F$)
160 IF FF>6 THEN 50
170 B$=CHR$(27)+CHR$(94)
180 LPRINT:LPRINT:LPRINT:LPRINT
190 LPRINT CHR$(27)+CHR$(107)+F$
200 LPRINT TAB(20)""STRING$(41,"")""
210 LPRINT TAB(20)" *** The PANASONIC  ASCII Characters ***
220 IF FF=0 THEN S$="-- Courier --"
230 IF FF=1 THEN S$="--SansSerif--"
240 IF FF=2 THEN S$="-- Courier --"
250 IF FF=3 THEN S$="-- Prestige--"
260 IF FF=4 THEN S$="-- Script  --"
270 IF FF=6 THEN S$="-- Bold PS --"
280 LPRINT TAB(20)"              "S$"              "
290 LPRINT TAB(20)""STRING$(41,"")"":LPRINT
300 LPRINT:LPRINT
310 FOR X=2 TO 25
320 FOR Y=0 TO 9
330 Z=10*X+Y:R=Y*9
340 IF Z=21 THEN R=8
350 IF Z>21 AND Z<34 THEN R=Y*8
360 IF Z>33 AND Z<40 THEN R=Y*9-3
370 IF Z=128 THEN R=64
380 IF Z=129 THEN R=73
390 IF Z=256 THEN 490
400 PRINT #1,TAB(R):PRINT #1,USING"###";Z;
410 IF Z<21 THEN 470
420 IF Z=21 THEN 460
430 IF Z>21 AND Z<33 THEN 470
440 IF Z=32 THEN 470
450 PRINT #1," "BSCHR$(Z);
460 NEXT Y
470 LPRINT:LPRINT:NEXT X
480 LPRINT CHR$(12):CLS:END
```

It is recommended that the sheet printed out by this program be kept handy near your keyboard.

Listing 1

On the Leading Edge

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

The good news is that March 6 has come and gone. The bad news is that March 6 was Michelangelo's birthday, and some idiot came up with a virus to "celebrate". In general, the best news is that there were very few reported occurrences of the Michelangelo virus, so perhaps many people found it before it caused too many problems. Unfortunately, not all occurrences of viruses are reported, so it's difficult to tell how big the problem really was. But I have observed that several writers are walking around with at least one arm in a sling as a result of patting themselves on the back because of getting the "news" out about Michelangelo. It's probably true that the media contributed significantly to the knowledge that this new virus could cause problems, it's sometimes difficult to separate fact from hype. And of course there's always the Friday the 13th virus, which also happens to occur in March this year and just a week later than Michelangelo. Since those time bomb dates have come and gone, it's much easier to take a look at them without getting into a panic. So, let's start by taking a look at what a virus really is.

What a Virus Is

There seems to be a lot of confusion about what a virus really is, and that confusion has even caused a lot of problems in reporting. The problem is that many reports talk about every destructive program as a virus, and that's not true. There are other types of destructive programs, such as a Trojan Horse which looks like one thing, usually a helpful or interesting program, but it usually destroys data one way or another.

There are two key factors that specifically identify what kind of generally destructive program must be dealt with. First, you must understand how the program is transmitted. And second, you need to know what the trigger is.

The first factor, the transmission technique, is what distinguishes a virus from the other kinds of destructive programs like a Trojan Horse. As I mentioned earlier, a TROJAN HORSE program looks like it does one thing, but it usually does something else like destroy a disk directory or scramble the disk's FATs. For example, a few years ago there was one program that displayed a Christmas tree, and then it scrambled the FATs with a message like: Arf, Arf, I got you!" This program was simply transmitted because people copied the file that looked like it just displayed a Christmas tree, and after doing so, it started destroying things.

On the other hand, a true virus has the capability of reproducing itself, usually by "attaching" the destructive program code to a file or files. In most cases, the target files are usually COM or EXE files, but there are known viruses that can attached themselves to data files too. For example, some viruses first attach themselves to COMMAND.COM so that other COM or EXE files may be infected whenever they are run.

It is, of course, possible to have a combination of the two. For example, a trojan horse program might be used to initially get into a computer system, and then the virus code could begin spreading and infecting other files in that system. The original program might be benign (i.e., causing no damage) so that it is just a simple "carrier" of the virus code. Because of that, it is sometimes difficult to be precise on exactly what the transmission mechanism is. It may be a combination of techniques. That's part of what makes the problem so difficult to detect, isolate, and eliminate.

For the most part, most of the viruses you will find today infect other systems without using the trojan horse approach because that is so easy to spot. The single most important point about a virus is that it spreads by attaching itself to one or more files.

But a virus can also be benign or malicious, although the malicious viruses, such as Michelangelo, are the ones that you will most likely hear about. And that brings up the second point: the trigger.

The trigger can be just about anything. For Michelangelo, the trigger is the March 6 date (every year by the way). For the Friday the 13th virus, the trigger is a combination that requires both the day and the date. A destructive program could also keep a count of how many times the program is run, and then it could perform its destruction based on that count, say every 17th time. Or, a program could be set up to destroy data every time it is run, such as the Christmas Tree virus I mentioned earlier.

Obviously, if you know what the trigger is, you can take some steps to prevent any problem, especially if you suspect there may be a virus in your computer. For example, you can change the date in your system so that it never hits March 6 or Friday the 13th, but that only prevents the activation; it does not eliminate the virus.

Detecting and eliminating a virus is tricky, and there are a lot of good programs available that can do the job. We'll take a brief look at a few in a minute, but first let's look at one of the characteristics that will help you determine if your system has the Michelangelo virus.

Finding the Michelangelo Virus

All viruses have some kind of "signature" and are usually quite difficult for a user to detect, but Michelangelo is different. You don't need to be a technical wizard and look at hex bytes to find Michelangelo. All you need to do is run CHKDSK and see how much total memory the program reports.

If your system has 640 K of conventional memory, then CHKDSK will report 655,360 total bytes of memory. That's normal. If you remember that each kilobyte is actually 1,024 bytes, you can calculate that number by taking 640 K times 1,024 bytes per kilobyte. And that 1,024 conversion factor is valid, regardless of how much memory your system has. So far, so good.

However, if CHKDSK reports 653,312 total bytes of memory on a 640 K system, then the Michelangelo virus may be in your computer. If you subtract the two numbers, you will find that this virus takes exactly two kilobytes (2,048 bytes) of memory. Unfortunately, there are few viruses that are this easy to detect without a special program. If you find that CHKDSK reports this odd number of bytes, then I suggest you get one of the "virus-killer" programs that can cope with it.

Before we take a quick look at several of the "virus-killer" programs, it is important to take some immediate precautions if you suspect that Michelangelo is present. First, do NOT exchange floppy disks with anyone, because that's how the virus is transmitted in the first place. There are no confirmed reports that Michelangelo is transmitted through file uploads or downloads, so bulletin boards should not be a concern here. It's only transmitted through floppy disks, so far as anyone knows.

Second, take a look at the suggestions in this article for an antivirus program. Talk to your friends to see if anyone else has found this or any other virus, especially if you exchange floppy disks. Michelangelo attaches itself to the boot sector on a floppy, and it apparently occupies four, 512-byte sectors. If you suspect that you have the Michelangelo virus, then you will need to consider buying an antivirus program. The ones mentioned later in this article can detect and "disinfect" any disks, either floppy or hard drives, and they will specifically work for the Michelangelo and many other viruses.

If you find evidence of the Michelangelo or any other virus in your system, then you may want to be more careful about downloading files from bulletin boards and exchanging disks from other systems. But more on that after we take a brief look at some popular antivirus programs.

Killing Viruses

The first step in getting rid of a virus is to find it. There are generally three ways to detect a virus: scanning a disk for virus "signatures", TSR monitoring of memory and disk accesses, and file change detection. Although there are a lot of good antivirus programs available, I think that the best programs use all three methods, and the ones mentioned here use all of them.

Central Point Software (known for PCTOOLS) has developed the Central Point Antivirus (version 1.2 is the latest as I'm writing this) software, which is one good choice. The Norton AntiVirus software (version 2.0 is the latest version as I write this) is another. Both can run under Windows. And Untouchable 1.0 is available from Fifth Generation Systems, best known as the developers of FastBack. Untouchable cannot run under Windows.

There is a lot of other antivirus software available, and I have selected only three of the best known from a rather long list. Nearly everyone should be familiar with at least one of the companies mentioned, and you may just want to pick the one you're most familiar with. Or, you may want to choose a package based on price. You can find that information at the end of this article as usual. But there is another way to find the virus with a program you may already have because I have recommended it for a number of years.

HyperACCESS/5

HyperACCESS/5 is my favorite communications program, and one reason I continue to recommend it is because of its many features. One of those features is HyperGuard which provides a virus detection by scanning files for virus signatures. HyperGuard actually provides three different kinds of virus detection.

The first is perhaps the most important. Many viruses (not Michelangelo) are spread by file transfers from other systems, such as a bulletin board, and you may not know about the virus until it's too late. HyperGuard can detect and will automatically abort the file transfer (unless you override it) if a virus signature is identified in the incoming file. And you can keep HyperGuard up to date by dialing the Hilgraeve Bulletin Board and downloading the latest version of the signature file which is the same one used by the IBM Virus Scanning Program. Or, if you use HyperACCESS/5 at work, you can just call your IBM Account Representative for the latest copy of the signature file.

The second kind of virus protection can be invoked if you use HyperACCESS/5 to copy files from one disk to another, such as copying files from a floppy disk to a hard drive. While this is a little "extra step" because you need to start HyperACCESS/5, it provides virus protection during the copy process to be sure that no identified viruses are copied to your hard drive.

The third kind of virus protection allows you to use HyperACCESS/5 to scan existing files on a floppy or hard disk for viruses. If the scan is "clean", then you don't have any of the virus signatures that are included in the signature file.

I think this is really a dynamite feature to have in a communications program because virus detection is perhaps the most important part of protecting your system, especially when you are downloading files from a bulletin board or copying files from a floppy disk. It is important to note that HyperGuard does NOT provide the capability to eliminate a virus from your system, and you will need to use one of the standard antivirus programs if you do find a virus after it has infected your system.

Like most virus detection programs, HyperGuard can't detect viruses in compressed or packed files (e.g., ZIP, ARC, etc.) because the virus signatures are also compressed. In order for HyperGuard and most other virus detection programs to identify a virus, you will need to unpack these files and then scan them for a virus. I suggest copying compressed files to a floppy, unpacking them, and then using HyperACCESS/5 to copy them to your hard drive, since that will provide good protection, just in case.

HyperACCESS/5 is an amazing program with lots of neat features, and it continues to be highly recommended.

Powering Up for Windows

I have found there are lots of things that one needs to know about Windows, and as a result, I am announcing a new Powering Up series and book: *Powering Up for Windows*. This new series will contain lots of information that you need to know about Windows, especially about configuring WIN.INI and SYSTEM.INI, not to mention all of the usual hints and tips that I have managed to dig out one way or another.

Although I have considered a series on Windows for some time, I had decided to delay it until a better version of Windows was available. If you have been using Windows 3.0, then you probably have run into problems like Unrecoverable Application Errors (UAEs) and system integrity viola-

tions like I have. Except in a few isolated instances, there is not much you can do except reboot the system (and lose data) when those problems occur, but reports are that Windows 3.1 will be much more stable to the extent that those problems will occur rarely, if ever.

So, *Powering Up for Windows* will include information about both Windows versions, 3.0 and 3.1. I think you will find it as helpful as the other Powering Up series, and if you don't have any reference other than the Windows manual, you may find it essential because I have included information that is not in the Microsoft manual.

To give you an idea of the kinds of information you will find in this new Powering Up series, I'll share a tip that will help you speed up the Windows startup. You can suppress the Windows "advertising" screen by using the command: *WIN :*. That's *WIN*, followed by a space, followed by a colon (:). It may reduce your startup time by only a second or so, but any trick to speed up Windows is something I've spent a lot of time looking at. I'm still using my Z-386/16, and it's really too slow to run Windows 3.0, at least for my taste.

Speaking of Windows

As many of you know, I have been quite critical of Windows over the years. One of the major problems I've seen is its speed. DOS programs are much faster, and one needs a really fast system to run Windows. I recommend nothing less than an '386/33 with a fast (20 ms. or less) hard drive and LOTS of memory.

I've been working with Windows 3.0 over the last six months, and I'm still concerned about all of the errors that occur. I have even bought the Word Perfect version 5.1 for Windows, and that is really slow to start up compared to the DOS version. When I make a critical comment about Windows, I inevitably get a letter stating that I'm wrong about something, but that's the way it is. It is a fact that Windows is slow on older systems such as mine. It's not even a rocket on the faster systems, but it is tolerable for the most part. Reports are that Windows 3.1 will be faster, and I hope so.

My major problem with Windows 3.0 is that it is not sufficiently reliable to use as a production environment for my work. UAEs happen with no notice, and even though I have tried to duplicate them, I usually can't. When a UAE occurs, I know that I have lost anything entered since the last save to disk, and that's the reason I don't trust Windows yet. If you have any idea about going to a Windows environment and have not yet done so, I recommend you wait for Windows 3.1, which will hopefully cure at least most of the major problems like the UAEs.

For those of you who went through the "PC revolution" back in the 1982-86 timeframe, the fact that Windows has major hiccups and loses data is no surprise. When I first started using PC-compatible software back then, it seems like every other program I tried managed to freeze at least once during the day. Sometimes it was a "soft freeze" that I could fix with CTRL-ALT-DEL. Other times it was a "hard freeze" that required a power-off reset. And it didn't matter what the problem was, I lost data in both cases.

That's one of the major reasons I used my Z-100 for my production work for so long. Aside from the fact that the Z-100 was generally a great system, I did not have any problems running software under the CP/M operating system because all of that was pretty stable at that point. And when you consider that I used it with WordStar to edit files that were hundreds of kilobytes with only 64 K of memory, it's spectacular. I have a total of 5 MB in my Z-386/16, and that's about the minimum I would want for Windows.

But I think it's important to keep in mind that Windows will eventually become stable enough that it is reliable for production work, and it won't bomb and lose data. It's just a matter of time for all these details to get worked out. But there is one other interesting thing I've noticed in Windows.

Fonts and Typefaces

Microsoft has been promoting a new scalable typeface technology called TrueType that will be built into Windows 3.1. That in itself is an interesting story which has been a battle among some microcomputer companies, but I've found some really confusing things in Windows. Unfortunately, it looks like too many programmers got loose in developing some of the displays. I noticed this originally in Windows Write, so let's take a look at how this works.

A typeface is a character set that has the same design or look. Helvetica, Times Roman, Courier, Script, and LinePrinter are examples of typeface names. But there is more to a typeface than just the name. Italic, bold, and bold italic are different forms of the "normal" typeface, which is called "regular" in Windows. A Times Roman character set may, for example, actually have four different typefaces which includes "normal" (or regular), italic, bold, and bold italic. Although you normally write most of a document in the normal typeface, you may occasionally use another one for emphasis or other reasons.

When a typeface is scaled to a specific size, it is then called a font. For example, "10 point Times Roman italic" is a font and so is 10 point Times Roman bold", so that it is clear that a font includes both a size and a typeface. You will occasionally hear someone refer to just "10 point Times Roman" which generally assumes the normal typeface, but may also include other fonts of the same size, such as "10 point Times Roman italic".

By the way, font sizes seem to be a mystery to many people, but it is really easy to figure out if you know there are 72 points to an inch. A 36 point font is half an inch high, an 18 point font is a quarter of an inch high, and so on. This is just one of those trivial details that is not included in many books I've seen, even in some manuals for word processors.

To confuse the issue, Windows itself seems to consistently and incorrectly call a typeface a font. For example, I looked at the Characters Menu under Fonts..., and there is list labeled Fonts which contains typeface names like Courier, LinePrinter, etc. Again, a typeface doesn't become a font until its size is specified, and the Size is a separate entry in the dialog box.

The difference between a typeface and a font is important because you will find that at least some Windows applications use the terms correctly. Word Perfect 5.1 for Windows and Ventura Publisher are a couple of applications which use the terms correctly.

WP5.1 Win

I recently bought the Word Perfect 5.1 Upgrade for Windows, and it's considerably different from the DOS version which I also have. It's also a lot slower to start, but I expected that. Word Perfect says that WP5.1 Win has complete file compatibility with the DOS version, and my testing indicates no problems of any kind. Unfortunately, WP5.1 Win is not quite able to exterminate all the Windows 3.0 bugs, and I have managed to get UAEs and system integrity violations at odd times. I have also talked to one of my friends who uses Word for Windows (WinWord), and he reports the same general problems with UAEs and system integrity violations. I've also managed to blow up Ventura Publisher, so I'm satisfied that the problems are part of Windows, not with the applications. Hopefully, Windows 3.1 will reduce this kind of thing to a tolerable level.

Perhaps you know about the Word Perfect contest about how many WP5.1 Win copies would be shipped the first month. Would you believe they shipped 369,693 copies the first month? That's spectacular! There seems to be little doubt that the Windows interface is the "wave of the future", and now that REAL applications are becoming available, I'm sure that Windows will be even more popular.

DOS Clusters

I occasionally find something during my testing that is interesting and can help make computer usage more efficient. I was recently checking out cluster factors on my

hard drive and came up with some interesting information, which happens to apply to MS-DOS versions 4 and 5. But before I get into the details, let's briefly review what a cluster is and why it's important.

Disk media is basically divided into sectors, which are 512-byte, pie-shaped units. For example, a 360 K floppy disk has 9 sectors, and an MFM hard drive has 17 sectors. In nearly all computer systems, data are stored in blocks. For a microcomputer, a block is known as a cluster, which is the minimum allocation unit for a given media. That is, the least amount of space that DOS can use to store a file is called a cluster, regardless of what the size of a file actually is. A cluster factor is defined as the number of sectors on a specific media type that is the minimum allocation unit. Some disk media have 1 sector per cluster, 2 sectors per cluster, and more, such as a hard drive that can have up to 16 sectors per cluster in DOS 4 and 5. To help understand this concept, let's look at an example.

Let's say you have an AUTOEXEC.BAT file that is 96 bytes long as shown by the DIR command. And let's say the floppy disk has a cluster factor of 1, which means that each cluster consists of one sector, which is 512 bytes. Since we want to store this 96-byte file, DOS actually stores it in a cluster so that there is really 461 bytes of "wasted" space.

But if you have a hard drive with a cluster factor of 16, that's 16 times 512 or 8,192 bytes. Since DOS cannot allocate disk space to anything smaller than a cluster, there is a LOT of wasted space which is the cluster size (8,192) less the file size (96 in this example) or 8,096 bytes. As you might guess, there can even be more wasted space if the file size is only one byte larger than a multiple of the cluster size. If, for example, a file is 8,193 bytes, DOS will have to allocate two clusters for disk storage, which will result in 8,191 bytes of wasted space in the second cluster.

Wouldn't it be neat if you could adjust the cluster size to "optimize" the storage capacity and minimize the wasted space? You can, at least for a hard drive. Unfortunately, floppy disks have fixed cluster sizes, but you can adjust the partition size of a hard drive to get better storage utilization, because it turns out that the cluster factor is directly related to partition size. Figure 1 shows the cluster factors for partition sizes in MS-DOS versions 4 and 5.

A quick look at Figure 1 makes it easy to see that you can "optimize" the storage space by defining a partition size in the 16-127 MB range which minimizes the wasted space by using a smaller cluster size. The amount of space you can "recover" by simply changing the partition size will depend on how large your files are. But if you have a lot of small files, you can recover a considerable amount of space, especially if you have a large hard drive (e.g., 330 MB) that is currently defined as a single partition.

If you decide to change your partition size, the first thing you must do is back up everything on the hard drive, because changing the partition size will quite effectively destroy all data. This will take some planning because the backups should be made based on your new partition sizes. I generally recommend that all applications remain on drive C because you will probably have to reinstall them if you move them to a different drive. The remainder of the drive(s) can be used for data, which also makes it easer and faster to back up your system on a regular schedule. If, for example, drive C only contains applications and programs that obviously don't change much (unless you install something new), you will rarely need to back it up after the first full backup. In my system, I only take a full backup of drive C once a year, and I make differential backups (with PCTOOLS) when I install something new. Drives that contain data must be backed up much more frequently, perhaps once a week or at least once a month.

Next, you will need to change the partition size by using PART in ZDS MS-DOS version 4 and FDISK in MS-DOS version 5. To change a partition size, remember that you must first delete the existing partition, then add partitions as required. Don't forget to make the first partition bootable using PART or FDISK after the partitions are set up.

The next step is to FORMAT each of the new partitions. Use the `FORMAT C:/S` command to transfer the appropriate DOS files to make drive C bootable. Unless you have some special requirement to have the other partitions bootable, I recommend using FORMAT without the /S switch, since there is no real reason to waste space with the system files. And finally, you will have to restore all of the files to the new drives.

Aside from the fact that you will get better storage utilization by considering the cluster factor, you can also make backups a lot faster, especially if you have a large hard drive. For example, you may want to create a partition where data changes frequently, perhaps daily or weekly. Another partition may contain data that changes less frequently, say on the order of once a month or quarter. And another partition may contain data that is updated only once a year, such as tax information. If you consider how you might improve storage capacity, as well as the advantages of making backups faster, the idea of changing the partition size can be a real timesaver.

Partition Size	Cluster Factor	Cluster Size
Under 16 MB	8	4,096 (4 K)
16-127 MB	4	2,048 (2 K)
128-255 MB	8	4,096 (4 K)
256-512 MB	16	8,192 (8 K)

Figure 1
Cluster Sizes in MS-DOS 4 and 5

Powering Down

Although most of the Windows information will be covered in the new Powering Up series, you will occasionally find some tricks and hints about using Windows in this column. And if you've found something, be sure to let me know so we can share it with other users.

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 49023-0217
(616) 982-3463 (ZUG Software only)

Central Point Anti-Virus 1.2 $129.00
Central Point Software

Continued on Page 18

Getting Started With ...

G. Anderson
3808 Windway Court
Orlando, FL 32817

PCTools

Central Point Software did leave out the kitchen sink in version 7.1 of PCTools, but that's about the only thing not in here. You get seven books for a suggested retail price around $179:

1. Getting Started
2. Desktop Manager
3. Data Recovery & System Utilities
4. Hard Disk Backup for DOS
5. Commute
6. Windows Utilities
7. DOS Shell/File Manager

I totaled the above capability using the other brand of tools software, and would have to spend about $450. So, as you can see PCTools may not have the best of everything, but it certainly is the best value. Although version 7.0 had some bugs in the Windows applications, version 7.1 has corrected those bugs and was sent out free to owners of the buggy version.

System Requirements. First, let's see what you need to run PCTools in a DOS environment. You'll need a PC compatible, any CPU, with at least 512K RAM (640K recommended). PCTools will run with DOS 3.0 or higher, but Central Point recommends 3.2 or higher. Your system will need one floppy drive and one hard drive with 4.6 MB free.

Second, these are some optional items. You'll need a Hayes compatible modem to use Commute, E-Mail, or telecom features. Commute has to have 640K RAM and DOS 3.3 or higher. PCTools works very well with or without a mouse. It supports Microsoft ver 6.14 or higher and Logitech/Dexxa ver 3.4 or higher devices. Fax boards supported include Connection CoProcessor (from Intel Corp), SatisFAXtion Board (Intel Corp), and SpectraFax (from SpectraFax Corp). PCTools can be used on the following Networks: Novell, IBM PC LAN, and PowerLan.

Finally, if you use Windows 3.x there are several applications within PCTools which run in this environment. You must have a high capacity floppy drive.

Installation. There are certain things you can do with PCTools right out of the box. You can repair a disk; unformat a hard disk; recover a deleted file; and recover damaged dBase, Lotus 1-2-3, or Symphony files. However, you really want to install these tools on your hard disk.

Be sure to have a blank formatted system floppy before you start INSTALL. If you have more than one floppy drive, use the one that your computer tries to boot from before booting off the hard disk. This floppy will be your emergency recovery disk which PCTools will make for you during installation. Make sure you test this floppy after you finish the installation.

Insert INSTALL disk in floppy drive f: (A or B in most cases). Then type f:INSTALL and press ENTER. You will then be presented with three options:

1. Install on a stand-alone PC

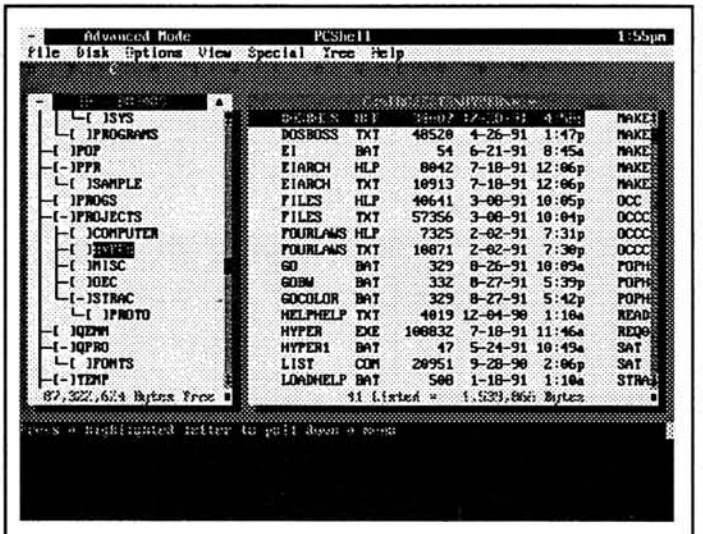


Figure 1

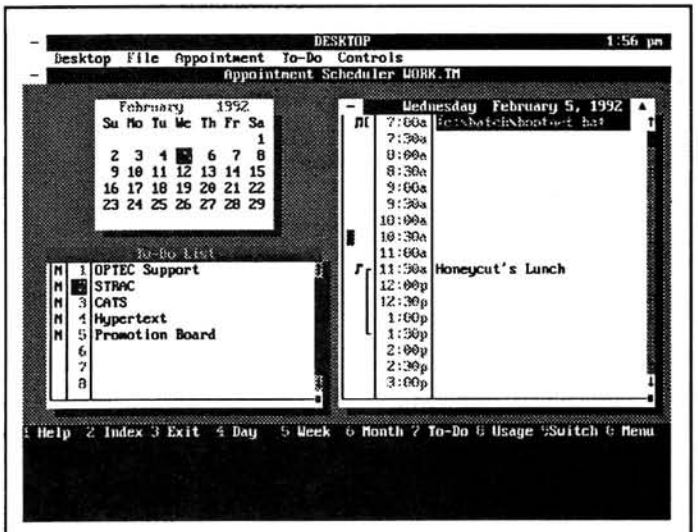


Figure 2

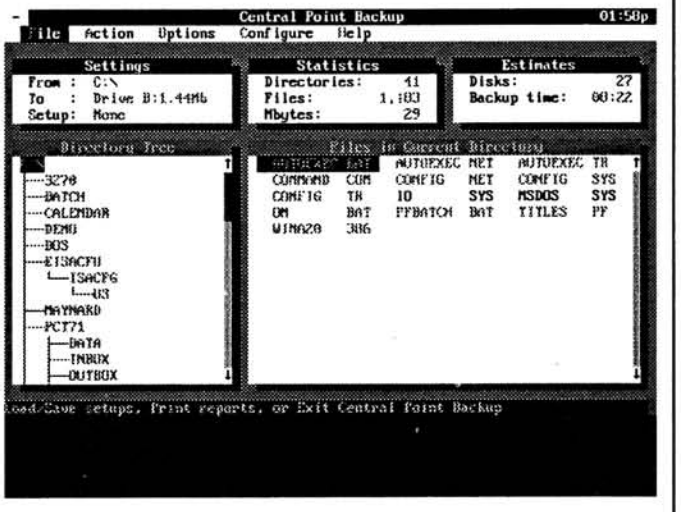


Figure 3

2. Install on a network server
3. Install as network workstation if PCTools has already been installed on server.

Once you make your selection, you start installing the programs on your disk. It is extremely simple. You can install all of PCTools or just the parts you need if you're short on hard disk space.

I would suggest letting PCTools rename DOS's FORMAT.COM file. This will allow automatic use of PCTools format programs which work with other programs such as UNFORMAT.

Also, allow PCTools to modify your AUTOEXEC.BAT and CONFIG.SYS files. It makes copies of your old ones. The advantage to this is you see the order and program parameters to load PCTools programs as TSR's. If you keep this order, you should have no conflicts among the TSR's. If you decide not to use them after all, you can edit these files and REM out the ones you don't want. For example, I don't run PCSHELL as a TSR, only as a stand alone program from the DOS command line. But I know how to load it as a TSR without having to hunt through the documentation.

You will also want to edit AUTOEXEC.BAT to set an environment variable that tells PCTools programs which directory holds your personal information (like appointment data). This is especially important if you run PCTools off of a network. In CONFIG.SYS make sure files is set to 25 or greater, and buffers is set to a small number (3) if you're using PC-CACHE.

Finally, the installation process will search your hard disk for applications which it will add to the menu, a part of PCSHELL. Now that we have PCTools on the computer, let's take a look at what we have.

PCSHELL (Figure 1). This is a utility program that provides easy access to DOS commands, PCTools utilities, and other applications. It works in a DOS-based environment with multiple windows (not to be confused with MS Windows 3.x) displaying different information. It can be used to view just about any format file. This is a real strength; try looking at certain word processing or spreadsheet documents outside of their respective application. File viewers include WordPerfect, WordStar, Lotus, Quattro, dBase, Paradox, graphics, and more. Other functions include file and disk operations, ASCII and HEX editors, transfer utility between two computers, and configurable menu. PCShell can be run TSR (10 k) or as a stand alone program.

DESKTOP (Figure 2). This utility can also be run as a TSR (? k) or a standalone program. Since this is where the appointment calendar is, its much more practical to run it as a TSR. It includes these major applications:

1. Mini word processor for ASCII or

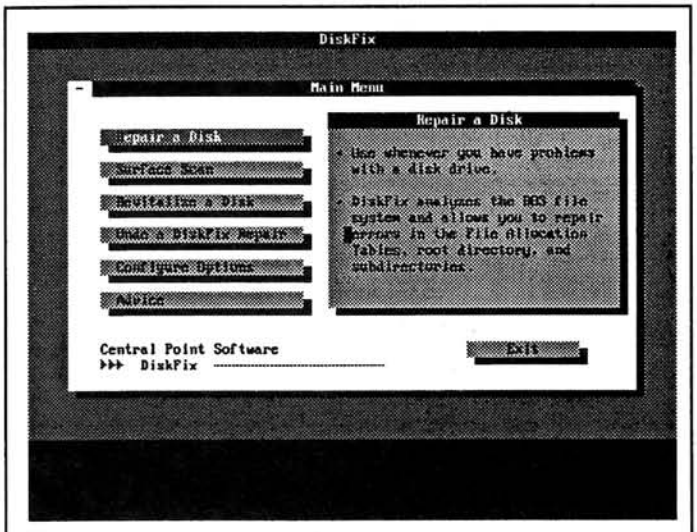


Figure 4

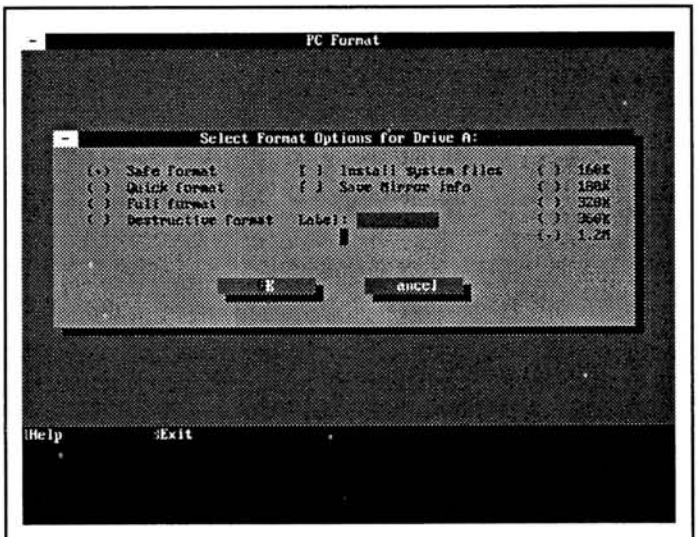


Figure 5

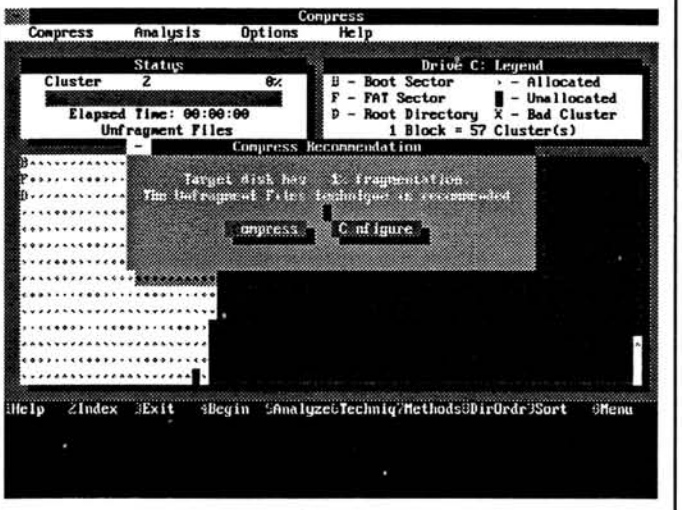


Figure 6

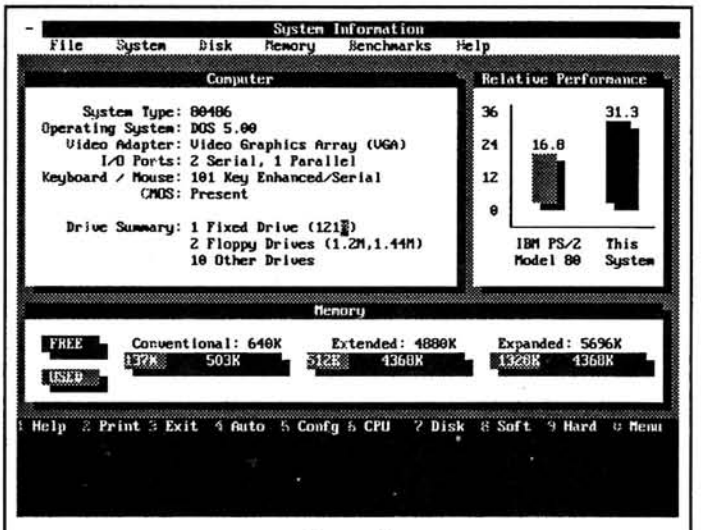


Figure 7

PCTools formats.
2. Outliner
3. Appointment calendar and to-do list
4. Database program which can read dBase
5. Macros
6. E-Mail
7. Modem communications program which can be run in the background
8. FAX board support for a stand-alone or network PC.

COMMUTE. This program lets you operate another computer (i.e., office) from some other location (i.e., home). As a practical example for using this program, think of how many trips to client sites you could save if you could fix their computer via the modem.

CPBACKUP (Figure 3). Allows you to backup and restore valuable data. I guess it would do the same for junk data. It supports backups to floppies, other hard drives, or tapes. Unattended backups can be scheduled as long as the medium has enough capacity. Obviously you could not backup 20 MB to floppies unless you are there to feed the computer multiple floppies.

Data Recovery. There are four major programs in this area:

1. UNDELETE: If you delete the wrong files, you stand a good chance of getting them back depending on how long ago you deleted and whether or not you use delete tracking.
2. DISKFIX (Figure 4) detects and fixes various floppy and hard disk problems related to FAT, directories, clusters, and sectors.
3. UNFORMAT can do just that if the disk was formatted with PCTools.
4. FILEFIX recovers damaged Lotus, Symphony, or dBase files.

Security. PCTools has a wealth of utilities to help you recover lost or damaged data or protect data from prying eyes.

1. MIRROR takes a snapshot of important areas of the hard disks such that you can recover from accidental formats or hard disk failures.
2. DATA MONITOR is a TSR which guards against data loss and protects confidential files. DELETE PROTECTION works with UNDELETE to improve your chances of recovering deleted files. There is a SCREEN BLANKER which includes a password option. DIRECTORY LOCK encrypts a directory and gives access only through use of a password. Important data can be write protected. DISK LIGHT displays an indicator on the screen every time a disk is accessed.
3. WIPE completely clears the data of deleted files from the disk.
4. PCSECURE encrypts, compresses, and password protects files.
5. VDEFEND protects against 500+ viruses. By downloading files from the Central

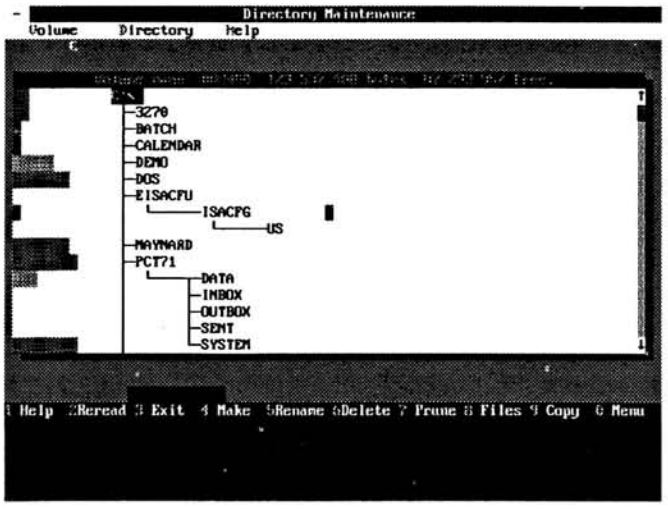


Figure 8

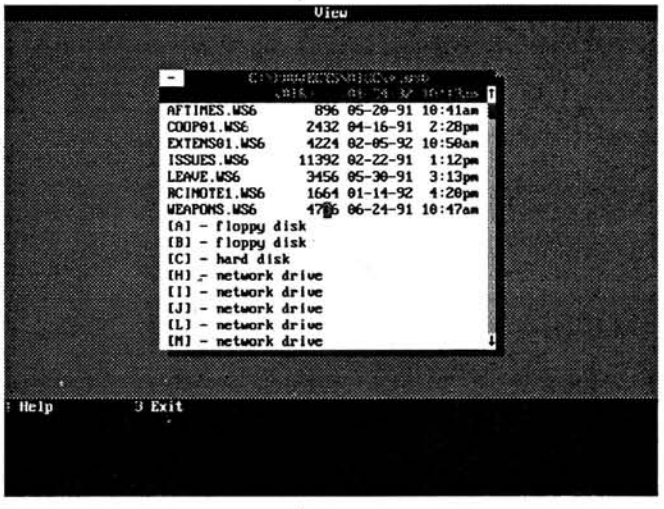


Figure 9

Point bulletin board, you keep your virus protection current.

6. PCFORMAT (Figure 5) formats any kind of disk such that UNFORMAT or UNDELETE can recover data.

Performance. There are several programs which improve the performance of your computer or display information about your system.

1. COMPRESS (Figure 6) unfragments a disk (hard or floppy) with numerous options.
2. PC-CACHE speeds up hard disk access. It is a TSR of course, and can use extended or expanded memory if you have it. If you have a memory manager, it can also be loaded high.
3. SYSTEM INFORMATION (Figure 7) tells you all about your computer's parts.
4. FILEFIND locates files on any drive by name, size, or contents.
5. DIRECTORY MAINTENANCE (Figure 8) allows you to search for, change to, add, rename, delete, and move directories.
6. VIEW (Figure 9) is the stand alone program which PCSHELL uses to look at any format file (see discussion under PCSHELL).

MS Windows 3.x Applications. There are a few of the above utilities which have been written as Windows 3.x applications. These programs use Windows 3.x features and standards. They get installed only if you choose so.

1. CPBACKUP
2. SCHEDULER coordinates appointments, backup schedules, and commute schedules. Tasks are started as scheduled whether you are using Windows or DOS at the appointed time.
3. CPLAUNCHER is a custom menu for starting DOS or Windows 3.x programs.
4. UNDELETE
5. TSR manager configures memory resident PCTools programs for Windows 3.x use.

Using PCTools. There are several ways you can start PCTools programs. First, you can type PCTOOLS and press enter from the DOS prompt. This gives you a main menu from which you can launch PCTools programs as well as other programs. Second, from the DOS prompt, type the name of the PCTools program (e.g., PCSHELL) and press the enter key. With this method, you can run the program with command line parameters as necessary (e.g., COMPRESS C: /SF). Finally, if a PCTools program has been loaded as a TSR (memory resident), then pressing the appropriate hotkey will "popup" that program. When finished with the program, you press the hotkey again to return to your original screen. Some TSR's do not "popup"; they just remain in memory and do their job.

Common Features. There are many common keystrokes and actions no matter what PCTools program you are running. The TAB key activates different windows within a program. The ARROW keys change the position of the cursor or highlight box. If there are menus in the program you are running, there are two ways to pull them down. Pressing the F10 key will activate the menu bar so you can use ARROW keys to highlight a menu choice and press ENTER to select that menu choice. The second way of selecting a menu choice is to press ALT plus the highlighted letter of the menu name. Once a menu is pulled down, use the ARROW keys to move the highlight box and press ENTER to select that menu item. Or, press the highlighted character of the menu item to activate it. All of the normal cursor movement keys will scroll through directory lists, file lists, or files. F3 or ESC will close a window. Closing the main window will also exit the program. PCTools programs work extremely well with a mouse or trackball. In many of the programs, the bottom line is context sensitive help, with more extensive hypertext help available with the F1 or F2 key.

If a PCTools program has a window, there are certain bits of information always in the same places. At the very top is a Title Bar containing the name of the program being used. Right below that is the Menu Bar containing names of pull-down menus. Below the Menu Bar, is the Application Window. In some cases, more than one window can be open, but only one is active (remember the TAB or ESC key to change activation). At the bottom of the screen is a Message Bar which can contain context sensitive help or function key definitions. Three function keys are always the same unless you change them: F1=HELP, F3=EXIT, and F10=MENU. Upper left corner is the Close Box for mouse users. ALT-SPACE will display the system menu.

Some programs will contain a Drive Selection Bar right below the Menu Bar. To select a different drive, press CTRL and desired drive letter. PCTools will let you use the DOS wildcard characters ? and *. Some programs will let you use multiple file specs. For example, *.BAT *.TXT AUTOEXEC.BAT allows action on all files ending with BAT and TXT except AUTOEXEC.BAT.

With any PCTools program, use PCCONFIG to change colors, display characteristics, mouse operations, or keyboard display.

Continued from Page 14

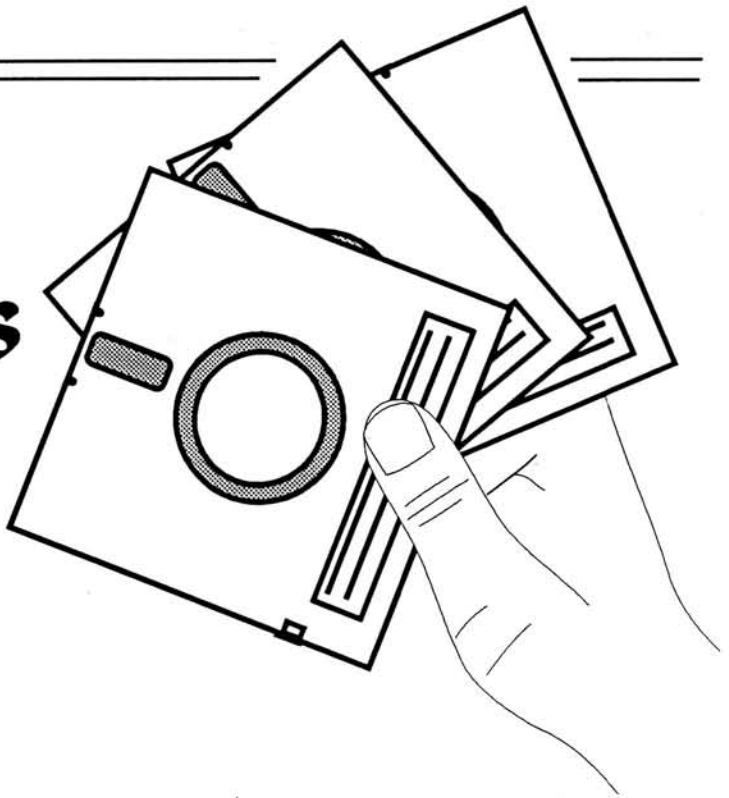
15220 N.W. Greenbriar Parkway,
Suite 200
Beaverton, OR 97006
(503) 690-8088

Norton AntiVirus 2.0 $129.00
Symantec Corporation
10201 Torre Avenue
Cupertino, CA 95014
(408) 253-9600

Untouchable 1.0 $165.00
Fifth Generation Systems
10049 N. Reiger Road
Baton Rouge, LA 70809-4562
(800) 873-4384

HyperACCESS/5 (DOS) $99.00
Hilgraeve, Inc.
Genesis Centre
111 Conant Avenue, Suite A
Monroe, MI 48161
(313) 243-0576

About Floppy Disks and Drives



Rich Hamersley
1989 Laseine
St. Joseph, MI 49085

This is the first article of planned series of articles which attempt explain the some of the inner workings of the PC's floppy disk subsystem. This article is fairly general in scope; subsequent articles will focus on specific aspects of the disk subsystem from the hardware, BIOS, DOS, and user application perspectives.

An Overview of Floppy Disk Basics

Floppy disks have become a standard and necessary part of virtually all personal computers. To some PC users, the floppy disk is the only form of mass permanent data storage the user has. To others, the floppy disk is simply a convenient means to load new programs and data files onto the hard drive and an effective device for backing-up important data files.

Experienced PC users have grown so accustomed to floppy drives and floppy disks that using them requires little conscious thought. But the vast majority of PC users know very little about how the devices work, how data are stored on those thin plastic disks, or how the information is organized so it can be retrieved later when needed.

When the IBM PC was first introduced, it offered only the floppy drive as a high technology solution to the user's data storage needs and even the floppy was an optional accouterment, since those first PCs allowed for an audio tape cassette interface. Many of those early floppy drives could only store a fraction of what today's smallest capacity drives store. The original Disk Operating System (DOS 1.0) only supported the relatively low capacities. After only a short time, however, capacities started to be increased as the IBM designers gained experience and confidence in the the capabilities of the available technology. For example, single-sided disks and drives were quite common then and double-sided disks a rarity; now, single-sided are for all practical purposes extinct.

For several years, floppy drives were only available in the 5¼-inch-wide and 2½- to 3-inch-high high form factor. This is also referred to as a full-height drive. The early improvements in drive manufacturing led to "half-height" (about 1¼-inch-high) form factor which is almost universally used today for 5¼-inch drives. Recent advances have led to even more dramatic size reductions so that now it is possible to find 3½-inch drives in portable PCs that are barely one-half inch in height.

Until the introduction of the IBM PC/AT (the AT stands for Advanced Technology), 360-kilobytes of data were the most a standard PC floppy drive could store on a single disk. With the PC/AT in 1984 came the 1.2MB, 5¼-inch floppy drive and to support this higher capacity drive DOS version 3.0 was required. This drive also marked the first time a drive could read disks that were written by a smaller-capacity drive type. More on that later.

The 3½-inch drive became a standard with the announcement of the PS/2 series of PCs and soon became popular on AT machines in both the 720K and 1.44M capacities. A big advantage the high-capacity 3½-inch drives had over their 5¼-inch cousins was that the 720K disk could be formatted and written in a 1.44MB drive and still be reliably used in 720KB drives. The 1.2MB drives, though generally quite reliable at reading 360KB diskettes, are poorly-suited for writing to the 360KB diskettes. 360K drives more often than not cannot read a 360K diskette that has been written or formatted by a 1.2MB drive.

Table 1 lists some of the disk capacities used on IBM and compatible PCs. The 2.88MB drive is currently quite rare, but should become commonplace in the next year as demand for higher transportable storage grows.

Disk Geometry

Physically, a diskette is a thin, circular sheet of a flexible plastic material that has been coated with a ferromagnetic substance much like that of the magnetic tape in an audio cassette. The disk is protected by surrounding it with a plastic jacket that has openings to allow the drive mechanism the access it needs.

In the drive, the disk is rotated at a constant speed. A "read/write" head, which can magnetize a tiny area of the disk and detect the magnetized areas as the disk spins, creates invisible magnetic "rings" called tracks on the disks.

Tracks are numbered from 0 starting from the outermost track and counting inward. The precise positioning of the head is accomplished almost universally using a stepper motor. A stepper motor, as the name implies, is a motor which can be rotated in small angular increments on the order of a few degrees. The motor can also be held in a position, which is quite necessary when accessing data on a particular track. The stepper motor on a floppy drive can be stepped both in toward and out from the innermost track. To assist the drive controller, a signal is provided that indicates when the head is positioned at track 0. Other tracks are reached by issuing a

Table 1
Typical floppy disk capacities

Disk Capacity	Drive Capacity	Form Factor	Sides	Sectors/ Track	No. of Cylinders	Rotation Speed (RPM)
160KB	360KB	5¼	1	8	40	300
180KB	360KB	5¼	1	9	40	300
320KB	360KB	5¼	2	8	40	300
360KB	360KB	5¼	2	9	40	300
360KB	1.2MB	5¼	2	9	40	360
720KB	720KB	3½	2	9	80	300
1.2MB	1.2MB	5¼	2	15	80	360
720KB	1.44MB	3½	2	9	80	300
1.44MB	1.44MB	3½	2	18	80	300
2.88MB	2.88MB	3½	2	36	80	300

specific number of step pulses to the step motor as necessary to move the head from its current position to the desired track.

The precise location and spacing of the tracks are standardized allowing disks written on one drive to be readable by another. In the case of 3½-inch disks, Track 0 forms a circle 3.00" in diameter. Also standard is the track spacing, 135 tracks per inch (TPI), though there are only 80 tracks on a side of the disk. This means that to step from track to track the head only has to be moved by .0074" and that the total distance from Track 0 to Track 79 is only .585" or just a little over half an inch.

The length (that is the circumference of the circle formed by the track) of Track 0 is approximately 9.42" and, on a 1.44M diskette, the data is stored at a linear density of about 10610 bits per inch (BPI). For track 79, the length is only 5.74" and the linear bit density is 17434 BPI, 64% higher than Track 0. For this reason, Track 0 should provide the most reliable storage and in fact, that is where DOS stores the most critical data when the disk is formatted.

The magnetic properties of the disk coating also play an important role and as such must be carefully matched to the intended disk capacity. The permeability or magnetic "hardness" of the coating must be higher as bit and track densities increase. Higher capacity disks require a magnetically "harder" media than the lower capacity disk. This is necessary primarily to minimize the effect of the read/write head's magnetic field on adjacent tracks and bits.

As the permeability of the media increases, so must the intensity of the magnetic field while the size of the head must be reduced to keep the magnetic field localized. This is illustrated in Figure 1 which shows that when the track density is doubled from 40 to 80 tracks (as in the case of going from a 360K drive to a 1.2M drive), the track width on the higher capacity disk is less than half that on the 40 track disk.

During a write, as the disk spins under the head, the bits must be as equally spaced as possible. Since the disk is spinning at a constant rate, distance between bits is equivalent to a fixed interval of time. The inverse of this bit time interval is known as the bit data rate (DTR) and ranges from 250 kilobits-per-second (KBS) for 360K and 720K disks to 500KBS for 1.2MB & 1.44MB disks. The new 2.88MB drives record data at 1000KBS (1 million bits-per-second) and yet remain compatible with the 720K & 1.44M disks. Besides being able to store more data, the higher capacity drives offer another advantage in that they read and write data faster than 360K and 720K drives due to the higher data rates.

Most drives spin the disk at 300 RPM (5 revolutions per second). The lone exception (as shown in Table 1) is IBM's misfit 1.2MB drive which spins at 360 RPM. Since this drive can be used to read 360K disks this presents somewhat of a problem. The 360K drive spins at 300 RPM and records data at 250KBS. At 360 RPM in the 1.2MB drive, the 360K disk's DTR is no longer 250KBS, but instead is an oddball 300KBS (250 * 360/300). Some (mostly older) 1.2MB drives approached the 360K disk compatibility issue by actually offering both 300 and 360 RPM spin rates, but these drives are rare since it is much simpler and cost-effective to support a 300KBS data rate than it is to support a multi-speed drive motor. Virtually all Floppy Disk Controllers support the 300KBS data rate so there's no need for dual-speed 5¼ drives. The 1.44MB drives can also read and write 720K disks, but the data rate when doing this is the same 250KBS rate as used on 720K drives. Drives such as the 1.2MB and 1.44MB are referred to as dual-rate or multi-rate drives.

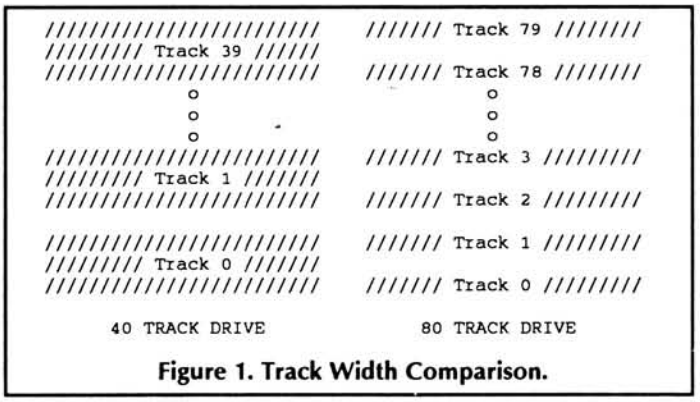


Figure 1. Track Width Comparison.

Continued on Page 28

The FDC/FDD Interface

One would think that with IBM's experience and leadership in the personal computer industry that something so simple as the interface with a floppy drive would be standardized. But such is not the case. It seems that the designers could not see past the current design's intended capabilities and such new developments leading to higher and higher capacity disks and drives had to be kludged to work with existing machines. No forethought was availed to provide for drives capable of supporting multiple disk capacities.

Figure 2 shows floppy drive connector pin assignments typically found in a PC. The interface is actually pretty simple. There are serial data in and out signals for disk reads and writes and a Write Gate to enable the head for writing to the disk. Track 0 and Index outputs give the controller information about the disk and head positions and the Step and Direction inputs tell the drive when and how to move the head to a different position. The drive and motor select inputs determine whether or not the drive pays an attention to the other control signals.

Media Sensing

ZDS has traditionally used media sensing drives in its computers. What this means is that the drive has circuitry that can detect the capacity of the media currently in the drive. Media sensing drives have a pin on the interface with the FDC that can be used to pass on the media size information to the firmware if the hardware support for this has been provided. ZDS, like IBM, chose not to include this extra interface hardware. Instead, another method is used by the firmware to determine the media size. More on that in a subsequent article.

The use of media sensing drives has sometimes been cause of frustration for some customers who, in a false sense of frugality, wish to use 720K disks as if they were 1.44M disks but find that they cannot because the drives themselves won't allow it. While this practice may seem to work OK for some vendors' machines and affords the user the perception of saving

Lotus 1-2-3

Release 3.1 Part 2

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

Unfinished Business

I neglected to mention a couple of important items in Part 1 of this article. First, 1-2-3 is a registered trademark of Lotus Development Corporation, 55 Cambridge Parkway, Cambridge, MA 02142. Secondly, my sincere thanks to Pat Swayne for his SCreen IMage capture utility (SCIM). SCIM was used to capture all the screens contained in this article.

Quick Review

Last time we discussed the basics of moving around the spreadsheet, entering labels, using shortcut keys, and taking advantage of the WYSIWYG interface to polish up our sample spreadsheet. By now you should have a basic understanding of the way 1-2-3 operates. If you don't feel comfortable with 1-2-3 yet, go back and review Part 1. Due to the amount of material left to cover, I will be stepping up the pace somewhat. You will need to use the things you learned in Part 1 of this series. In Part 2, we will cover formulas, functions, and three-dimensional worksheets. After completing Part 2, you should be getting more comfortable with the way 1-2-3 operates, and should be getting some ideas for your own worksheets.

Getting Started

Since you saved your worksheet last time, there is no need to retype all the information from Part 1. Enter /File Retrieve. When 1-2-3 prompts you, "Enter name of file to retrieve:", type "XYZ91SUM" and press [ENTER]. Instead of typing the file name, you also may use the arrow keys to highlight the file you wish to retrieve in the third line of the control panel, and then press [ENTER]. If you have many files on your hard drive, you can press [F3] at the "Enter name of file to retrieve:" prompt to list the files, use the arrow keys to highlight the file you want to retrieve, and press [ENTER]. You should now see your worksheet from last time, complete with all the WYSIWYG enhancements.

Three Dimensions

If you have used a spreadsheet for any length of time, you have probably found it frustrating to try to fit all your information on one page. Spreadsheets lend themselves very well to keeping track of large amounts of information, but it can be difficult to organize into manageable sections, since only a portion is visible on the screen. Attempting to print the worksheet presents additional problems. I have seen users who have worksheets that are so large that they must be printed out in sections and taped together for a presentation. These problems can be greatly reduced with 1- 2-3 by using its 3-dimensional capability.

In the sample worksheet, you set up a summary report for XYZ Company. Normally, this worksheet would contain all the information for the individual branches of the company, in addition to the summary information for the entire company. You would have to maintain figures for Gross Sales, Expenses, Profit, and Margin Percentage for each individual branch. If this were a real company, you would undoubtedly have much more information to maintain for each branch. With 1-2-3, you can use the top sheet to display only the summary information desired, and place the detail information for individual branches on their own worksheet. This means that the detail worksheets may be as simple or as complex as necessary without cluttering up the summary worksheet. You can even create multiple summary worksheets, presenting a different mix of information, using data from the individual branch worksheets. The ability to divide the information into several individual worksheets helps you generate more concise reports and maintain better organization of your data.

To use the 3-dimensional capability of 1-2-3 on your sample worksheet, enter /Worksheet Insert Sheet. You are now given a choice whether to insert the sheet Before or After your current sheet. Since you want the summary worksheet to remain on top, press [A] for After. 1-2-3 now asks, "Enter number of worksheets to insert:". Since there are three branches of the company and you need one worksheet for each branch, press [3] and then [ENTER]. 1-2-3 will now insert three blank worksheets after your sample worksheet. To move between worksheets, use the [CTRL-PGUP] and [CTRL-PGDN] keys. The [CTRL-PGUP] key moves you to the next worksheet letter, and the [CTRL-PGDN] key moves you to the previous worksheet letter. You may also press [CTRL-HOME] to move directly to sheet A.

Move the cell pointer to sheet B and position the cell pointer in location B:A1. Type "XYZ Company (Chicago Branch)", and press [ENTER]. Move the cell pointer to location B:A2. Type "'1991 Monthly Sales

Report", and press [ENTER]. If you forget to enter the single quote (') before "1991", 1-2-3 will beep and place you in EDIT mode. Edit the entry in the second line of the control panel by moving the cursor to the beginning of the text using the arrow keys. Enter a single quote ('), and press [ENTER]. Remember that you also may press the [HOME] key to move quickly to the beginning of the text when editing. To center the text, move the cell pointer to location B:A1. Enter :Text Align Center, highlight the range B:A1 through B:E2, and press [ENTER]. To draw an outline box, enter :Format Lines Outline, highlight the range B:A1 through B:E2, and press [ENTER]. To add shading to the outline box, enter :Format Shade Light, highlight the range B:A1 through B:E2, and press [ENTER]. To add a drop shadow, enter :Format Lines Shadow Set, highlight the range B:A1 through B:E2, and press [ENTER]. You should now have a shaded, outlined, shadow box containing centered text. Worksheet B should now look like Figure 1.

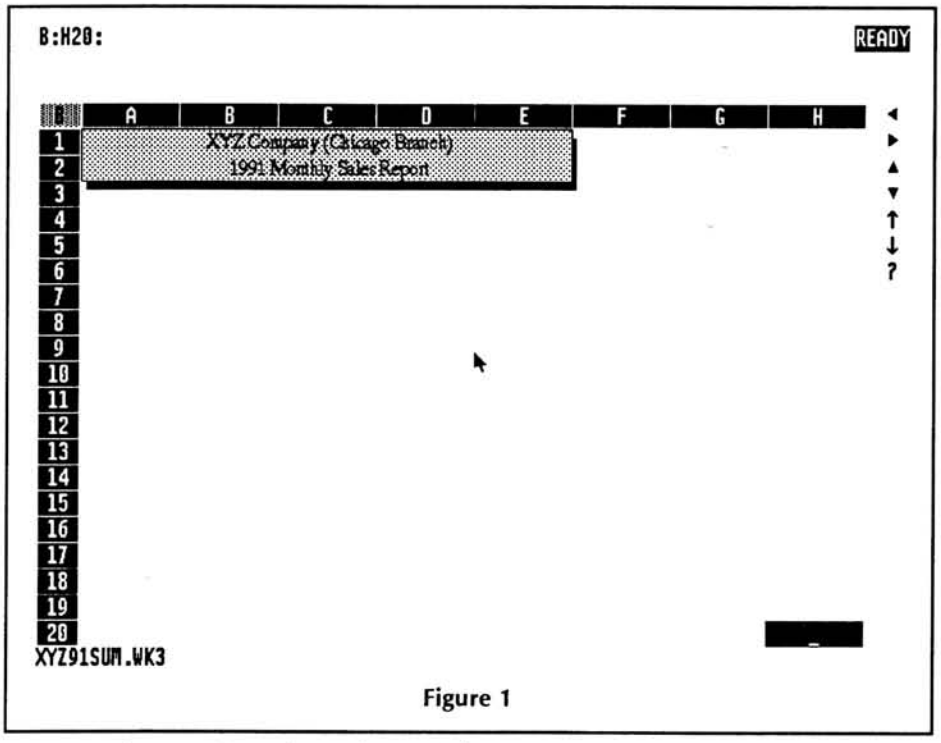


Figure 1

You can copy information from one worksheet to another by using the /Copy command just as you did in Part 1. To copy the two title rows, move the cell pointer to cell B:A1. Remember that this means sheet B (B:) and cell A1 (A1). Enter /Copy, highlight the range B:A1 to B:A2 using the arrow keys, and press [ENTER]. Since you originally entered the text in cells B:A1 and B:A2, you do not need to highlight the entire range from B:A1 to B:E2. When 1-2-3 asks, "Enter range to copy TO:", press [CTRL-PGUP] to move to sheet C. Press the period (.) key to anchor the beginning of the target range and press [CTRL-PGUP]. Your cell pointer should now be on sheet D. Press [ENTER] and 1-2-3 will copy the text from sheet B to sheets C and D. Move the cell pointer to sheet C. Notice that only a portion of your WYSIWYG enhancements were copied to the other worksheets. WYSIWYG has a copy command of its own to copy the special formatting. Move the cell pointer back to sheet B and position it at location B:A1. Enter :Special Copy, highlight the range B:A1 to B:E2 and press [ENTER]. Move the cell pointer to C:A1, press the period (.) key to anchor the beginning of the target range, move the cell pointer to D:A1, and press [ENTER]. If you completed these steps correctly, worksheets B, C, and D should look identical.

Move the cell pointer to cell C:A1. Press the [F2] key, which is the edit key for 1-2-3. In the second line of the control panel 1-2-3 should display "^XYZ Company (Chicago Branch)", and the mode indicator should read "EDIT". Use the [BACKSPACE] key to delete "(Chicago Branch)". Now type "(Los Angeles Branch)", and press [ENTER]. If you entered everything correctly, the text in this cell should read "XYZ Company (Los Angeles Branch)". Depending on how good your typing skills are, it is often faster to copy and edit a cell rather than retyping the entire cell. Move to cell D:A1 and edit this cell to read "XYZ Company (New York Branch)".

To complete the shells of the individual branch worksheets, enter the following data into the proper cells. Remember not to type the double quotes (").

B:A5 - "^Month"
B:B4 - "^Gross"
B:B5 - "^Sales"
B:C5 - "^Expenses"
B:D5 - "^Profit"
B:E5 - "^Margin"
B:A6 - "\-"
B:A20 - "^Totals"

Worksheet B should now look like Figure 2.

Use the copying skills you learned in Part 1 to complete the worksheet for the Chicago Branch of XYZ Company. When

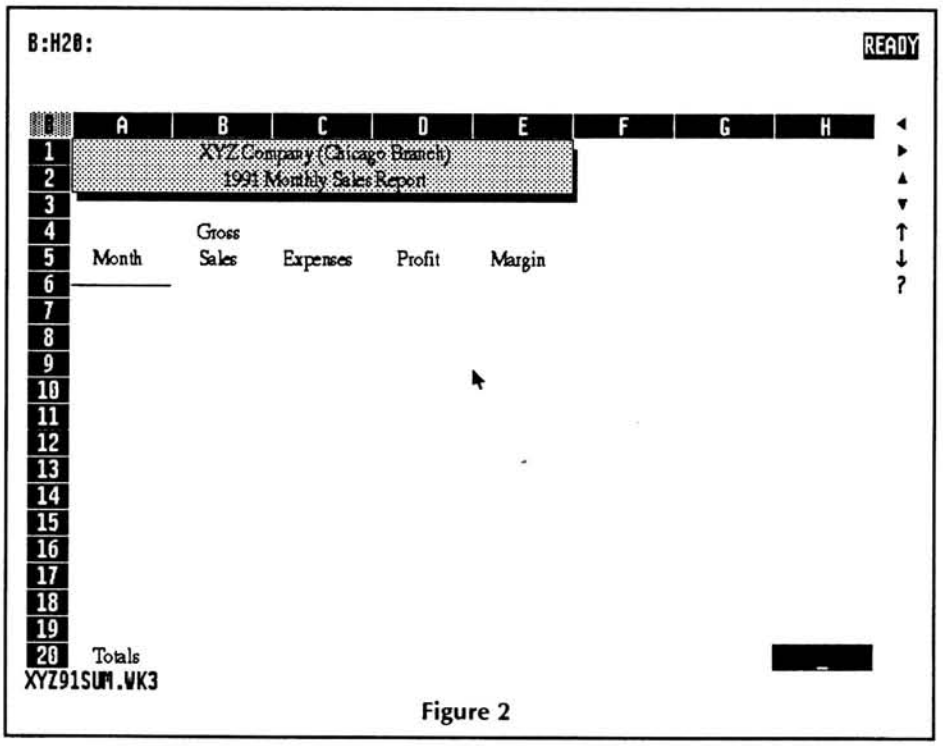


Figure 2

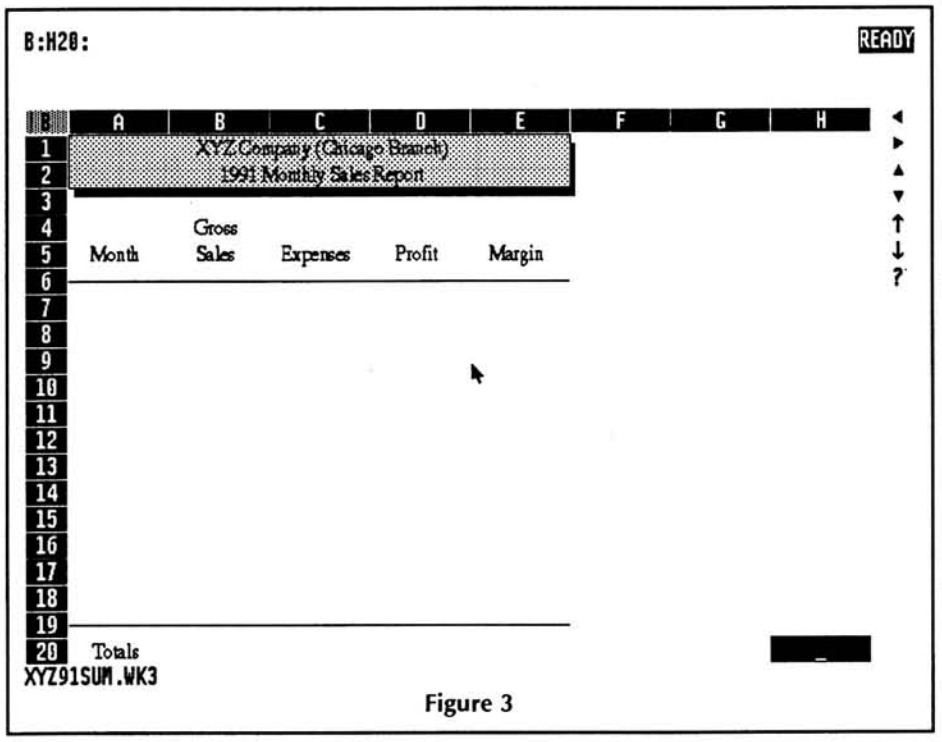


Figure 3

completed, your worksheet should look like Figure 3.

To save time, copy the entire range from B:A4 through B:E20 over to the Los Angeles and New York individual branch worksheets (sheets C and D). To do this, move the cell pointer to B:A4 and enter /Copy. Use the arrow keys to highlight the range B:A4 through B:E20, and press [ENTER]. When 1-2-3 prompts, "Enter range to copy TO:" press the [CTRL-PGUP] key to move to worksheet C. Move the cell pointer to C:A4, which is the start of the destination range. Press the period key (.) to anchor the range here. Press the [CTRL-PGUP] key to move to worksheet D. The cell pointer should now be positioned at location D:A4. Press [ENTER], and 1-2-3 will complete the process, copying your source range (B:A4 through B:E20) to worksheets C and D. When you complete these steps, you should have worksheet shells for the Chicago, Los Angeles, and New York branches of XYZ Company.

Values

1-2-3 treats any cell entry that begins with a number or mathematical symbol as a value. If you are entering numbers into a worksheet, you can simply type them in and press [ENTER]. The period (.) is used to indicate a decimal point. Enter the following data into the proper cells. Do not become discouraged because of the amount of data that must be entered. There is a reason for using these figures that will become clear to you later in this article.

B:B7 - "91218.38"
B:B8 - "112276.48"
B:B9 - "93047.18"
B:B10 - "87840.39"
B:B11 - "92817.49"
B:B12 - "60061.39"
B:B13 - "74773.29"
B:B14 - "79965.58"
B:B15 - "91187.55"
B:B16 - "94899.47"
B:B17 - "104472.11"
B:B18 - "120980.34"

B:C7 - "62838.14"
B:C8 - "87443.20"
B:C9 - "67814.01"
B:C10 - "63038.16"
B:C11 - "67815.35"
B:C12 - "39262.94"
B:C13 - "54890.47"
B:C14 - "81236.74"
B:C15 - "79440.00"
B:C16 - "74299.80"
B:C17 - "87312.08"
B:C18 - "93408.50"

Notice that the figures you entered are difficult to read due to the varying length of each number, as shown in Figure 4.

Numbers with zeroes following the decimal point are truncated after the last significant digit, causing the decimal points not to line up properly. If WYSIWYG is unable to display the entire number, it may also truncate the number and round it. 1-2-3 has many formatting options to eliminate these problems, depending on your specific requirements. 1-2-3 release 3.1 allows you to display up to fifteen decimal places. For this worksheet, two will be sufficient. Appropriate formatting options for this application would be Fixed, Currency, or comma(,). Fixed formatting displays a number with a fixed number of decimal places. Currency format displays a fixed number of decimal places (usually 2), commas in the thousands position, and a dollar sign ($) preceding each number. Comma (,) format is similar to Currency format except that it

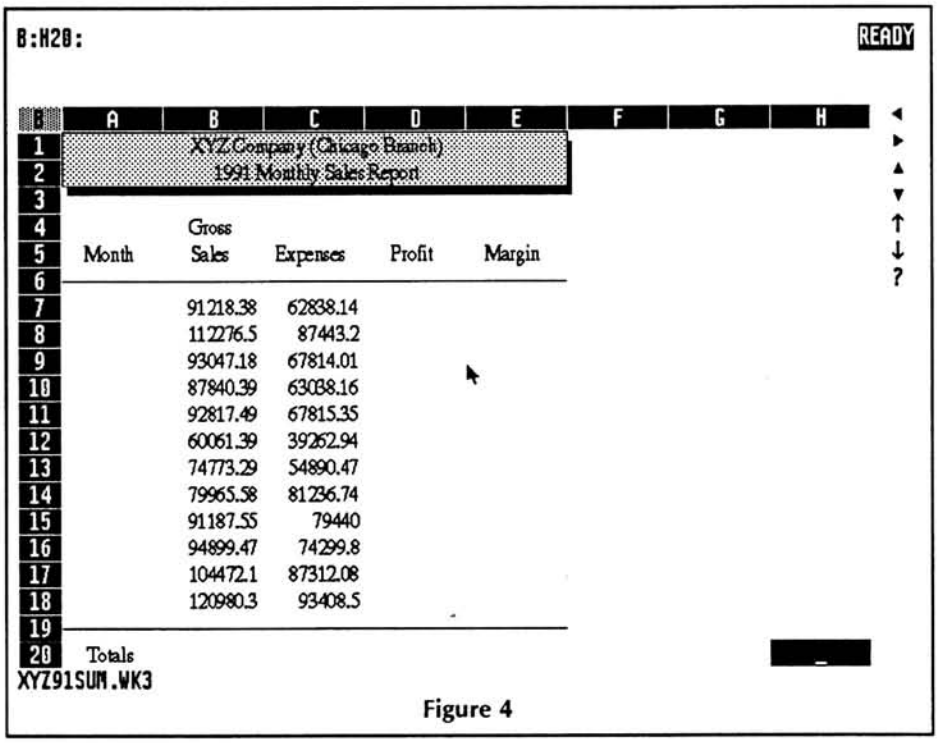


Figure 4

does not display the dollar sign. Examples of each type of formatting option are as follows:

Fixed	10000.00
Currency	$10,000.00

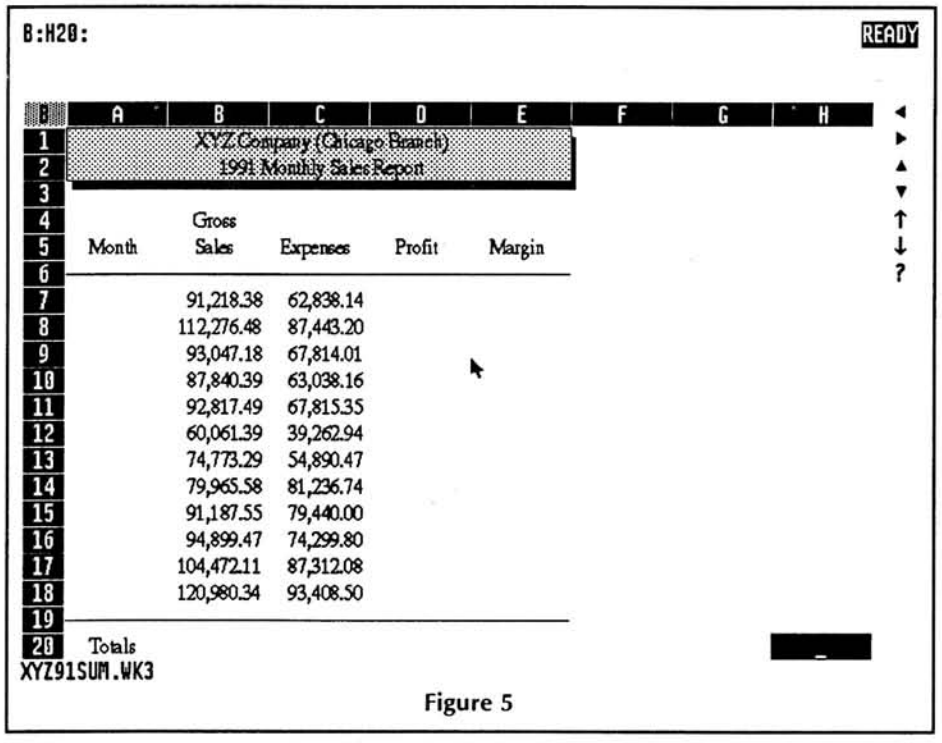


Figure 5

,(comma) 10,000.00

My preference for financial worksheets is to use the ,(comma) format. Move the cell pointer to B:B7. Enter /Range Format ,(comma) [ENTER]. Highlight the range between B:B7 and B:C18 by using the arrow keys and press [ENTER]. Notice that all the numbers are aligned properly, making them more readable. Please refer to Figure 5.

Formulas

Formulas are used whenever calculations are needed in a worksheet. Move the cell pointer to B:D7. To calculate profit, you would subtract Expenses from Gross Sales. Enter "+B7-C7" and press [ENTER]. 1-2-3 interprets this formula by taking the contents of cell B7 and subtracting the contents of cell C7. It is important to remember to place a plus sign (+) before the first cell reference in a formula, otherwise 1-2-3 will interpret your data as a label. Formulas can be copied from one cell to another in the same manner as text, but there is an important difference. When text is copied, it is not altered in any way. When formulas are copied, 1-2-3 will automatically update any cell references contained within the formula when it is copied to a new location. For example if you copy the formula from cell B:D7 to cell B:D8, the formula will become "+B8-C8". Try it and see for yourself. Since the new location of the formula is one row below the old formula location, 1-2-3 increments the row references in the formula by one. This is called a relative cell reference, meaning that the formulas are adjusted depending on their physical relation to the original formula. Copy the formula in cell B:D8 to cells B:D9 through B:D18. Format the range B:D7 through B:D18 with ,(comma) format. Notice that the ,(comma) format shows negative numbers in parentheses as shown in cell B:D14.

Move the cell pointer to cell B:E7. To calculate Margin you would divide Profit by Gross Sales. Enter "+D7/B7" and press [ENTER]. To display the result as a percentage, we could either multiply by 100, or simply format the cell as Percent. The advantage of using the Percent format is that 1-2-3 automatically multiplies the result by 100, and adds the percent symbol (%) within the same cell. Older spreadsheets forced you to insert an extra column to contain the % sign. Enter /Range Format Percent [ENTER] [ENTER]. The cell is now formatted to two decimal places, and the percent symbol is displayed to the right of the number. Copy the formula in cell B:E7 to B:E8 through B:E18. Notice that the percent format was also copied to the target range along with the formula. You can save time by formatting the original cell prior to copying.

Formulas also may be used to transfer the contents of one cell into another, even if the cell contains text. Move the cell pointer to B:A7. Enter "+A:A7" and press [ENTER]. The text you entered in location A:A7 of the summary worksheet (Jan) now appears in the worksheet for the Chicago branch. One limitation of using this method is that only the text itself is transferred. Label prefixes are not copied. An advantage of using formulas to update text entries is that changes made to the original cells are transferred automatically. For example, if you wanted to change your month designation from "Jan" to "January", you would only have to update your summary page. The changes would be reflected throughout the rest of your worksheets. Copy the formula in B:A7 to the range B:A8 through B:A18. Now copy the range B:A7 through B:A18 to the range C:A7 through D:A18. If you completed these steps properly, worksheet B should look like Figure 6.

Enter the following figures into the proper cells. These have been rounded off to make them easier to enter.

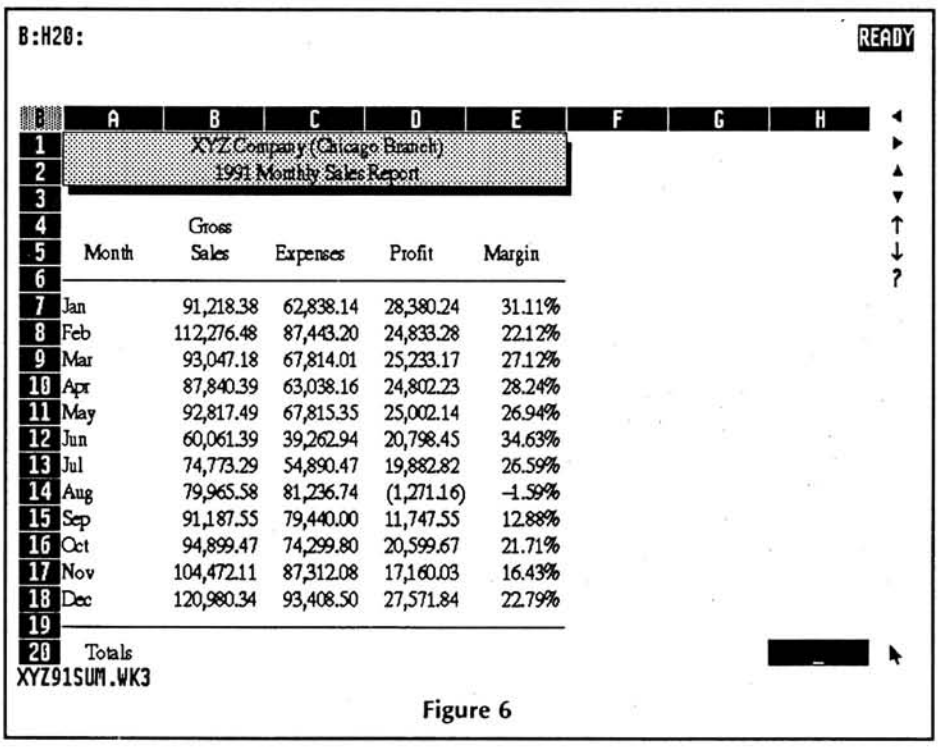


Figure 6

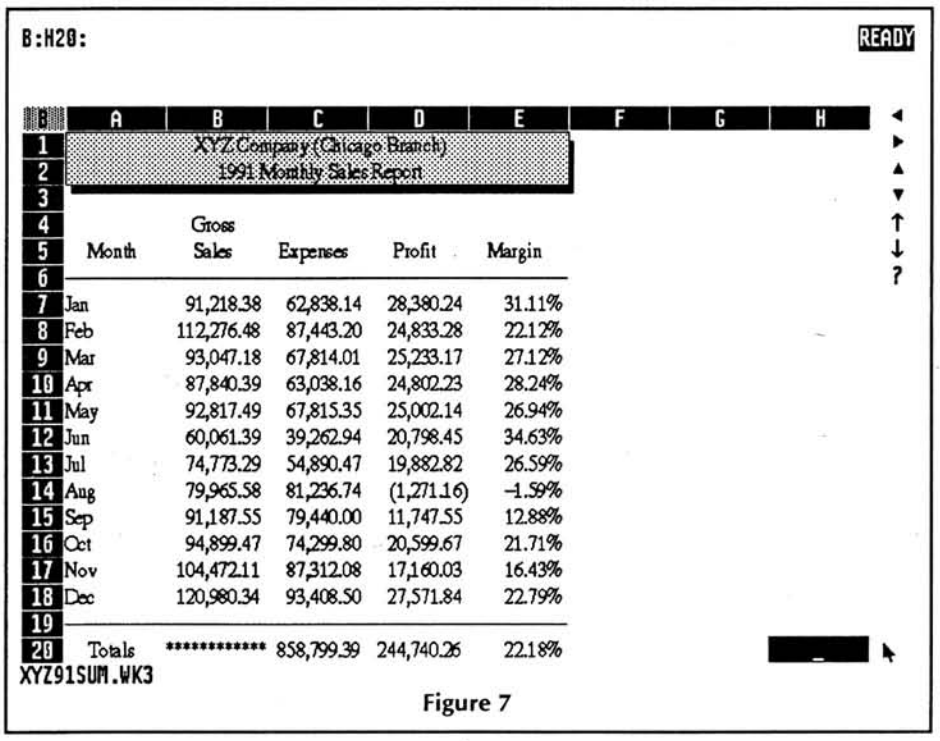


Figure 7

C:B7 - "79000.00"
C:B8 - "93000.00"
C:B9 - "97000.00"
C:B10 - "102000.00"
C:B11 - "86000.00"
C:B12 - "78000.00"
C:B13 - "69000.00"
C:B14 - "83000.00"
C:B15 - "92000.00"
C:B16 - "88000.00"
C:B17 - "94000.00"
C:B18 - "108000.00"

C:C7 - "59000.00"
C:C8 - "72000.00"
C:C9 - "91000.00"
C:C10 - "95000.00"
C:C11 - "69000.00"
C:C12 - "63000.00"
C:C13 - "51000.00"
C:C14 - "70000.00"
C:C15 - "79000.00"
C:C16 - "78000.00"
C:C17 - "103000.00"
C:C18 - "92000.00"

D:B7 - "128000.00"
D:B8 - "119000.00"
D:B9 - "99000.00"
D:B10 - "122000.00"
D:B11 - "103000.00"
D:B12 - "91000.00"
D:B13 - "94000.00"
D:B14 - "102000.00"
D:B15 - "142000.00"
D:B16 - "121000.00"
D:B17 - "133000.00"
D:B18 - "151000.00"

D:C7 - "99000.00"
D:C8 - "88000.00"
D:C9 - "101000.00"
D:C10 - "98000.00"
D:C11 - "71000.00"
D:C12 - "59000.00"
D:C13 - "63000.00"
D:C14 - "76000.00"
D:C15 - "118000.00"
D:C16 - "93000.00"
D:C17 - "109000.00"
D:C18 - "124000.00"

Move the cell pointer to C:B7. Enter /Range Format ,(comma) [ENTER]. Use the arrow keys to move the cell pointer to C:C18. Press [CTRL-PGUP] to move to worksheet D. Notice that the same range you highlighted on sheet C is highlighted here. Press [ENTER] and your entries on both worksheets will be formatted with ,(comma) format.

Use your copying skills again to copy the formulas contained in cells B:D7 through B:E18 to C:D7 through D:E18. If you have trouble, go back and review the procedure you used earlier.

@Functions

Move the cell pointer to B:B20. To total this column you could enter "+B7+B8+B9+B10+B11+B12+B13+B14+B15+B16+B17+B18", but 1-2-3 provides various @Functions that make tasks like this much easier. All @Functions begin with an @ (at sign), followed by the name of the function, followed by one or more arguments enclosed in parentheses. The @SUM function is used to add a range of cells together. Enter "@SUM(B7.B18)" and press [ENTER]. In this case, @SUM is the name of the function, and (B7.B18) is the argument. The argument that you entered specifies the range of cells to be added. 1-2-3 also allows you to point to the range you wish to add. Move the cell pointer to B:C20. Enter "@SUM(", but do not press [ENTER] yet. Move the cell pointer to B:C7. Notice that 1-2-3 has updated the formula in the second line of the control panel by adding the current cell pointer location. Press the period key (.) to anchor the beginning of the range here. Using the arrow keys, move the cell pointer to cell B:C18. The entire range

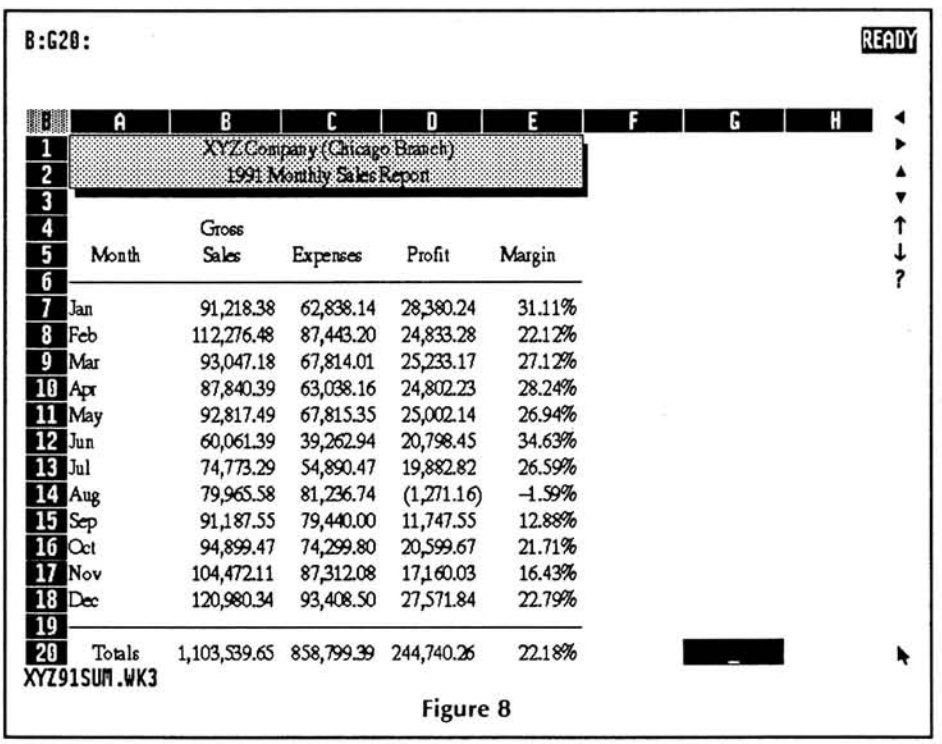


Figure 8

is now highlighted, and the formula has been updated in the control panel to reflect the entire range. Enter ")" to complete the expression, and press [ENTER]. If you are working with formulas that require specific cells and you are not sure of their location, you can use this technique to point at them. If you do not enter the closing parenthesis, 1-2-3 will beep and switch you into EDIT mode. You can now enter the closing parenthesis and press [ENTER].

Copy the formulas in cells B:D18 and B:E18 to cells B:D20 and B:E20. Format cells B:B20 and B:C20 with ,(comma) format. It is not necessary to format cells B:D20 and B:E20, since their format was copied with their formulas. Now copy the formulas in the range B:B20 through B:E20 to the range C:B20 through D:E20. If you completed these steps correctly, your Chicago branch worksheet should look like Figure 7.

Column Widths

Move the cell pointer to B:B20, and take a close look at this cell. 1-2-3 is displaying asterisks (*) across the entire width of the cell. This means that the number 1-2-3 is attempting to display is too wide to fit within the boundaries of the cell. Enter /Worksheet Column Set-Width 10 [ENTER]. 1-2-3 will widen the column width of column B to ten characters. The number in cell B:B20 should now be displayed properly, and your worksheet should look like Figure 8.

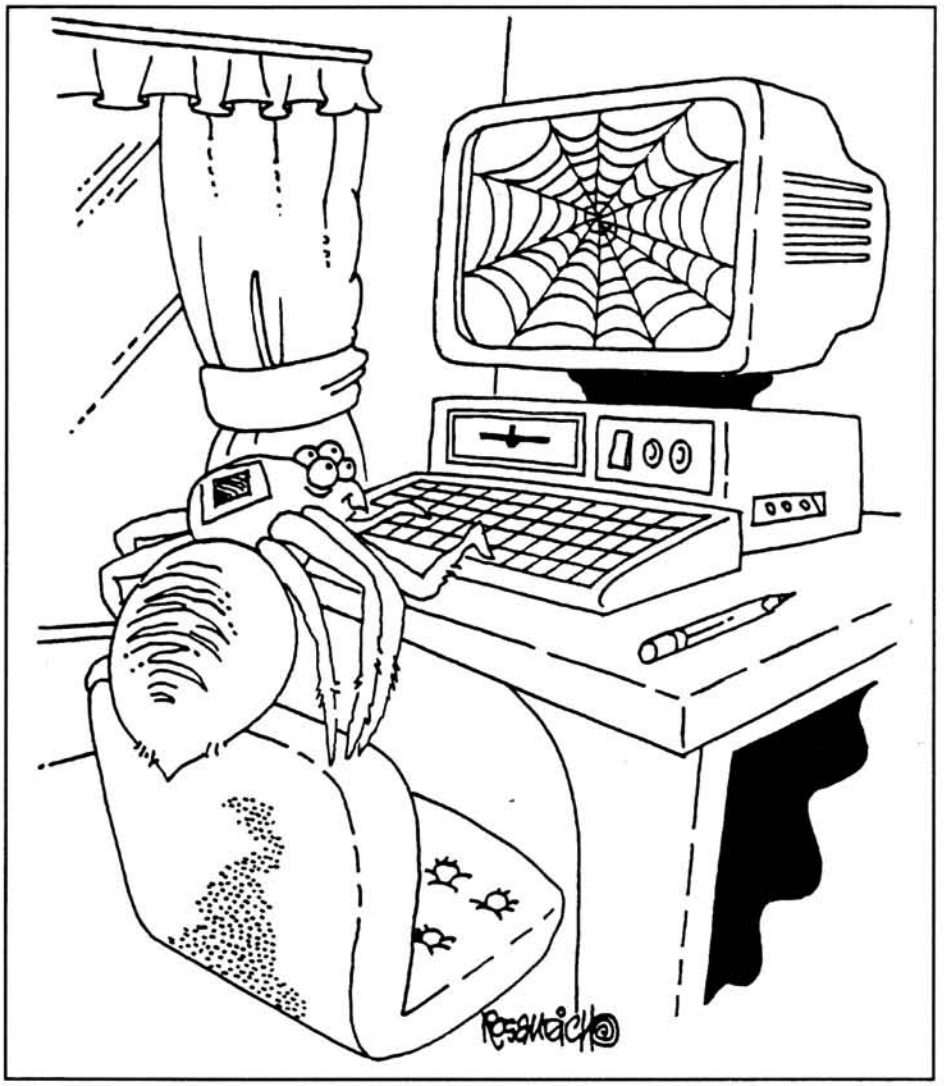
1-2-3 allows you to set column widths individually or globally. By using individual column widths, each column only takes up as much space as is needed. This feature alone was enough to make me switch from VisiCalc to SuperCalc (CP/M) back in my early spreadsheet days. It became impossible to generate clear, concise reports when only one column width was available throughout the entire worksheet. With 1-2-3, you can vary the column width between 1 and 240 characters.

Set the column width for columns B, C and D to 10 characters for each of the individual branch worksheets. Remember that you must move the cell pointer to the column you want to change, since 1-2-3 changes the width of the column that contains the cell pointer. When you are done, move the cell pointer to sheet A. Enter /Worksheet Global Col-Width 10 [ENTER]. This command sets the global column width to ten characters from 1-2-3's default of nine. If you want one or more columns to be a different width, you can still set them individually.

Your individual branch worksheets should now be complete. To save your work, enter /File Save [ENTER] Backup. The Backup option saves your old file with a .BAK extension, and saves the new file as XYZ91SUM.WK3.

Next Time

In the next part of this article you will learn how to bring the data from the individual branch worksheets to the summary worksheet. You also will learn how to add some additional WYSIWYG enhancements to make the summary worksheet even more appealing. Most of the mundane chores have been completed; what remains is to see what 1-2-3 can do with the data you entered. ✱



Tracking Toxins:

Researcher Uses Bees and PCs to Study Environmental Hazards

David Dalton
Manager, Corporate Communications
Zenith Data Systems

Dr. Jerry Bromenshenk, of the University of Montana, examines a frame containing a bee colony.

Monitoring environmental contaminants may seem like an unusual role for honey bees, but it is one to which they are well-suited.

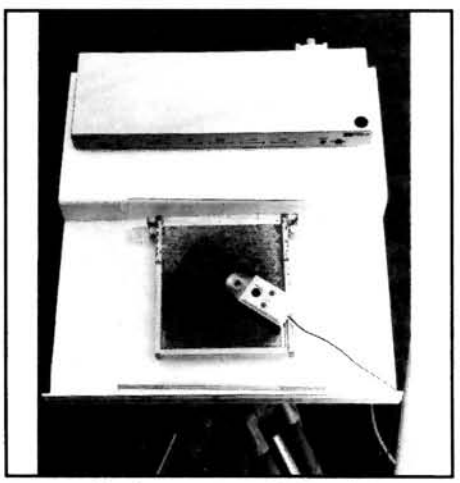
As they search for nectar, pollen and water, bees also encounter — and carry back to their hives — toxic chemicals and small amounts of other hazardous materials, which researchers can recover and analyze.

Each year, between three and four million colonies of honey bees are transported across the country to pollinate crops in fields, orchards and vineyards. Because they are kept throughout the U.S. in urban and rural areas, bees can sample literally thousands of locations in a short amount of time.

Dr. Jerry Bromenshenk, professor of research in the University of Montana's division of biological sciences, studies honey bees to detect hazards in the environment.

"Bees are like flying dust mops,: he said. "They bring back to their hives traces of whatever happens to be in a given location."

Bromenshenk said that colonies can become contaminated in two ways. The bees' foraging activities put them in contact with airborne pollution that can be inhaled, adhere to their bodies or becomes mixed with pollen grains. Contamination also can occur through the water bees drink and with which they control the temperature and humidity of their hives.



Bromenshenk uses a sonic digitizer linked to a Zenith Data Systems SupersPort 286 laptop to assess bee colony dynamics.

To simulate bee population dynamics and the effects of contamination on bee colonies, Bromenshenk uses a laptop PC and a software program called PC BEEPOP. The program, which was developed by Bromenshenk and his associates and supported by the U.S. Environmental Protection Agency through its Corvallis Environmental Research Laboratory, aids researchers in predicting a bee population's responses to environmental variables such as pollution.

Also included in the software is BEETOX, a toxicology database with information on more than 400 chemicals and their effects on pollinators. Manufacturers are required by federal law to disclose such data.

PC BEEPOP has pull-down color menus, on-line help, automatic error checking and graphics that make it easy to use. Although the program requires minimal technical expertise, it is designed primarily for advanced applications, such as ecological assessments of hazardous waste sites and pesticide applications.

PC BEEPOP consists of a feedback system of interdependent elements that can be changed to study variables such as climate, disease-causing agents and pollution, and their effects on the queen, workers and drones in a hive.

When BEETOX is used in conjunction with field and laboratory tests, the PC BEEPOP model provides an effective tool for assessing population responses to variables such as changes in temperature, examining a colony's responses to diseases and predators, and ranking waste sites according to toxicity.

"Because many factors can influence test results, it is often difficult to determine cause and effect," Bromenshenk said. "Measurement of a bee colony can reveal effects such as mortality, shortened life span, and disrupted foraging or housecleaning patterns. By testing for toxicity, scientists can make inferences about such events and their relationship to the ecosystem."

Bromenshenk and his colleagues demonstrated the potential of bees as environmental monitors in a study of the potential

of bees as environmental monitors in a study of the Puget Sound region of Washington State. They asked 81 beekeepers in urban and rural areas to collect samples and perform measurements for contaminants at 96 sites. After collecting and analyzing data, Bromenshenk and his colleagues identified sites from which high concentrations of pollutants emanated.

"Our results showed that beekeepers can effectively use colonies of bees as a self-sustained system for environmental monitoring over large geographical areas," he said.

Bromenshenk also studies bee colony dynamics such as brood rearing, hoarding behavior and storage of food, and has sought more efficient ways to determine how each cell in a hive is used. In the past, researchers have used grid overlays, visual estimates, photography and combinations of these methods to arrive at approximate measurements.

"Counting each cell is the most acurate method, but it is a time consuming process," he said.

To assure accuracy and reduce the time involved, he now uses a sonic digitizer that traces each cell within a honeycomb. The digitizer emits sonic pulses that are picked up by sensitive microphones. A counter measures the time it takes a sound signal to reach a sensor, and a built-in microprocessor uses simple geometry to calculate the equivalent distance moved and convert the signal to Cartesian coordinates. According to Bromenshenk, the digitizer is accurate to 1/100 of a centimeter. A software program determines how the cells are used, and sends data to Bromenshenk's laptop.

"With the digitizer, measurement can now take place on-site, as well as in the laboratory," he said.

Bromenshenk notes that although bees are exposed to a variety of toxins, there has been little effect on the quality of the honey they produce. "In spite of our best efforts to find contaminants, we have found that honey is a remarkably clean food product," he said.

Bromenshenk uses a sonic digitizer linked to a Zenith Data Systems SupersPort 286 laptop to assess bee colony dynamics.

Continued from Page 20

PIN	SIGNAL	Active State	FDD I/O	PIN	SIGNAL	Active State	FDD I/O
1	Ground	-	-	18	DIRECTION (Step In)	Low	Input
2	High Density Select	High	Input	19	Ground	-	-
3	Ground	-	-	20	STEP	Low	Input
4	NC	-	-	21	Ground	-	-
5	Ground	-	-	22	WRITE DATA	Low	Input
6	NC	-	Output	23	Ground	-	-
7	Ground	-	-	24	WRITE GATE	Low	Input
8	INDEX	Low	Output	25	Ground	-	-
9	Ground	-	-	26	TRACK 0	Low	Output
10	MOTOR ENABLE 1	Low	Input	27	Ground	-	-
11	Ground	-	-	28	WRITE PROTECT	Low	Output
12	DRIVE SELECT 2	Low	Input	29	Ground	-	-
13	Ground	-	-	30	READ DATA	Low	Output
14	DRIVE SELECT 1	Low	Input	31	Ground	-	-
15	Ground	-	-	32	HEAD 1 SELECT	Low	Input
16	MOTOR ENABLE 2	Low	Input	33	Ground	-	-
17	Ground	-	-	34	DISK CHANGE	Low	Output

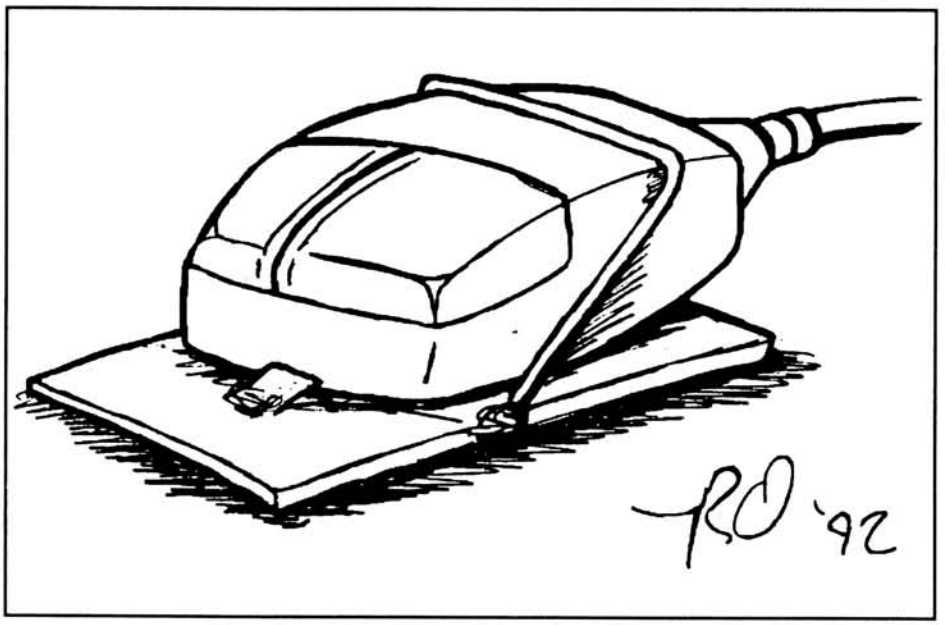
Figure 2. Typical Pin Assignments on Floppy Drives.

some money on disks, it is a savings that is at the risk of the integrity of the data being stored on the disk. Because the media is specifically designed for the intended capacity, use of 720K disks at the 1.44M data density virtually guarantees the user that data will eventually be lost and is a practice that's strongly discouraged.

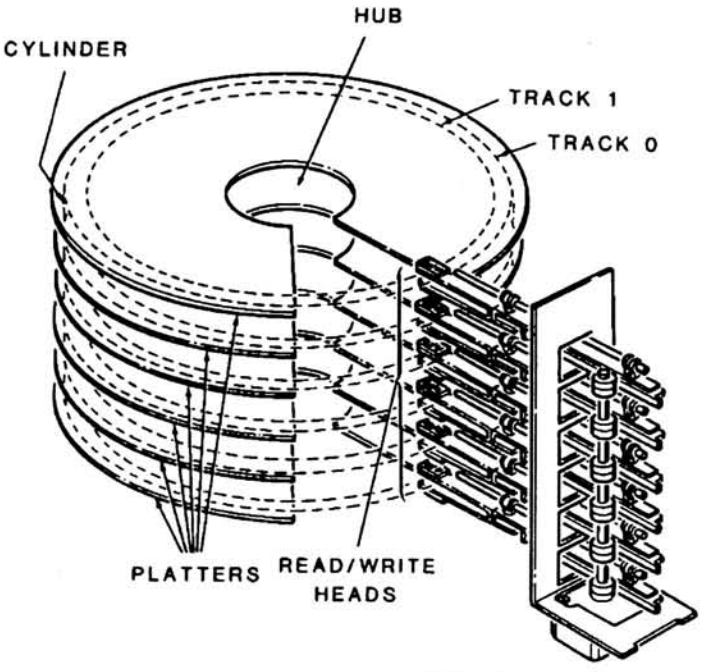
Write protection

Another hardware feature built into virtually all floppy drives is the write protection mechanism. It is standard for a disk to have a means to protect itself: 5¼" disks have a slot that can be covered with a write protect tab and 3½" disks have a hole with a sliding door to do the same thing. The disk drive has either a mechanical or optical means of detecting the state of the disk's write protect and the drive's electronics are designed such that it is impossible to write to a disk that has been so protected. This prevents the drive from writing or formating a disk that has been write protected.

The next article will go into detail about the Floppy Disk Controller: how it operates, the interface with the CPU, how it can control more than one floppy drive, and how it tells the Floppy Drive what to do.



Drive Lessons



A Guide to PC Hard Disk Technology

David R. Veit
51640 Bitterly Court
Granger, IN 46530

Overview

Most of us have watched in amazement as PC technology has exploded in the past decade. We have seen practically everything associated with the PC industry become faster, denser, more complex and efficient, while physical size and price have plunged. The evolution of hard disk drives is no different.

Eleven years ago when IBM released its first PC, hard drives with the capacity and price as we know them today simply did not exist. Sure, drives with multiple disk platters existed in the world of IBM mainframes and DEC minicomputers. But drive technology was in another era and the notion of providing a reasonably priced hard disk drive in a small package for these new personal desktop computers from IBM and others, such as Heath Company, was just out of the question. However, we now take for granted something entirely unthinkable just a few years ago. The industry has provided us with easily affordable drives in the 60Mb or 80Mb range and beyond for our home computers!

This is the first in a series of articles intended not to take us on a nostalgic trip to the past. It will attempt to explain just what is involved in the world of hard drives. I will walk through the fundamentals of the hard drive, including such topics as the physical makeup of drives, performance, BIOS issues, controller interface types, DOS structuring, and finally a section on how to recover data from a disk disaster.

Fundamentals

How many of us know our hard drives simply as a DOS prompt " C:> "? There is certainly more to C:> than knowing you have a computer closet that holds a specified amount of junk files. But how is this closet organized to hold all of the junk we accumulate? In our DOS world, hard drives (and floppies, too) are organized by the operating system in a format consisting of sectors and tracks. And, with the exception of some optical disk drives that have become available, the drives in our PCs are based on principles of electromagnetics. What does all of this mean? It is easier to grasp the concept with a picture of words. Think of your drive as having one or more rotating disk platters with their surfaces coated with a magnetic material. Also, imagine an arm, somewhat similar to a phonograph player's arm, moving back and forth across the surface of each platter. This drive arm has a device attached to the end of it, a magnetic read/write head, that is capable of detecting or making minute changes in magnetic fields close by.

This magnetic head actually rides on a cushion of air above the rotating drive platter without actually touching it. From this position, it is able to read data from the drive by sensing the magnetic field associated with the magnetic material passing below it. Writing data to the disk is similar, this time the magnetic head alters the pattern of the magnetic material on the platter.

The actual organization of the magnetic data on the platter is highly regimented. The platter is divided into a series of concentric circles or rings, called tracks, (groups of identically positioned tracks on multiple platter sides are commonly referred to as "cylinders") with the actual number of tracks on any given platter being dependent on the size and manufacturer of the drive. Keep in mind that these tracks do not touch each other, they are not like a phonograph record which has a single, spiral groove. Each of these tracks is in an area where the drive's head can move to read or write data. The drive's data must reside in the defined track area and must fall directly below the area where the drive head is positioned. If the head and track are not properly aligned, then data cannot be accurately transferred.

The disk organization is broken down further into groups called sectors. Think of the platter as being grouped into areas similar to the way a pie or pizza is sliced. Another way to describe the result is to visualize the individual circular tracks being sliced into smaller semi-circular tracks. Since the sector area is larger on the outer portion of the disk platter as compared to the inner portion, the inner tracks are "shorter" than the outer tracks. The consequence of this is that the inner tracks must either hold less data than their outer cousins, or hold the same amount of data but in a more condensed fashion. In DOS systems, sectors on inner tracks hold the same amount of

data as sectors on outer track.

In practice, many disk drives have multiple platters rotating within them, with multiple read/write heads. The exact configuration of sectors, tracks, platters, and heads is highly dependent on the drive and controller type, its size, and manufacturer.

Physical Components

The primary components of a hard disk are its platters, read/write heads, and head actuator mechanism. These are physically contained within a sealed chamber on the drive assembly itself and are commonly referred to as the Head Disk Assembly (HDA). This chamber is rarely opened, about the only time it should be opened is when a firm specializing in recovering data from a failed unit opens it in a special "clean room". It is certainly never opened by the user. Doing so is an invitation to permanent data loss. Other electrical components make up the drive assembly, the amount of circuitry present is dependent on the type of PC host interface it uses.

Disk Platters

Drives contain one or more disk platters, where physical size of the drive is usually the determining factor. When multiple platters are present, they are attached and locked to the same spindle axis and rotate together at the same speed. The key to these platters holding data is that they are coated with a thin layer of magnetic substance, or media. The two popular types of magnetic substance are oxide media and thin film media.

Oxide media is more common in older drives, and in lower cost new drives. It is a very soft coating which makes it very prone to damage when head crashes occur, i.e. where the read/write head actually comes into contact with the platters. Thin film media is much more common in newer, higher-capacity drives. It allows for a greater density of data storage than oxide media and thanks to its very hard surface, makes the platter more capable of resisting head-crash damage.

Read/Write Heads

The read/write heads are the drive components responsible for transferring data to/from the drive's platters. There are various techniques used to mount the heads to the arm assembly, but the concept to remember is that the heads are meant to ride on a cushion of air above the rotating platter. The gap between head and platter is very small, on order of 5-20 millionths of an inch. This small distance is the reason why you don't want to bump your drive during operation. An accidental bump can cause the head to move slightly, causing it to strike the platter, and resulting in what is commonly known as a "head crash". It does not take an Einstein to realize that when a head comes into contact with a platter rotating at 3,600 RPM, it is likely to cause damage to the platter's media. Often, the resulting crash will disperse minute media fragments within the sealed chamber area, ultimately destroying the ability to read data from the drive.

There are safeguards implemented to prevent media damage on drives, especially on smaller drives that are used in portable computers. Some drives require use of a special DOS utility that will "park" the head assembly in a "landing zone" and lock it in place so that the head does not rattle against media areas containing data. The landing zone is the area on a platter that is off-limits to data and is designated as the area where the heads will land once the drive is turned off. Recall that the heads will ride above the rotating platter during normal operation, but must come down somewhere once the drive is turned off. Some drives automatically perform this parking function every time they are turned off. However, this varies among the different drive types and vendors. Another term for the process of parking the heads in a safe place is "shipping" the drive heads.

Like the disk platters, there are also two types of read/write heads common to PC drives: composite ferrite and thin film heads. Thin film heads are a part of newer technology and are more desirable than composite ferrite heads primarily because they are lighter in weight. This trait allows them to fly closer to the spinning platters, resulting in denser data storage on the platters. Another factor of their lighter weight is that thin film heads are simply less likely to crash than the heavier composite ferrite heads.

Head Actuator Mechanism

Probably the most important component within your disk drive is the head actuator mechanism. This is the assembly that positions the read/write heads over the desired data track on the platter. Since this is the most critical moving mechanical part in the drive, its speed, accuracy, reliability, and ability to withstand shock, vibration, and temperature change is absolutely critical to drive performance.

In the world of hard drives, there are two types of actuator mechanisms: stepper motor and voice coil. Without any argument, the voice coil actuator type is superior. In virtually every category of importance, the voice coil outperforms the stepper motor type. It is faster, is not affected by temperature, and is more reliable.

Stepper motors, as used in all floppy drives and most hard drives under 40 Mb, are electrical motors that move the head and arm assembly across the surface of the platter in discrete "steps" according to defined mechanical detents (stopping mechanisms), usually determined by the distance between data tracks on the platter. The motor cannot position the heads in between these defined steps. This becomes a problem when the disk platters begin to heat after prolonged operation. The platters expand with heat, causing data tracks to move slightly in relation to their original position. Since the heads cannot compensate less than a single incremental step, they cannot correct for temperature-induced errors. The consequence is that the stepper motor will not properly align the head with the data track, causing read errors.

Voice coil mechanisms are superior for several reasons. They operate on slightly different principles, the primary difference being that there are no predetermined "steps" or detents governing the movement of the head assembly. Instead, the voice coil is an electromagnetic device which slides the head assembly "smoothly" across the surface of the platter. It accomplishes this by using a special "index" head that is able to sense the position of special, magnetic indexing marks that are permanently marked on one unused side of a platter. The index head with its associated electronics then positions itself over an index track and the read/write heads over the correct data track, and then fine tunes its position by sensing exactly where the strongest magnetics signals are located. Think of the index head as a "guide" head, since it is directly responsible for guiding the read/write head assembly to the correct data track and aligning it perfectly. This technique is not affected by temperature, instead it actually compensates for temperature induced variations in the track positioning.

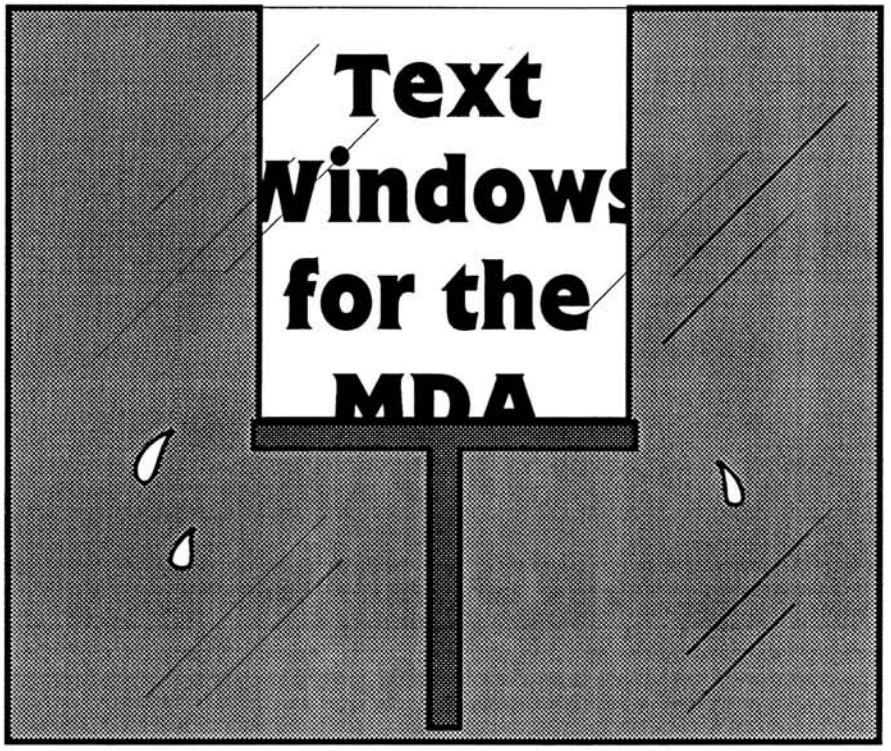
Performance

Which types of hard drives are fastest? What makes them fast? What are the factors that affect drive performance? Many of us think of drive performance simply in terms of "access time" as advertised by drive manufacturers. Is it sufficient to compare drives by this single benchmark?

Access Time

There is no question that the "average access" times touted by drive vendors are a good indication of the drive's performance capability. Access time is an indication of the average time it takes for the read/write heads to move to a given data track and sector. Generally measured in milliseconds, times can range from around 65-80 mS for slow, inexpensive drives, to less than 20mS for high performance drives. For a better indication of how important this number can be, consider a database program that performs a great number of accesses due to the random nature of records stored all over the drive. The single, 65 ms access time needed to search for one

Continued on Page 40



Robert Moon
P.O. Box 2045
Ponte Vedra. FL 32004-2045

The MDA (monochrome display adapter) operates only in the monochrome text mode, which is video mode 7. The MDA is a character-based display subsystem capable of displaying only preformed characters. The screen displays and user interfaces are usually not too appealing. They certainly are not attention getters when consisting of only text; however, improvements can be made to the character-based displays and user interface. The text can be positioned and grouped into text areas and the cursor can be hidden when not needed. The greatest improvement is to put a border around the text area, creating a text window. A borderless text window can also be made simply by changing to a contrasting background within the window area. The text can be underlined, boldface, flashing (blinking) on and off, or highlighted (reverse video). We will develop some procedures to accomplish this improved character-based display.

The Hercules Graphics Card, an alternative monochrome display adapter, has graphics capability, but in text mode we can think of it as an MDA.

ASCII and the Attribute Byte

Computers understand only numbers; consequently, there is a convention for mapping characters to numbers, known as ASCII (American Standard Code for Information Interchange). Using one byte per character gives us 256 different combinations. The first 128 bytes (0 thru 7fh or 127) is the standard ASCII character set. The second 128 bytes (80h(128) thru 0ffh(255)) is the extended ASCII character set, which is not standard. The extended ASCII set is used for math symbols, foreign characters, graphical symbols, etc. Each computer manufacturer defines their own extended set. The dot pattern for each preformed ASCII character is stored in the character generator ROM (Read Only Memory) located on the MDA. There is an attribute byte associated with each character that determines its appearance. The character can be displayed as normal video, underlined, boldface (intensify), blinking, or as reverse video. These attributes can be combined to produce ten different character appearances. Not bad for black and white (or green or amber).

The attribute byte is shown in Figure 1.

CRT Mode Control Port

Bit 7 of the attribute byte is controlled by bit 5 of the CRT mode control port at address 03b8h. If bit 5 is set, then foreground blink is enabled. This is the default setting at boot-up. If bit 5 is cleared, then foreground blink is disabled and bit 7 of the attribute byte becomes background intensity.

Figure 2 the MDA CRT mode control port (3b8h) assignments.

Characters and the Video Memory

Two bytes are required to store the character in the video memory, which is located on the MDA. One for the ASCII character code and one for the text attribute. The character is stored at even addresses and the attribute is stored at odd addresses.

Borders can be created using the block graphic characters that are part of the IBM extended ASCII character set. Table 1 shows some graphic characters and the ASCII code in decimal and hexadecimal that are suitable for drawing borders or boxes.

In monochrome text mode (video mode 7), the screen displays 25 rows by 80 columns for a total of 2000 characters. The

Bit	Attribute
7	Blink foreground or background intensity (depends on bit 5 of the CRT mode control port)
6,5,4	Background
3	Foreground intensity(boldface)
2,1,0	Foreground

7 6 5 4 3 2 1 0	Hex	Attribute
0 0 0 0 0 0 0 0	00	No display
0 0 0 0 0 0 0 1	01	Underline normal video
0 0 0 0 0 1 1 1	07	Normal video (white on black)
0 0 0 0 1 0 0 1	09	boldface underline normal video
0 0 0 0 1 1 1 1	0f	boldface normal video
0 1 1 1 0 0 0 0	70	Reverse video (black on white)
1 0 0 0 0 0 0 1	81	Blinking underlined normal video
1 0 0 0 0 1 1 1	87	Blinking normal video
1 0 0 0 1 0 0 1	89	Blinking boldface underline normal
1 0 0 0 1 1 1 1	8f	Blinking boldface normal
1 1 1 1 0 0 0 0	f0	Blinking reverse video

Note: If bit 5 of the CRT mode control port is cleared, then the attribute 0f0h will produce a bright background. The background must be white — there is no bright black.

Figure 1

Bit	Function	Note
7	0 = page 0	Hercules has two pages of video memory,
	1 = page 1	MDA has only one. Default is page 0
6	Not used	
5	0 = blink off = background intensity on	
	1 = blink on = background intensity off	Boot-up default is blink on
4	Not used	
3	0 = video off	Boot-up default is video on
	1 = video on	
2	Not used	
1	0 = text	Hercules has graphics, the MDA does not
	1 = graphics	Boot-up default is text
0	0 = 40 X 25	Boot-up default is 80 X 25
	1 = 80 X 25	

Writing 09h to port 03b8h will disable the foreground blink feature and enable the background intensity feature. Write 29h back to the port to restore the default conditions.

Figure 2

screen can be thought of as a matrix of 2000 cells or blocks that will hold one character each. Therefore, we need 4000 bytes of video memory to store one screen. The first row of 80 characters and attributes are stored in memory, using 160 bytes. The next row of 80 characters and attributes, also using 160 bytes of memory, follows sequentially. The entire screen display is stored in memory in this fashion.

The MDA video memory, also known as the video buffer, is at segment 0b000h.

Displaying a character at an exact row and column position on the screen requires that the character be put into an exact location in memory. We can't just stuff the row and column numbers into memory because two bytes of storage are needed for each character. There is a simple formula that we can use to calculate the character location in memory. This location is the character offset into the video segment at 0b000h.

FORMULA: character offset = row number * 80 * 2 + column number * 2

EXAMPLE: The memory location for the character at row 12 and column 39 is

```
12 * 160 + 39 * 2 = 1998(07ceh)
```

This is about the center of the screen. The first screen character is at memory location 0 (segment:offset-0b000:0000) and the last screen character is at memory location 3998(0f9eh) (segment:offset-0b000:0f9e).

The first character attribute is stored at memory location 1 (segment:offset-0b000:0001) and the last character attribute is stored at memory location 39999(0f9fh) (segment:offset-0b000:0f9f).

Borders

Drawing a box or border on the screen is similiar to drawing one on paper. We first calculate the memory location of the upper left corner and put the code for the upper left corner graphic character into memory at that location. We draw the left side by moving down one row and putting the code for the vertical character into memory. We continue moving down one row at a time putting the vertical character code into memory until the desired height is reached.

The bottom is drawn by putting the code for the lower left corner into memory and then moving one column to the right, putting the code for the horizontal character into memory. We continue moving one column to the right putting the horizontal character code into memory until the desired length is reached.

We draw the right side by putting the character code for the lower-right corner into memory and then moving up one row at a time putting the vertical character code into memory until the desired height is reached.

The top is then drawn by putting the upper right corner character code into memory. We then move to the left one column at a time putting the horizontal character code into memory until we are back to the starting point - the upper left corner.

When we move across the screen to the right we add one column as many times as necessary. To move left, we substract one column. We can't add one row to move down the screen because one row is stored in memory as 160 columns. We must add 160 columns to move down one row, and to move up, we substract 160 columns.

Because the screen starts at zero and uses two bytes for storage, we'll add 158 columns and substract 162 columns, per row.

We can create boxes or borders of any size that will fit on the screen and locate them anywhere on the screen. Since the MDA updates the screen fifty times a second, it is fast.

Decimal	Hex	Character
176	B0	▒
177	B1	▓
178	B2	▓
179	B3	│
180	B4	┤
181	B5	╡
182	B6	╢
183	B7	╖
184	B8	╕
185	B9	╣
186	BA	║
187	BB	╗
188	BC	╝
189	BD	╜
190	BE	╛
191	BF	┐
192	C0	└
193	C1	┴
194	C2	┬
195	C3	├
196	C4	─
197	C5	┼
198	C6	╞
199	C7	╟
200	C8	╚
201	C9	╔
202	CA	╩
203	CB	╦
204	CC	╠
205	CD	═
206	CE	╬
207	CF	╧
208	D0	╨
209	D1	╤
210	D2	╥
211	D3	╙
212	D4	╘
213	D5	╒
214	D6	╓
215	D7	╫
216	D8	╪
217	D9	┘
218	DA	┌
219	DB	▌
220	DC	▄
221	DD	▌
222	DE	▐
223	DF	▀

Table 1

Shadows

Shadows can be used to accent the border. They can be put on the right or left side. The left shadow is drawn by moving left one column and down one row, putting the first shadow character code into

memory, and continuing down, row by row, putting the vertical character code into memory. The bottom of the shadow is drawn by moving right one column at a time until one column to left of the lower right corner of the border is reached, then putting the horizontal character code into memory. The right shadow is drawn similiarly, but first, we have to move right to one column past the upper right corner of the border. The vertical character code is put into memory as before and the horizontal character code is put into memory moving to the left one column each time. Shadows are drawn much the same as borders but have only one side and the bottom. The shadow height has one more row than the border height.

This method of writing to the screen is known as direct video programming. It is possible to use the BIOS or DOS functions to draw boxes or borders, but it is slow.

BIOS Video Functions

We will, however, use some functions from the BIOS video routines (interrupt(int) 10h). These will be used to clear the screen, move the cursor, turn the cursor off and on, to erasecharacters, and to print a character string with attribute.

Clear the Screen

We use the scroll page up function, number 6, in the clear_screen procedure. This function scrolls text upward a specified number of lines in a specified window. By defining the window as the entire screen, it will be cleared as blank lines are inserted at the bottom and scrolled up. Specifying zero for the number of lines to scroll up will cause the entire screen to blank. The blank lines can have the attribute of normal or reverse video.

A window without borders can be made with this function by using the reverse attribute as the background. For example, if the background is black, after specifying the window, scroll up using the reverse attribute. This will create a white window on a black background. Scrolling up with lines having the normal attribute on a white background will create a black window.

If you clear bit 5 of the CRT mode control port, you can have dim white windows and bright white windows.

Cursor On and Off

The set cursor size and blinking rate characteristics function, number 1, is used to turn the cursor off and on. We will be concerned with only bits 5 and 6 of the ch register. They are the ones that control the cursor blinking rate characteristics. They are shown below:

Bit 6	5	Characteristic
0	0	normal (medium)..boot-up default
0	1	invisible
1	0	fast
1	1	slow

From the above table, assuming a normal cursor blink rate, we see that setting bit 5 will turn the cursor off. Conversely, clearing bit 5 will turn the cursor on.

For the curious, bits 0 - 3 of ch register control the cursor starting scan line. The MDA default is 11. Bits 0 - 3 of cl register control the cursor ending scan line. The MDA default is 12. This is the two line cursor near the bottom of the character block. The MDA cursor has available 14 lines, numbered from top to bottom (0 - 13).

The cursor size can be varied by changing the starting and ending scan lines. Starting at 0 and ending at 13 will create a large block cursor. You can speed up or slow down the blink rate by altering bits 5 and 6 as shown above.

We will use the get cursor position and characteristics function, number 3, to get the current settings. The blinking characteristics is returned in ch register. ORing ch register with 20h will turn the cursor off. ANDing ch register with 0dfh will turn the cursor on. This way we won't alter other settings by mistake. Remember that this applies only to the normal blink rate. If you change the rate to slow or fast, you can't use 20h and 0dfh.

Move Cursor

The set cursor position, number 2, is used by the goto_xy procedure to put the cursor anywhere on the screen. The row number is put into dh register and the column number is put into dl register. Rows can be 0 to 24 and columns can be 0 to 79.

Print and Erase

The print_string procedure uses the write character and attribute at the cursor function, number 9. The character is put into al register and the attribute is put into bl register. The cursor is not automatically advanced, so we will have to handle that chore.

Likewise, we'll use this function to erase characters by printing spaces with the same attribute as the character we want to erase. We put the number of spaces to write in cx register and the attribute in bl register.

Program

Listing 1 is a program showing the procedures with a demonstration of how they are used.

The procedure main is a demonstration of the procedures we developed. You are asked to press any key to proceed to the bright screen demo and again to return to DOS.

This is a small number of the different windows that can be produced. You can modify the the values as you desire to create different borders. A flashing border gets attention as does a flashing message. These are good for announcing errors. The underlined and highlighted (reverse) attribute will emphasize words or phrases. Like "they" say, your display and interface screens are limited only by your imagination.

Now we can have attractive displays and interfaces that have the polished look of a professional package.

```
;                   Procedures to create text windows for the MDA and
;                   a demonstration of a few windows
;
;
;                           Written by Robert Moon, 3 Sep 1992
;
;                           Copyright (c) 1992 Robert Moon

title    Txtwnd

.model   small


;Equates

;Monochrome text attributes

UL_NOR              equ     01          ;Underlined, normal video
NOR                 equ     07          ;Normal video (white on black)
HUL_NOR             equ     09          ;Boldface underlined characters, normal video
HL_NOR              equ     0fh         ;Boldface characters, normal video
REV                 equ     70h         ;Reverse video (black on white)
BUL_NOR             equ     81h         ;Blinking underlined characters, normal video
BL_NOR              equ     87h         ;Blinking characters, normal video
BHUL_NOR            equ     89h         ;Blinking boldface underlined characters
BHL_NOR             equ     8fh         ;Blinking boldface characters, normal video
BR_BG_REV           equ     0f0h        ;Blinking characters, reverse video
                                        ; or bright background, reverse video


;Shadow orientation

RGT                 equ     0           ;Shadow on the right side of border
```

```
LFT             equ     1       ;Shadow on the left side of border

;Structures

;Border definition

BORDER  struc
        ATTRIBUTE       db      ?       ;Text attribute byte

        TOP_RIGHT_ROW   db      ?       ;Top right corner
        TOP_RIGHT_COL   db      ?       ; of the border
        TALL            dw      ?       ;Height of border sides
        WIDE            dw      ?       ;Length of border top and bottom
BORDER  ends

;Pattern definition

PATTERN struc
        VERT            db      ?       ;Vertical line character
        TOP_HORZ        db      ?       ;Top horizontal line character
        BOT_HORZ        db      ?       ;Bottom horiz line character
        TL_CNR          db      ?       ;Top left corner character
        BL_CNR          db      ?       ;Bottom left corner character
        BR_CNR          db      ?       ;Bottom right corner character
        TR_CNR          db      ?       ;Top right corner character
PATTERN ends

;Shadow definition

SHADOW  struc
        O_SHAD          db      ?       ;Shadow orientation
        A_SHAD          db      ?       ;Shadow attribute
        V_SHAD          db      ?       ;Shadow vertical character
        H_SHAD          db      ?       ;Shadow horizontal character
        CA_SHAD         db      ?       ;Shadow start corner character
        CB_SHAD         db      ?       ;Shadow middle corner character
        CC_SHAD         db      ?       ;Shadow end corner character
        T_SHAD          dw      ?       ;Shadow height
        W_SHAD          dw      ?       ;Shadow width
SHADOW  ends


.code

MAIN            proc    near

                mov     ax,_data        ;Data segment
                mov     ds,ax
                mov     ax,0b000h       ;Mono video memory segment
                mov     es,ax
                call    CLEAR_SCREEN    ;Clean the screen
                call    CURSOR_OFF      ;Hide cursor

;Create a large blinking border

                mov     si,offset LGE_BDR       ;Border data
                mov     bx,offset LGE_PAT       ;Pattern data
                call    DRAW_BORDER

;Print large border message

                mov     dx,0119h        ;Move cursor
                call    GOTO_XY
                mov     si,offset LGE_M ;Point to messsage
                mov     bl,REV          ;Reverse video
                call    PRINT_STRING

;Create border with right shadow

                mov     si,offset R_BDR ;Border data
                mov     bx,offset R_PAT ;Pattern data
                call    DRAW_BORDER
                mov     si,offset R_SHAD        ;Shadow data
                call    DRAW_SHADOW

;Create border with left shadow

                mov     si,offset L_BDR ;Border data
                mov     bx,offset L_PAT ;Pattern data
                call    DRAW_BORDER
                mov     si,offset L_SHAD        ;Pattern data
                call    DRAW_SHADOW

;Create a  window with the scroll up function with dim background

                mov     ah,6            ;BIOS scroll up function
                mov     al,8            ;Number of  lines to scroll up
                mov     bh,REV          ;Attribute of fill background
                mov     ch,10           ;Top row
                mov     cl,32           ;Left column
                mov     dh,18           ;Bottom row
                mov     dl,72           ;Right column
                int     10h             ;BIOS video service

;Pause to show the screen

                mov     dx,1800h        ;Position cursor
                call    GOTO_XY
                mov     dx,offset PSE1_M        ;Point to bright message
                mov     ah,9            ;BIOS function to print strings
                int     21h
                mov     ah,7            ;BIOS function to wait for key
                int     21h

;Erase the messsage.... don't need it any more

                mov     dx,1800h        ;Start of message position
                call    GOTO_XY
                mov     bl,NOR          ;Normal video erase attribute
                mov     cx,NPSE1        ;Number  of  chars to erase
                call    ERASE

;Disable foreground blink and enable background intensity

                mov     dx,03b8h        ;MDA CRT control port address
                mov     al,09h          ;Clear bit 5;set bits 3 and 0
                out     dx,al           ;Send it out

;Erase large border message

                mov     dx,0119h        ;Position cursor at message
                call    GOTO_XY
                mov     bl,NOR          ;Attribute for erased area
                mov     cx,NLGE         ;Number of characters to erase
                call    ERASE

;Print new large border message

                mov     dx,0119h        ;Reposition cursor
                call    GOTO_XY
```

```
                mov     si,offset RLGE_M        ;Point to new message
                mov     bl,REV                  ;Character attribute
                call    PRINT_STRING

;Create a window using the scroll up function with a bright background

                mov     ah,6                    ;Scroll  up function
                mov     al,4                    ;Number of lines to blank
                mov     bh,BR_BG_REV            ;Attribute for reverse video
                mov     ch,12                   ;Top row
                mov     cl,34                   ;Left column
                mov     dh,16                   ;Bottom row
                mov     dl,70                   ;Right column
                int     10h

;Message in the bright window

                mov     dx,0f27h                ;Cursor  position
                call    GOTO_XY                 ;Move cursor
                mov     si,offset BRT_M         ;Address of message
                mov     bl,BR_BG_REV            ;Character attribute
                call    PRINT_STRING            ;Print it

;Pause to show  display

                mov     dx,1800h                ;Put cursor out  of the way
                call    GOTO_XY
                mov     dx,offset PSE_M         ;Point to pause message
                mov     ah,9                    ;DOS function to print string
                int     21h
                mov     ah,7                    ;DOS function to await key
                int     21h

;Erase message

                mov     dx,1800h                ;New cursor position
                call    GOTO_XY
                mov     bl,NOR                  ;Attribute
                mov     cx,NPSE                 ;Number of chars to erase
                call    ERASE

;Enable foreground blink...restore the default setting

                mov     dx,03b8h                ;CRT mode control port
                mov     al,29h                  ;Set bits 5, 3, and 0
                out     dx,al

;Change large border message again

                mov     dx,0119h                ;New cursor position
                call    GOTO_XY
                mov     cx,NRLGE                ;Amount to erase
                mov     bl,REV                  ;Attribute
                call    ERASE                   ;Erase old message
                mov     dx,0119h                ;Reposition cursor
                call    GOTO_XY
                mov     si,offset LGE_M         ;Point to new message
                mov     bl,REV                  ;Attribute
                call    PRINT_STRING

;Change bright background message

                mov     dx,0f27h                ;New cursor position
                call    GOTO_XY
                mov     cx,NBRT                 ;Number of chars to erase
                mov     bl,BR_BG_REV
                call    ERASE                   ;Erase old message
                mov     dx,0f27h                ;Reposition cursor
                call    GOTO_XY
                mov     si,offset RBRT_M        ;New message
                mov     bl,BR_BG_REV            ;Attribute
                call    PRINT_STRING

;Exit

                mov     dx,1800h                ;Put cursor out of the way
                call    GOTO_XY
                call    CURSOR_ON               ;Show cursor
                mov     ax,4c00h                ;Go to DOS
                int     21h

MAIN            endp

;Draw a border on screen

DRAW_BORDER     proc    near

;Compute location
                mov     dh,[si].TOP_RIGHT_ROW   ;Border top
                mov     dl,[si].TOP_RIGHT_COL   ; right corner
                mov     ax,160                  ;2 bytes for attr and char 80*2
                mul     dh                      ;Row * columns = screen row
                shl     dl,1                    ;Column * 2 (2 bytes required)
                xor     dh,dh                   ;Can't add bytes to words
                add     ax,dx                   ;Col + row = attr+char location
                mov     di,ax                   ;Need location in di

;Top left corner

                mov     ah,[si].ATTRIBUTE       ;Text attribute
                mov     al,[bx].TL_CNR          ;Top left corner character
                stosw

;Left side

                mov     al,[bx].VERT            ;Vertical line character
                mov     cx,[si].TALL            ;Number of line characters
LEFT_SIDE:
                add     di,09eh                 ;Next row down, add 158 col
                stosw
                loop    LEFT_SIDE               ;Put all in a line downward

;Lower left corner

                add     di,09eh                 ;Next row down, add 158 col
                mov     al,[bx].BL_CNR          ;Lower left corner Character
                stosw
;Bottom

                mov     al,[bx].BOT_HORZ        ;Horizontal line character
                mov     cx,[si].WIDE    ;Number of line characters
                cld                             ;Move to  the right
                rep     stosw                   ;Put all in a line

;Lower right corner

                mov     al,[bx].BR_CNR          ;Lower right corner character
                stosw

;Right side
```

```
                mov     al,[bx].VERT            ;Vertical line character
                mov     cx,[si].TALL            ;Number of line characters
RIGHT_SIDE:
                sub     di,0a2h                 ;Next row up, sub 162 col
                stosw
                loop    RIGHT_SIDE              ;Put all in a line upward

;Top right corner

                sub     di,0a2h                 ;Next row up, sub 162 col
                mov     al,[bx].TR_CNR          ;Top right corner character
                stosw

;Top

                sub     di,4                    ;Move left two columns
                mov     al,[bx].TOP_HORZ        ;Horizontal line character
                mov     cx,[si].WIDE    ;Number of horz characters
                std                             ;Move to the left
                rep     stosw                   ;Put all in a line
                cld                             ;Clear direction flag
                ret

DRAW_BORDER     endp


;Put shadow on border

DRAW_SHADOW             proc    near

                mov     al,[si].O_SHAD          ;Get shadow orientation
                or      al,al                   ;Right or left?
                jnz     LEFT_SHAD               ;It's left

;Right side

                mov     ah,[si].A_SHAD          ;Get attribute
                mov     cx,[si].T_SHAD          ;Get height
                mov     bx,[si].W_SHAD          ;Get length
                shl     bx,1                    ;Width X 2; 2 bytes per storage
                add     di,bx                   ;Move to right side of border

;Start corner

                add     di,4                    ;Move right two columns
                mov     al,[si].CA_SHAD         ;Start corner character
                stosw
RIGHT:
                add     di,09eh                 ;Next row down, add 158 col
                mov     al,[si].V_SHAD          ;Vertical line character
                stosw                           ;Put into memory
                loop    RIGHT                   ;Do all

;Middle corner

                add     di,09eh                 ;Next row down, add 158 col
                mov     al,[si].CB_SHAD         ;Get middle corner Character
                stosw                           ; into memory

;Bottom

                sub     di,4                    ;Move left two columns
                mov     al,[si].H_SHAD          ;Get horizontal character
                mov     cx,[si].W_SHAD          ;Get length
                std                             ;Move to left
                rep     stosw                   ;Put all into memory
                cld                             ;Restore up direction

;End corner character

                mov     al,[si].CC_SHAD         ;Get end corner character
                stosw
                ret

;Left shadow

LEFT_SHAD:
                mov     ah,[si].A_SHAD          ;Shadow attribute
                mov     cx,[si].T_SHAD          ;Number of vertical lines

;Start corner

                sub     di,2                    ;Move left one column
                mov     al,[si].CA_SHAD         ;Start corner character
                stosw

;Left side

LEFT:
                add     di,09eh                 ;Next row down, add 158 col
                mov     al,[si].V_SHAD          ;Vertical line character
                stosw                           ;Put into memory
                loop    LEFT                    ;Put all in a line downward

;Middle corner

                add     di,09eh                 ;Next row down, add 158 col
                mov     al,[si].CB_SHAD         ;Middle left corner Character
                stosw                           ;Into memory

;Bottom

                mov     al,[si].H_SHAD          ;Horizontal line character
                mov     cx,[si].W_SHAD          ;Number of line characters
                cld                             ;Move to  the right
                rep     stosw                   ;Put all in a line

;End corner

                mov al,[si].CC_SHAD             ;End corner character
                stosw
                ret

DRAW_SHADOW     endp


;Blank the screen

CLEAR_SCREEN            proc    near

                xor     cx,cx                   ;Upper left corner (0,0)
                mov     dh,24                   ;Bottom is row 24
                mov     dl,79                   ;Right side is column 79
                mov     bh,07                   ;Normal video attribute
                mov     ax,0600h                ;Scroll up whole screen
                int     10h                     ;BIOS service
                ret

CLEAR_SCREEN            endp

;Position the cursor – dh holds row(y)  and dl holds column(x)
```

```
GOTO_XY         proc    near

                mov     bh,0            ;Display page 0
                mov     ah,2            ;Set cursor position
                int     10h
                ret

GOTO_XY         endp

;Turn cursor off... normal blink rate

CURSOR_OFF      proc    near

                mov     ah,3            ;Get cursor scan line info
                int     10h             ; and blink characteristics
                or      ch,20h          ;Set bit 5, cursor invisible
                mov     ah,1            ;Set cursor size and

                int     10h             ; blink characteristic
                ret

CURSOR_OFF      endp


;Turn cursor on...normal blink rate

CURSOR_ON       proc    near

                mov     ah,3            ;Get cursor scan line info
                int     10h             ; and blink characteristics
                and     ch,0dfh         ;Clear bit 5, cursor visible
                mov     ah,1            ;Set cursor size and
                int     10h             ; blink characteristic
                ret

CURSOR_ON       endp

;Display character string with attribute


PRINT_STRING    proc    near

                mov     bh,00           ;Page zero
                mov     cx,0001         ;Display character one time
GET_MORE:
                mov     ah,2            ;Set cursor position
                int     10h             ;Move it
                mov     ah,9            ;Write character and attribute
                lodsb                   ;Get character into al
                int     10h             ;Display it
                inc     dl              ;Advance cursor position
                or      al,al           ;End of message?
                jnz     GET_MORE        ;No, get more characters
                ret                     ;Yes, go back

PRINT_STRING    endp


;Erase character...cx has number of times to write, bl has attribute

ERASE           proc    near

                mov     bh,0            ;Page one
                mov     ah,9            ;Write character and attribute
                mov     al,20h          ;Character is space
                int     10h
                ret

ERASE           endp


;Data area

.data

;messages

PSE1_M  db      ' Press any key for bright background ','$'
NPSE1   equ     $-PSE1_M
PSE_M   db      ' Press any key to return to DOS ','$'
NPSE    equ     $-PSE_M
LGE_M   db      ' This is a blinking border ',0
NLGE    equ     $-LGE_M
RLGE_M  db      ' Foreground blink is disabled ',0
NRLGE   equ     $-RLGE_M
BRT_M   db      ' This is bright background ',0
NBRT    equ     $-BRT_M
RBRT_M  db      ' Bright background disabled ',0

;Border values

LGE_BDR BORDER  <BL_NOR,1,1,20,76>
R_BDR   BORDER  <NOR,3,3,4,50>
L_BDR   BORDER  <NOR,11,10,6,15>

;Pattern values

LGE_PAT PATTERN <0dbh,0dfh,0dch,0dbh,0dbh,0dbh,0dbh>
R_PAT   PATTERN <0dbh,0dfh,0dch,0dbh,0dbh,0dbh,0dbh>
L_PAT   PATTERN <0b3h,0c4h,0c4h,0dah,0c0h,0d9h,0bfh>

;Shadow values

R_SHAD  SHADOW  <RGT,HL_NOR,0b3h,0c4h,0bfh,0d9h,0c0h,5,50>
L_SHAD  SHADOW  <LFT,NOR,0dbh,0dfh,020h,0dfh,0dfh,7,15>


.stack

                end
```

QUIKDATA - 16 YEARS OF H/Z SUPPORT!

YOUR H/Z STORAGE & ENHANCEMENT EXPERTS!

ACCELERATE YOUR PC/XT/AT!

For your H/Z241, 248 and 386/16 we have the direct replacement **WESTERN IMAGING Z-33 ZUNI MOTHERBOARD** that's just loaded with features. Super fast 33Mhz 80386 main board with ten expansion slots **and** built in are two serial ports, one parallel port, floppy disk controller and an IDE hard disk controller! Add video (and your old MFM controller if you desire) and you still have plenty (9 slots) of expansion left! Support to 32 meg on-board RAM using SIMMs, and 32 additional megs with memory expansion.
WIZ-33 - $795 with 0K RAM **80387-33** - $195 Coprocessor
SIM1X9-8 $45 1 meg X 1 70ns SIMM DRAM
SIM256-8 - $13 256K X 1 70ns SIMM DRAM

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

MEMORY UPGRADES

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

Z150MP - $17 Will allow you to **upgrade your H/Z150/160 to up to 704K** on the main memory board, using up to 18 256K DRAM chips.

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

ZMF100 - $55 Will allow you to **upgrade your H/Z110/120** (old motherboards; with p/n less than 181-4918) **to 768K system RAM**. Requires 27 256K DRAM chips.
Z100MP - $55 Similar to ZMF100 above, but for new motherboards with p/n 181-4918 or greater.

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

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

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

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

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

HARD DRIVE & TAPE UPGRADE KITS

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

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

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

FLOPPY DRIVE SAMPLE

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

ANY TYPE FLOPPY IN YOUR PC/XT/AT

With the **CompatiCard**, you can install up to four additional drives in your PC/XT/AT computer. Add a 1.2 meg, a 1.44 meg, or any other drive, including 8" to any PC with an expansion slot. The CompatiCard (CCARD) will handle up to four drives, and the CompatiCard II (CCARD2) will handle up to 2 drives. CCARD4 has boot ROM to allow it to be used as primary boot controller in systems that allow you to remove floppy controller. CCARD4 also supports 2.8MB 3.5" floppy drives. Additional cables and external enclosures may be required.
CCARD2 - $79 **CCARD** - $99 **CCARD4** - $119

OR, add a floppy to any laptop or PC with a parallel port easily and inexpensively. With **Backpack**, you simply connect the external unit to your parallel printer port (do not lose printer function), install software, and away you go! No expansion boxes needed for laptops, and no slots required. Want a 2.8mb/1.4mb/720K floppy on your MinisPort? Want to add a 1.2 meg to your laptop? Want to add an additional drive to your desktop? Plug it in and go. 2.8MB 3.5" version will read and write 2.8 meg, 1.4 meg, and 720K format.
BPACK2.8 - $279 **BPACK1.4** - $229
BPACK1.2 - $229 **BPACK360** - $229

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

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

8-BIT/Z100

We carry a full line of replacement boards, parts and power supplies for the H/Z89/90 and Z100. We also have some H8 boards available. We continue to fully support and carry a full line of hardware and software products for the H8/H89/90 and Z100 computers. Of course we carry much more. Find out!

OTHER STUFF

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

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

QUIKDATA, INC.

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

UPGRADING A Z-248 TO A 386SX

NICK VISCO
5639 HEMING AVENUE
SPRINGFIELD, VA 22151-2707

Background

I originally purchased my 8 MHz Z-248 in April of 1987. The package included a 80286 processor, 512 KB of RAM, one 5-1/4" floppy (360 KB), one Seagate 20 MB (model 225) hard disk drive, one serial port, one parallel port, an EGA adapter (256 KB), an 84-key keyboard, ten 5-1/4" diskettes, MS-DOS 3.2, GW-BASIC, Windows 1.0, diagnostic software, an EGA color monitor (model 1380), and 10 slots (four already filled with a disk controller card, a CPU card, an I/O card, and a video card: leaving 6 open slots).

Minor Upgrades

Through the years I made minor hardware upgrades to accommodate my purchases of the latest software. I purchased a serial Logitech mouse and had to install a serial card since my original serial port was being used by the modem. I also purchased 128 KB of memory to fill the memory from 512 KB to 640 KB.

Having to deal with the new smaller floppies and my running out of hard disk space, I purchased a Teac high density (1.4 MB) 3-1/2" disk drive and a 40 MB Seagate (model 251-1) hard drive. I purchased updated Zenith Data Systems ROMs to access the 1.4 MB drive. Zenith Data Systems, thank their technical expertise, not only provided a disk controller card that allowed two floppy drives, and two hard drives; but also allowed the space to fit each internal bay of the machine with required power wires from the 200 watt power supply.

When my monitor died, I purchased a 14" VGA monitor with .28 mm dot pitch and an inexpensive VGA card. I am glad that the video port was on a card and not part of the motherboard.

With my purchase of Windows 3.0 and MS-DOS 5.0, my 640 KB of memory was insufficient. I decided to upgrade my memory beyond 640 KB and rid myself of this 128 KB memory card which had caused me problems with Prodigy, WordPerfect, and MS-DOS 5.0. I purchased an AST 286 memory card because it used standard SIMMs; but, I ran into memory parity errors after installing the card. I returned the AST card.

Options for Upgrade

My decision now was to look for another memory card, buy an upgrade kit/card to not only increase my memory but also my processor to a 386SX or greater, or to purchase a new computer.

Buying just a memory card would be the most inexpensive solution. My criteria for the card was that it use standard SIMMs so that in the future I might be able to use the memory in any new computer that I may have purchased. Zenith Data Systems memory cards do not use standard SIMMs for upgrades to 8 MHZ Z-248 computers. My first try at installing a standard memory failed. If a popular standard memory card like the AST '286 memory card did not work in my Z-248, the chances of any other memory card working in my Z-248 looked slim.

I decided now was the time to purchase a 386SX. I had two options; buy a new 386SX or buy an upgrade kit/card for my Z-248 computer. First, I looked through all the past excellent Remark articles on upgrading a Z-248 computer. I also attended a Computer Conference in Washington D.C. and picked up all the handouts and literature I could find on upgrading a Z-248 computer. After much research, I decided that a SXZ-248 card (a 386SX processor) from Technology Specialists (address and phone number are at the end of this article) in Virginia was my best bet. The ease of installation, use of standard SIMMs, price ($494 for a 386SX, 16 MHz with 1 MB of RAM), and nearby location, made this upgrade card the clear winner. With my first half of my research complete, I embarked on my search of a new 386SX computer. Then I could compare the two options.

I searched for the cheapest IBM compatible 386SX I could find. I found a 386SX with 1 MB of RAM, 85 MB hard drive (18 ms) and a 1.4 MB (3-1/2 inch) floppy drive for $685. The computer had the 386SX processor on the motherboard. The advantages vs the disadvantages of buying a new computer as compared to buying an upgrade card are summarized below.

Advantages

- Large hard drive (85 MB vs 60 MB)
- Faster hard drive
- Spare 3-1/2" floppy
- Newer shell with a new power supply
- Have a 286 8 MHz shell
- Have 20 MB/40 MB MFM drives

Disadvantages

- Cost $685 vs $494 ($191 difference)
- New computer not upgradeable to a new processor
- Need to transfer all data to new hard drive
- Need to transfer modem, mouse, monitor, video card, and 5-1/4" drive to new computer

Looking at the above, the important advantages were a new, faster, larger hard drive and extra 3-1/2" floppy drive. The important disadvantages were the $191 cost and the computer was not upgradeable to a new processor.

I considered the advantage of having a spare 3-1/2" floppy drive vs the disadvan-

tage of a $191 difference to equal out. So it came down to basically getting a new 85 MB hard drive, as compared to, not being able to upgrade my processor, and the time plus troubleshooting of transferring data and hardware to a new computer.

Decision

Although not an easy decision, I decided that I have built up a lot of confidence in my Zenith Data Systems '248 computer, and would update my hard drives at a later date. I think upgrading in steps is more costly, but when a hardware part breaks, you get the latest technology for that particular hardware at that time. The other options are to fix the old part, or buy a new system whenever a major component is outdated or fails.

So I went out and bought my 386SX upgrade card. The 386SX card is priced for different configurations. There are 16 MHz, 20 MHz, and 25 MHz versions (increasing cost, of course). The standard configuration is 1 MB (80ns) installed, but you can order cards with up to 32 MB of RAM. I decided to order the 20 MHz version with 1 MB of RAM. The 20 MHz version was $70 greater than the 16 MHz version. I decided to get just the minimum 1 MB of RAM, since extra RAM versions cost $75 for each increment of 1 MB. I purchased 3 MB of RAM at $42 for each 1 MB from a mail order firm and installed this RAM myself.

Installation

Installation was a snap. The card comes with an instruction booklet which I followed step-by-step. All you need is a screwdriver for the actual hardware installation. The 386SX card replaces both the Zenith Data Systems CPU and I/O cards. The 386SX contains the 386SX processor, ROM BIOS, CMOS (setup and date/time information), 1 serial, 1 parallel, and a ribbon cable to a second serial port that does not use a slot but just the opening of the slot at the back of the computer, and 8 standard SIMM memory slots where you can install 256 KB, 1 MB, or 4 MB SIMMs. SIMMs can be mixed and matched on the card in groups of 2 slots.

The first step in the instruction booklet was to access the setup on your existing Z-248 to copy down the setup information (hard drive types, floppy types, video display, etc). I have always kept a hard copy of this information in case of battery failure. Since this one card replaces both the CPU card and the I/O card (where setup information is kept on CMOS), you must know this setup information in a later step of the installation process.

The next step is to remove the speaker jack, CPU, I/O card and any Zenith Data memory cards. Then install the 386SX card and attach the speaker jack.

The 386SX ROM BIOS supports all 46 drive types which are standard on the Z-248. Instead of CTRL-ALT-INS to get to the setup screen, the hot key is CTRL-ALT-S to get to setup on the 386SX. Since no information is in the setup (CMOS), you will get a POST (power on self test) error. You will be prompted to press F2 to get to setup. Insert the time, date, RAM, hard drive, floppy drive, video display, etc., information on the setup screen. Save and exit the setup screen and you are ready to go.

Some ROM BIOS utilities that are available at the setup screen are a screen saver, password protection, quick boot (bypass floppy drive test and boot only from the hard disk), and soft key lock to temporarily lock the keyboard.

One feature on my Z-248 ROM BIOS which I used occasionally was the option to boot from hard drive 0, hard drive 1, floppy 0, or floppy 1. I could put different versions of autoexec.bat and config.sys on my two hard drives. This feature is not available on the ROM BIOS of the 386SX card. Beside this one minor drawback, I have been extremely pleased with my new(?) computer.

I have converted my original 8 MHZ Z-248 with 512 KB of memory into a 16 MHZ 80386SX with 4 MB of memory. Since I have upgraded everything else on my computer, I have a feeling that my next upgrade will be to my hard drives; but for now they are operating just fine.

Product Mentioned

SXZ-248 card (386SX processor)
Technology Specialists, Inc.
3110 Columbia Pike, Suite 303
Arlington, VA 22204
(703) 521-1886 or (800) 4-UPGRADE

Continued from Page 30

record can seem rather trivial next to 20 ms, but when thousands of records are being accessed during a sort process, much more time is spent moving from track to track than actually reading data. The slow drive can easily turn a 20 second job into 65 seconds and in a day and age where computers are expected to perform tasks instantaneously, waiting for the computer to do its job is almost unacceptable.

The primary factor in determining the drive's access time is the type of head actuator mechanism used. Drives with access times less than 25 mS all have the more expensive voice coil actuators. Slower drives tend to have stepper motor actuators.

Transfer Rate

Another benchmark of drive performance is its true transfer rate, measured in Mbits/second. This number is the theoretical upper limit of how much data can be transferred from the drive platters to the computer's CPU. This rate is based essentially on the speed of the drive platter rotation (usually around 3600 RPM) and the density of data on the data track. In other words, the drive's ability to transfer data is theoretically limited to the amount of data that passes under its read/write head during a period of time. As more data is crammed onto the track, and as the platter rotates faster, then the rate will increase. In a practical sense, however, this upper rate cannot be achieved. The factors limiting performance tend to be the drive controller electronics and the computer's CPU speed. As a rule, they cannot process data quickly enough and will not perform anywhere near the theoretical rate.

There are several methods used by drive vendors that vastly improve the drive's ability to pass data to/from the CPU. One technique is to optimize the drive's interleave ratio. The concept of an interleave ratio is really very simple, the classic way of describing it is comparing it to the task of gathering your luggage off of one of the baggage conveyor belts seen in airports. Suppose you have three bags on the conveyor belt. If all three bags are situated right next to one another, you may be able to grab the first one and put it aside, but in the period to do so, the other two bags passed you by. As a consequence, you have to wait until the next pass to grab the second bag, and possibly the third pass for the last bag. However, if the bags were placed farther apart, or if the conveyor was moving slower, then you could retrieve all three bags on the first pass. This is a simplified explanation of how the drive's head reads data from a rotating platter full of data. On slower drives with a 4:1 interleave ratio, data is read from one sector, the next three sectors are skipped, and then reading continues with the fourth sector. On faster drives, virtually all with 1:1 interleave ratios, data is read from consecutive sectors.

Another technique used to improve the drive's transfer rate is to incorporate a sector caching memory buffer on the drive controller. In some of the newer IDE drives, a 32K or 64K byte cache buffer is used which greatly improves average access times and overall data transfer rates.

Coming Next

The next article in this series will cover a variety of different drive topics, but if you don't know a IDE drive from an ESDI drive, then certainly stay tuned for Drive Lesson #2!

QuikMenu III

Part 1

Craig S. Stevenson
Stevenson Technical Services, Inc.
108 Second Avenue West, Box 297
Bertha, MN 56437-0297

My first menuing system was a shareware program called AutoMenu. It met my needs adequately for several years. As my requirements became more complex, I began to search for a replacement menu system. I looked at every shareware menu program I could find, and always came back to AutoMenu. It seemed as if all of the programs had some limitation or bug that made them unusable for my applications. Then I tried QuikMenu. It was different than most of the other menu programs because it had a graphical interface. In fact, it looked a lot like Windows. I put QuikMenu through all the tests I could think of, and decided that it was the most flexible and attractive menu program I had seen. I sent in my registration, and have been very pleased with the program and support since that time.

QuikMenu started out as shareware. The current shareware version is 1.07. QuikMenu III is a commercial product which uses Windows icons to further enhance the appearance of the menus. QuikMenu III comes complete with several ready-made icons, and includes an icon editor which can create new icons, or import Windows icons. At the end of this article, I will include purchasing information for both the commercial and shareware versions of QuikMenu.

Why Use A Menu?

Many people do not like IBM compatible machines because they do not appear user-friendly. Booting a computer and seeing only a DOS prompt can be a little intimidating. This is one of the main reasons for the proliferation of DOS shells and menu systems in the marketplace. Menu programs provide an insulator between the user and DOS. They assist in organizing all of the programs that reside on the hard drive by making them easily accessible. A properly designed menu system can be user-friendly, functional, and provide additional security for sensitive information.

Why QuikMenu?

I have looked at many menu packages, and have chosen QuikMenu because of its ability to perform all of the tasks that I require and because of its reasonable price. QuikMenu also offers enhancements to the DOS and network environments that I use. QuikMenu III features:

Graphical Icon Interface
Automatic Installation
Works With All Popular Software
Works With Mouse Or Keyboard
On-Line Context-Sensitive Help
File Manager
Icon Editor
Time Tracking
Calculator
Calendar/Daily Reminder
Phone Book Database
Telephone Dialer (Requires Modem)
Electronic Mail (Requires Novell Or LANtastic Network)
Extensive Password Protection
Multiple Menus (Up To 50 Pages Per Menu)
Hypertext-Style Page Links
Command Line Variables/File Selection
Menu Screen Saver
Selectable Screen Colors
PCX Menu Background (Requires EMS Memory)
Highly Customizable
Free-Form Layout
Not Memory Resident (No TSR)
Network Compatible
Supports CGA, EGA, VGA, And Hercules
Reasonably Priced
Site Licenses Available

What Can I Do With QuikMenu?

My company utilizes several computer systems using a wide variety of programs and environments. Programs are continually added to my stand-alone and network server hard drives, which means that I continually need more menu space to contain my program options. QuikMenu allows as many as 68 items on each page with up to 50 pages per menu. I am currently using a 10-Page menu on my main computer. I have an average of 15 to 20 software icons and a standard set of page icons on each menu page. I also stretch some icons the full height or width of the page and use them as divider bars. This organizes my menu by separating the page icons from the software icons. Figure 1 shows an actual page from my menu.

While most of the icons execute DOS applications, others perform repetitive tasks such as copying a file to or from the hard drive. I also use the built-in file editor to edit batch files when necessary. The calendar is used as a reminder on days when I have an important business meeting, or if it's my anniversary. (My wife usually puts that one in for me.) My wife also password-protects some of her accounting software to prevent me from accessing it and doing any

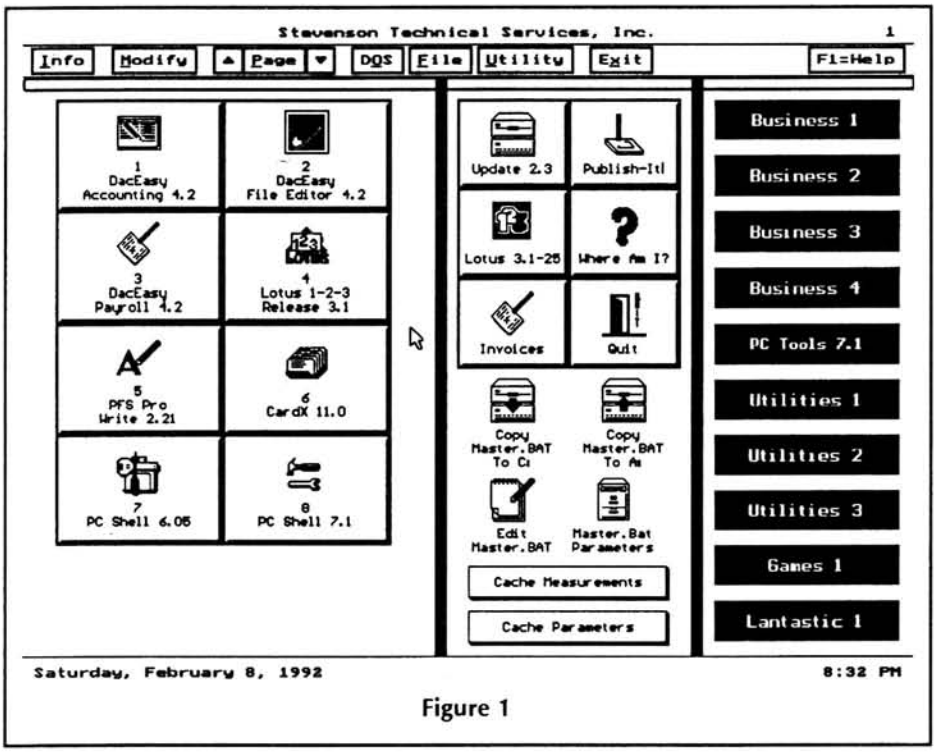


Figure 1

damage, (although she still asks me to fix things if anything major goes wrong). A keyboard stuffer comes in handy to perform repetitive keystrokes after entering a DOS application. QuikMenu III covers all these needs and more. I have recommended that my clients use QuikMenu, and they have been impressed by its many features. One of my customers uses it simply because it is "pretty". Others use it because it is easier to use than their previous menu program. Another customer, (a school district) utilizes the messaging services for their inter-office communications. QuikMenu may be customized to fit a variety of needs.

This Article

This article is divided into several parts. Part 1 will cover basic techniques used in setting up a simple menu. Enough information will be included to get you started with QuikMenu. Future parts will cover enhancing icons, the File Manager, QuikMenu's built-in utilities, command line options, password protection, and networks.

Although QuikMenu supports CGA, EGA, VGA, and Hercules graphics cards and monitors, the greatest flexibility can be obtained by using a VGA color system. QuikMenu makes extensive use of color, and a properly constructed menu can have a striking visual appeal. In order to make the screen captures more readable, my menu is temporarily set up in VGA monochrome mode. In the following section, I will show you how to set up, modify, and customize a basic menu. You will discover how to set the QuikMenu defaults, as well as add, edit, and copy icons. QuikMenu is one of very few programs I have found that makes productive use of a mouse. If you do not have a mouse, it will be difficult, but not impossible, to adapt.

Conventions

For the duration of this article, items that are to be pointed at and clicked on with the mouse will be enclosed in braces {}. Keyboard shortcuts will be enclosed in brackets []. QuikMenu field names will be enclosed in parentheses (), and text that you are required to enter will be enclosed in double quotes "". Do not type the double quotes, only what is enclosed in them. You will also be adding some familiar software to the menu. If you do not have the specified software, you may substitute other programs on your hard drive.

Starting QuikMenu III

To begin using QuikMenu, change to the \QUIKMENU subdirectory. You will enter different commands to run QuikMenu depending on how you installed it. If you installed QuikMenu for use on a non-network machine, simply type "QM" and press [ENTER]. If you installed QuikMenu on a network, you should type "QNET USERNAME". QNET is the batch file invoked to run QuikMenu, and USERNAME is typically your network login name. QuikMenu uses your username to create a file that contains your menu preferences. If you are a network user, type "QNET USERNAME" and press [ENTER].

Within a couple of seconds, the opening QuikMenu screen should appear. If you just installed QuikMenu, a dialog box will appear asking, "Do you want me to create a menu for you?" If you answer yes, QuikMenu will search your entire hard drive for programs that it recognizes. If it finds any, it will install an icon for each program that is found. For the purpose of this article, answer no at the prompt by selecting {ESC=No} [ESC]. I will teach you how to set up and modify these icons on

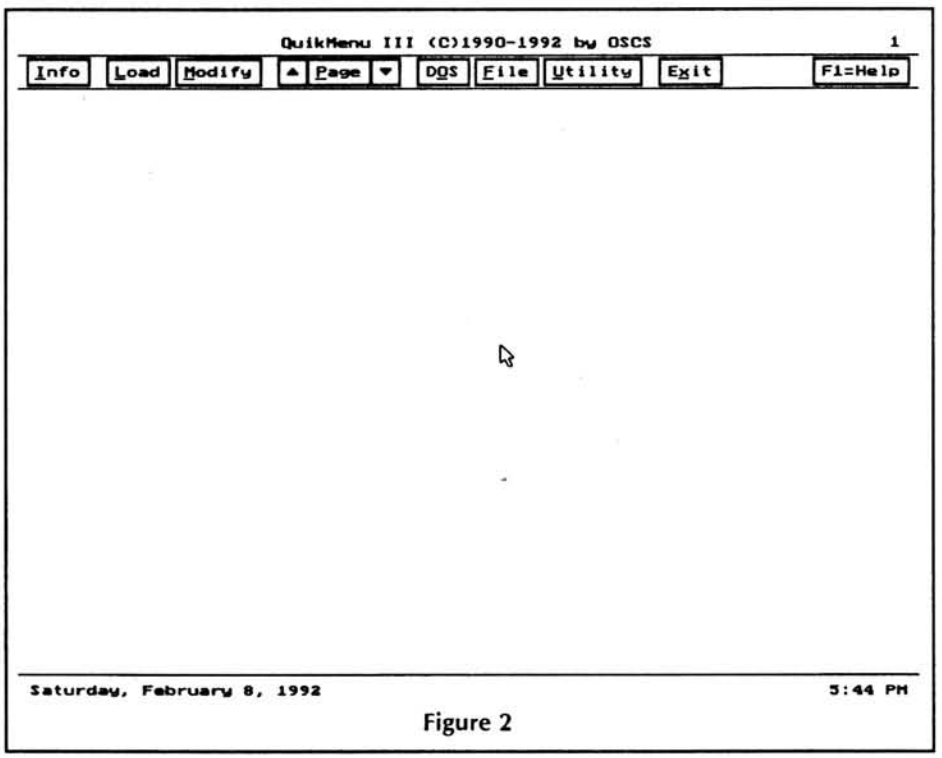


Figure 2

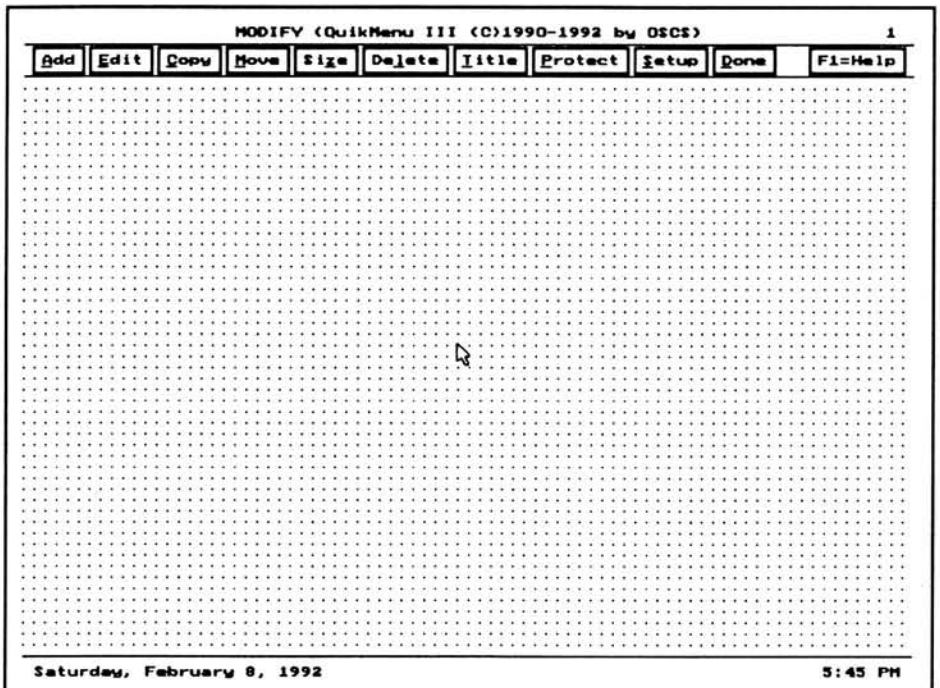


Figure 3

your own.

After answering no, you should see a blank menu page that looks like Figure 2.

The top line of the screen is the title bar. The title bar may contain the name of your business, the name of each menu page, or both. It is entirely up to you. The title bar also shows you which menu page you are on by the number displayed in the top right corner. This now displays 1, indicating that you are on the first page. The second line is called the menu bar, and contains several command buttons. When you are modifying the menu, the menu bar will display prompts when necessary. The largest portion of the screen is the menu background where the icons will be located. The bottom line of the screen is the status bar, containing the date and time indicators. At other times, the status bar will change to provide you with information or instructions when a prompt appears.

The Command Buttons

The command buttons, located on the menu bar, allow you to create and customize the menu. The buttons listed from left to right are: Info, Modify, Page, DOS, File, Utility, Exit, and F1=Help.

The Info button displays information about QuikMenu III, including the version number, copyright notice, technical support information, and the identity of the licensee.

The Modify button is the most complex of the command buttons, since it allows you to add, edit, copy, move, size, and delete icons. The more advanced options allow you to change the graphic, color, icon style, font, font size, and alignment of an icon. You will use Modify later in this article to access the QuikMenu III defaults and create software icons.

The Page button allows you to move back and forth between different pages of the menu. It includes an up arrow button on the left and a down arrow button on the right to move up one page or down one page, respectively.

The DOS button allows you to temporarily exit to a DOS prompt. You can operate from DOS as long as you wish, and return to QuikMenu by typing "EXIT" and pressing [ENTER].

The File button gives you access to the built-in file management system. This system allows you to copy, move, rename, delete, find, view, and obtain information on files. You may also manipulate directories by using the file management system. I will be covering the file management system in greater detail in a future part of this article.

The Utility button directs you to several other built-in features of QuikMenu, including the network mail services, calculator, calendar, phone book, and time log.

The Exit button removes QuikMenu from the computer memory completely. To return to QuikMenu, you must re-enter the commands you used to execute QuikMenu (see Starting QuikMenu III).

Help is available from the main menu screen, or while using the built-in features of QuikMenu. You may select {F1=Help} [F1] from the main menu to display general help for using QuikMenu. When you are using one of QuikMenu's built- in features or utilities, you may press [F1] to display context sensitive help.

Although you cannot see it now, there is another command button available on the menu bar. This button is called Load. Load allows you to retrieve an alternate menu temporarily for the purpose of making changes to it. QuikMenu's default setup has the Load button hidden to prevent access by unauthorized users. All of these command buttons can be hidden, and most of them can be password- protected. Hidden buttons cannot be accessed with the mouse, but can be accessed using keyboard commands.

What are Icons?

Icons are pictures and/or text used to identify programs or options. There are four type of icons, and each performs a separate function.

Software Icons - Software icons are usually used to execute DOS commands. Anything that can be done with DOS can be done with a software icon.

Page Icons - Page icons are used to move back and forth between menu pages. Page icons are similar to hypertext links, linking a specific menu page to its own icon. This allows rapid movement between menu pages, without having to move through intermediate pages.

Macro Icons - Macro icons are used to automate repetitive keystrokes or mouse actions within QuikMenu.

Dialer Icons - Dialer icons use a modem to dial telephone numbers. There must be a modem connected to your computer system in order to use a dialer icon.

Defaults

The first step to setting up a menu system is to set the program's defaults. To access QuikMenu's defaults, select {Modify} [ALT-M]. The screen will be redrawn and display a grid on the menu surface. You should notice an entirely new set of command buttons on the menu bar, and the screen should look like Figure 3.

The first six command buttons on the menu bar are used to manipulate icons. They are: Add, Edit, Copy, Move, Size, and Delete. The next command button, Title, allows you to change the information displayed on the title bar. Protect allows you to password-protect one or more menu pages. This can prevent users from accessing potentially destructive utilities and programs, or it can make sensitive business information more secure. Setup is where the defaults for QuikMenu are located, and Done is selected when you have completed any modifications. Select F1=Help in case you forget the purpose of any of these command buttons.

Select {Setup} [ALT-S] to modify the QuikMenu defaults. Your screen should look like Figure 4.

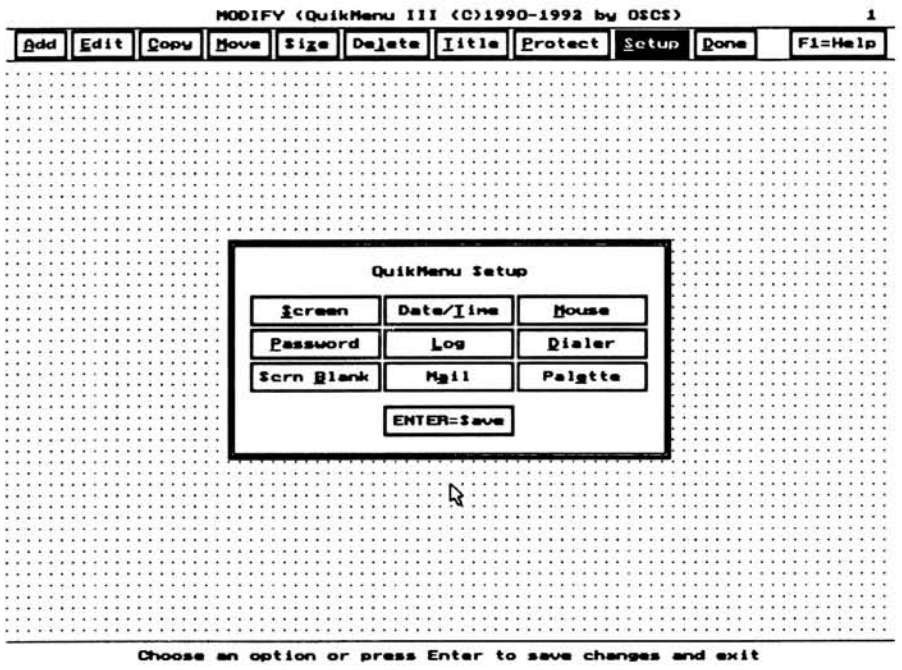


Figure 4

There are ten command buttons to choose from on the QuikMenu Setup screen. Nine of these buttons allow access to different setup areas of QuikMenu. The tenth button, {ENTER=Save}, saves your changes and returns you to the Modify menu. I will walk you through using each of them. Remember that the status bar will usually display the keys needed to select and de-select items.

Screen Defaults

Click on {Screen} [ALT-S]. Your screen should look like Figure 5.

This dialog box shows the default setup options for the screen. On the left side of the dialog box, you will find a set of 16 colors that make up QuikMenu's color scheme. Below the color choices, there is an option to use a PCX file as the menu background. Use of this option requires EMS memory.

On the right side of the dialog box, you are presented with six options. To select or de-select an option, click on the box next to it, or hold down the [ALT] key and press the option's underlined letter. These are as follows:

Sound Effects [ALT-S] — When this option is selected, your PC speaker will produce sound effects each time a command button or icon is selected. Although this is cute for awhile, it can sometimes interfere with your concentration. The default for sound effects is off.

Confirm Applications [ALT-C] — When this option is selected, QuikMenu asks for confirmation each time you wish to activate a software, macro, or dialer icon. This can be handy when you are first starting out, but it is not necessary since it adds an extra mouse click or keystroke every time you use it. The default for confirm applications is off.

3-D Interface [ALT-I] — If your computer's video display is capable of handling the 3-dimensional interface that QuikMenu offers, this option will be selected as the default. It is a good idea to have this option turned on since it adds a very nice look to the menu.

Display Speed Keys [ALT-D] — If this option is selected, QuikMenu will display the speed key, or keyboard shortcut, used to activate an icon. The default for display speed keys is on.

Display Grid [ALT-G] — When this option is selected, QuikMenu turns the grid on when you are in modify mode. This grid is helpful for aligning icons. The default for display grid is on.

Snap to Grid [ALT-N] — Snap to Grid means that the four sides of an icon snap themselves to the nearest grid line. This feature makes aligning icons easy and precise. Turn this on by selecting {Snap to Grid} [ALT-N].

Select {ENTER=Ok} [ENTER] to save your changes, and you will be brought back to the QuikMenu Setup screen.

Other Defaults

Click on {Date/Time} [ALT-T]. This dialog box allows you to update the current date and time kept on your computer's real-time clock. Check the date and time indicators to make sure they are correct. If they are incorrect, set them now, using the [TAB] key to move between fields. Press [BACKSPACE] to delete an existing entry and then type in the new entry. When you are finished, select {ENTER=Ok} [ENTER] to save your changes.

Click on {Mouse} [ALT-M]. This box allows you to adjust the speed of your mouse. If your mouse seems to be moving too fast, select {Slow} [ALT-S]. If your mouse seems to be moving too slow, select {Fast}

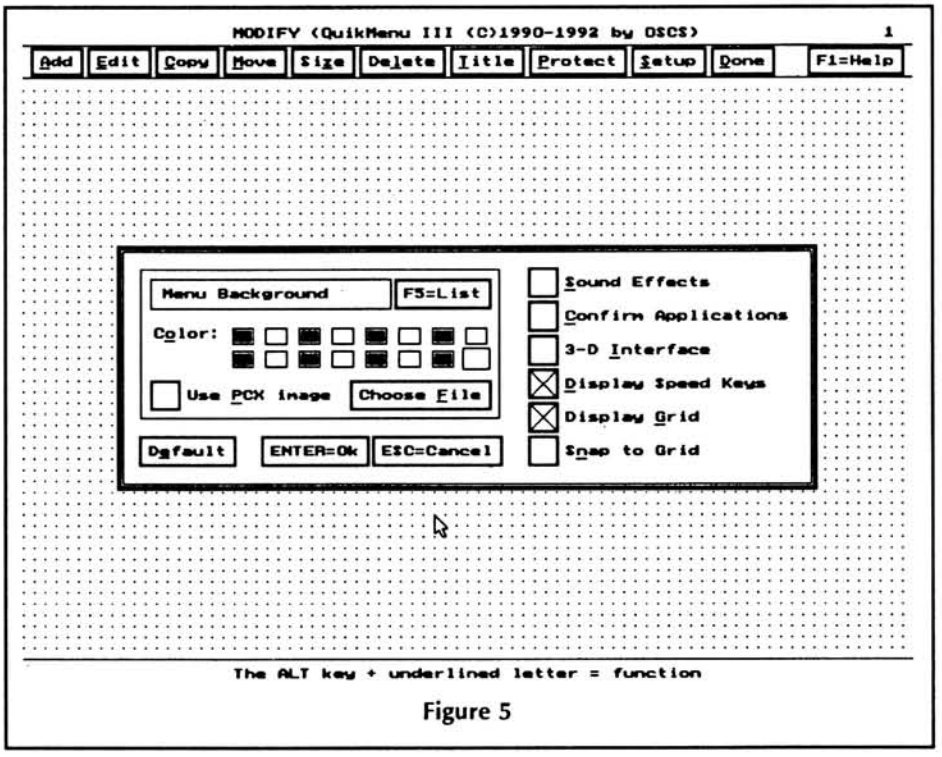


Figure 5

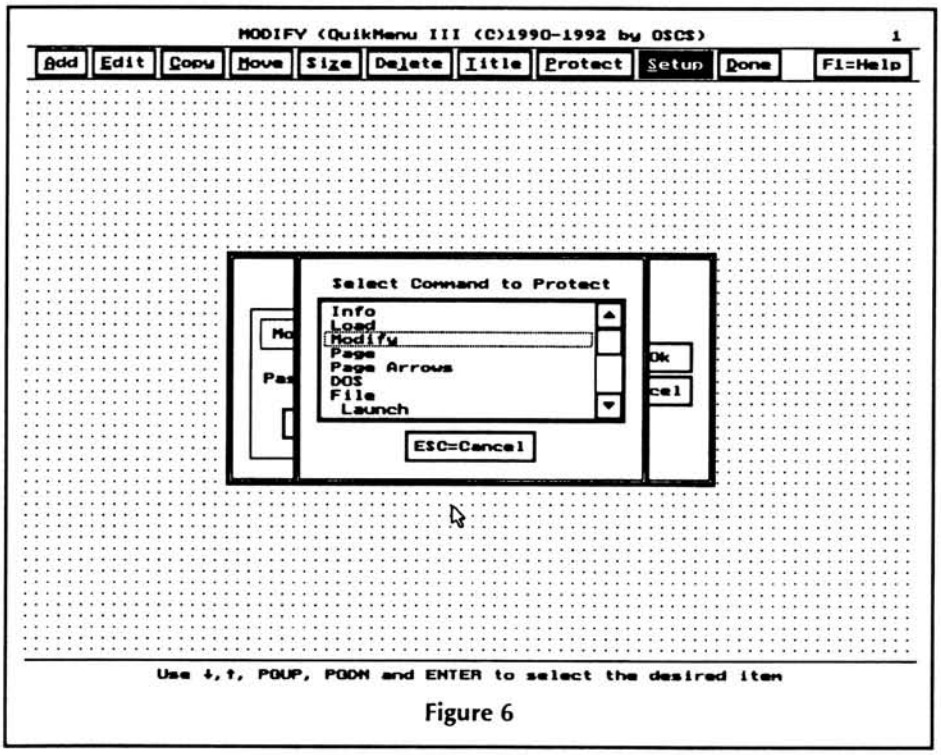


Figure 6

[ALT-F]. The default speed is medium. These changes will only apply to mouse movements within QuikMenu. Other programs will not be affected. Select {ENTER=Ok} [ENTER] to save your changes.

Click on {Password} [ALT-P]. Within this dialog box are the options to hide and/or password-protect the command buttons located on the main menu bar. Select {F5=List} [F5] to list the buttons you can hide or password-protect. Your screen should look like Figure 6.

The box at the front of this screen is called a list box. You will see list boxes several times within this series of QuikMenu articles. List boxes are normally used to select items with ease. They typically contain a title at the top, a scroll bar on the right side, and an outlined box in the interior called the item selector. If you are using a keyboard, move the item selector using the [UP ARROW], [DOWN ARROW], [PGUP], [PGDN], [HOME], and [END] keys. If you are using a mouse, move the item selector using the scroll bar. Click on the up and down arrows at the top and bottom of the scroll bar. These arrows will move the item selector up and down within the list box. You may also click and hold down the mouse button on the scroll button located within the scroll bar. Drag it slowly, and you will see the other available options presented within the list box. Go back to the top of the list, and release the mouse button when the option {Load} can be seen in the window. Click on it with the mouse to select it. If you are using the keyboard, use the arrow keys to highlight it with the item selector, and press [ENTER] to select it. QuikMenu will return you to the previous screen. Although this item does not have a password, the option (Hide This Button) is selected. De-select this option by clicking in the box next to the prompt, or press [ALT-H]. To password-protect this item, type in the word "QUIKMENU" at the (Password:) prompt. This button will no longer be hidden, but will require the password "QUIKMENU" before it may be used. Select {ENTER=Ok} [ENTER] to save your changes.

Click on {Log} [ALT-L]. This feature allows you to track activity for yourself or others. You can track applications entered, times they were entered, and the length of time accessed. You may also change the name of the log file from this screen. Although time logging is a useful feature, I have found that it tends to make QuikMenu perform a little slower. The difference is barely noticeable, but I prefer the increased performance. De-select the option (Keep a Time Log of All Menu Activity) [ALT-K], and select {ENTER=Ok} [ENTER] to save your changes.

Click on {Dialer} [ALT-D]. This dialog box allows you to select the dialing mode of your modem and the COM port it is attached to. If you have a modem attached to your computer, you may configure these options now. Select {ENTER=Ok} [ENTER] to save your changes.

Click on {Scrn Blank} [ALT-B]. QuikMenu's screen blanker is a built-in utility that prevents menu images from burning into the computer screen. The default is Digital Clock. The default blanking interval is 3 minutes. This means that after three minutes of inactivity within QuikMenu, the screen blanker you have selected will be enabled. QuikMenu will only blank its own screens, but not those of other programs. When you are finished, select {ENTER=Ok} [ENTER] to save your changes.

Click on {Mail} [ALT-A]. The mail feature is only available on network computers. If you are not using a network version

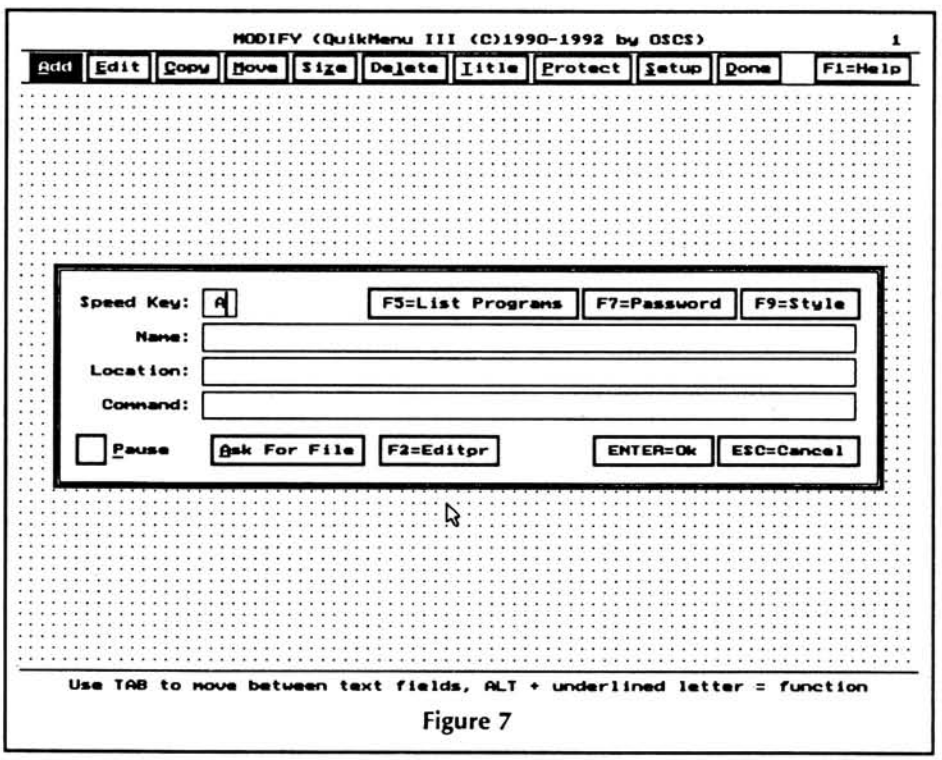


Figure 7

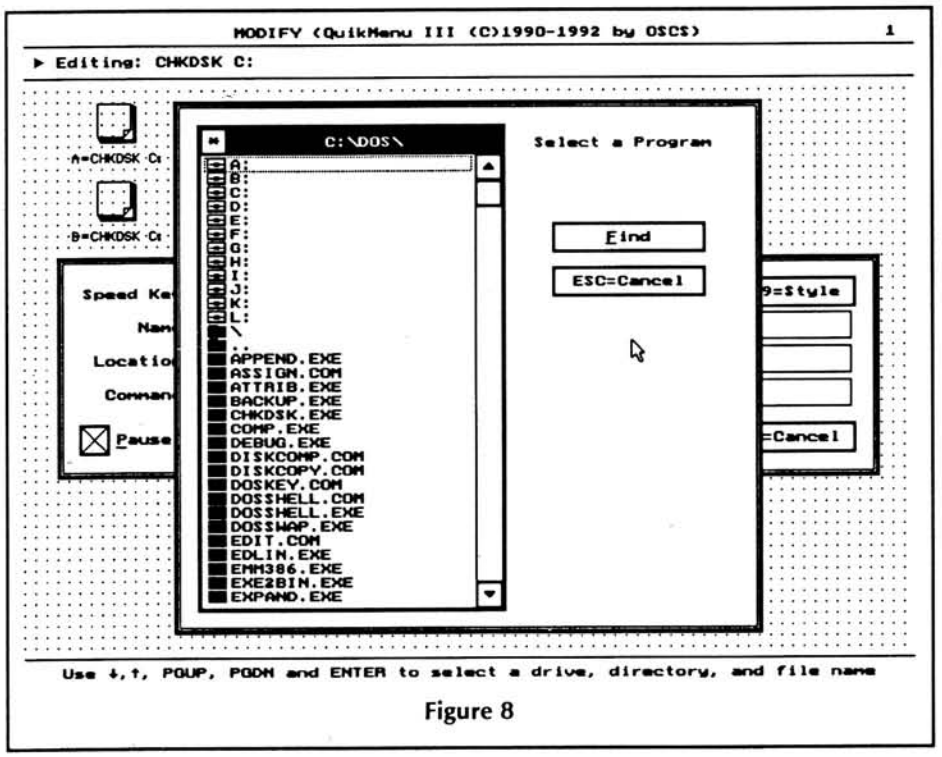


Figure 8

of QuikMenu and attempt to access this feature, QuikMenu will display the error message, "You must use the network version to access this feature!" If you are on a network computer, you may use this feature to have QuikMenu check your LANtastic or Novell mailbox continuously and notify you when you have mail waiting. You may also select the default sub-directory where mail should be kept. Select {ENTER=Ok} [ENTER] to save your changes or clear the error message.

Click on {Palette} [ALT-E]. If you are dissatisfied with the default colors of QuikMenu, you may modify the VGA palette to produce a different shade or color. (Note: You must have a VGA system to use this feature.) If you change the palette, and wish to restore the defaults, click on {Default} [ALT-D]. Select {ENTER=Ok} [ENTER] to save your changes.

Select {ENTER=Save} [ENTER] to exit QuikMenu Setup, and {Done} [ALT-D] to exit modify mode and save your changes.

Adding A Software Icon

To add a software icon to the menu, click on {Modify} [ALT-M]. Click on the command button labeled {Add} [ALT-A]. QuikMenu will prompt you if you want to add a Software Icon, Page Icon, Macro Icon, or Dialer Icon. Since you want to add a software icon, and it is already selected, click on {ENTER=Ok} [ENTER]. Your screen should look like Figure 7.

Figure 7 shows the editing dialog box for software icons. The first field is labeled (Speed Key:). Notice that this field is already filled in with the letter "A". QuikMenu supplies the next available letter or keyboard character that is not in use on this menu page. Press [TAB] to move to the (Name:) field, and type "CHKDSK". Press [TAB] to move to the (Location:) field. Enter the drive and path of the executable program. For DOS users, CHKDSK is normally located on the C: drive in the DOS subdirectory, so type "C:\DOS". (Some Zenith Data Systems DOS users may find CHKDSK in the C:\BIN subdirectory. DR DOS users should find CHKDSK in the C:\DRDOS subdirectory.) Press [TAB] again, to move to the (Command:) field. Type "CHKDSK", and click on {ENTER=Ok} [ENTER] to save the changes. (If you accidentally press [ENTER] before you finish, QuikMenu will save your changes, and ask you, "Where do you want to place this item?" Press [ESC] to clear this message, and select {Add} [ALT-A] to start over.)

When you are done entering the information for the new icon, the mouse pointer will take the shape of a hand and be positioned in the middle of the screen holding the icon. (On some screens, only an outline box will be visible.) QuikMenu will prompt, "Where do you want to place this item?" Move the icon to the upper-left corner of the menu background using the mouse or the arrow keys. Leave a margin between the edge of the menu background and the icon. Click the mouse button or press [ENTER], and the icon will remain in that position. Select {Done} [ALT-D], and you will be brought back to the main menu. Click on the icon labeled {CHKDSK}, or press [A], since that is its speed key, and the corresponding program will be executed. Notice that the program executed correctly, but did not pause for your review. In order to correct this problem, the icon must be edited.

Editing A Software Icon

Click on {Modify} [ALT-M], and then select {Edit} [ALT-E]. When you are prompted, "Which item do you want to Edit:", click on {CHKDSK}, or press [A]. The editing dialog box you used to add this icon now reappears. Look at the lower left corner of the editing dialog box. You will notice a small box with the word (Pause) beside it. Click inside the box [ALT-P] to select it. Select {ENTER=Ok} [ENTER], and your changes will be saved. Select {Done} [ALT-D] to return you to the main menu. Once again, click on {CHKDSK}, or press [A]. This time, the program not only executed, but paused for your review. It now waits for a keypress indicating you are finished with this program. Press the [SPACE BAR], and QuikMenu will return to the menu system and wait for the next command.

Copying An Icon

Now that you have an original icon, you can create others by copying it. Click on {Modify} [ALT-M]. Instead of adding another software icon using {Add}, select {Copy} [ALT- C]. QuikMenu will prompt, "Which item do you want to Copy:". Click on {CHKDSK}, or press [A]. When you do, QuikMenu displays a icon outline directly over the original, and prompts you, "Where do you want to place this item?" Move the mouse or use the arrow keys to move this icon down, leaving a margin between the two icons. Click the mouse button, or press [ENTER], to place the icon in this position. When you do, QuikMenu will display an error message stating, "An item with that SPEED KEY already exists on this page!" Select {ENTER=Ok} [ENTER] when you have read the error message. QuikMenu will then display a message saying, "Assigning new speed key..." QuikMenu will assign the next available character (in this case, B), as the new speed key for this icon. The copied icon contains the same attributes as the original icon, except for the speed key. Select {Edit} [ALT-E] to modify the new icon. When QuikMenu prompts, "Which item do you want to Edit:", click on {CHKDSK}, or press [B]. Use this icon to run PC Tools Shell 7.1. In the (Name:) field, you will need to [BACKSPACE] over the old name and type "PC Tools Shell 7.1". Do not retype the location and executable file name; QuikMenu offers an easier way. Click on {F5=List Programs} [F5]. Your screen should look like Figure 8.

If you do not remember the program's location or executable file name, you can use QuikMenu's list box to look it up. This list box looks slightly different than the one

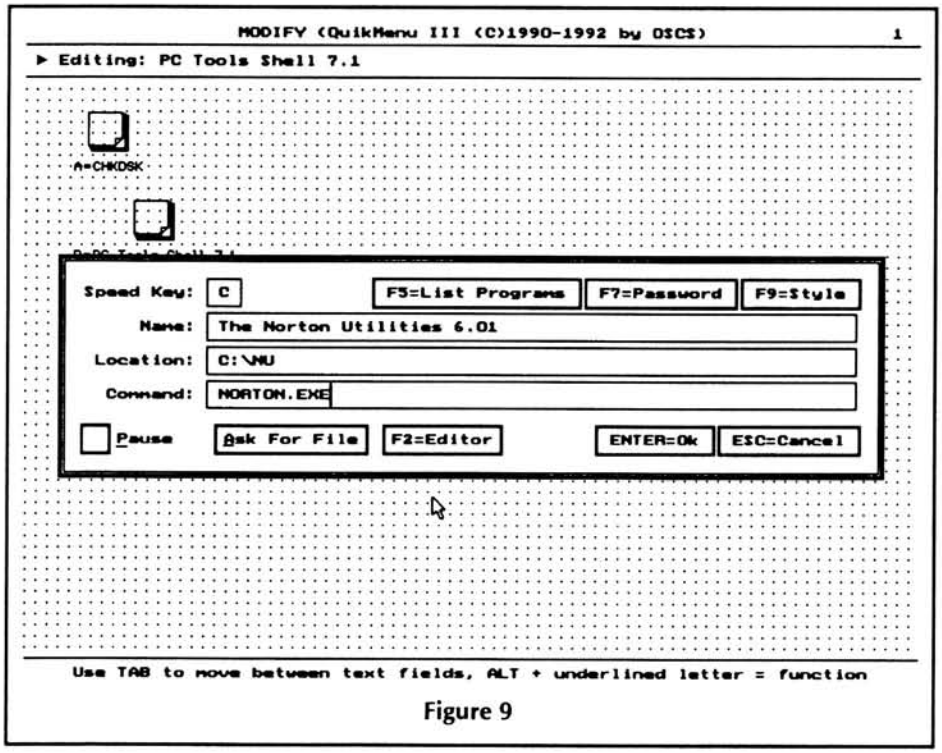


Figure 9

you used earlier. It contains a listing of drives, directories, and files on your hard drive.

To use QuikMenu's list box, you must be familiar with how it works. The list box title bar contains the current DOS path and is called the path box. Move the item selector up and down within the list box by clicking on the up and down arrows located on the scroll bar, or click and drag the scroll button within the scroll bar. If you are using a keyboard, you can use the up and down arrow keys to move the item selector. To select an item, click on it with the mouse or press [ENTER] when it is highlighted with the item selector.

If you need to back up to the parent directory, select {..}. To move to the root directory, select {\}. Drives are listed first, directories second, and files last. Each section is also listed in alphabetical order, making lookups easy. If you prefer a different sorting style, click on the asterisk located on the left side of the path box, or press [*]. This option will list files first, directories second, and drives last. Select the asterisk again [*], and the three sections will return to their previous order.

Select {\} to move to the root directory. Find the directory containing the PC Tools 7.1 files (usually PCTOOLS), and select it. Use the scroll bar or the arrow keys to scroll the list box until you see the file PCSHELL.EXE. Click on {PCSHELL.EXE}, or select it with the item selector, and press [ENTER].

If you cannot find this file, select {Find} [ALT-F] and type in the file name "PCSHELL.EXE". Select {Search Entire Drive} [ALT-S], and press {ENTER=Ok} [ENTER]. QuikMenu will search the entire partition to find this file. Select it, and QuikMenu will return to the editing dialog box. Notice that it entered the chosen location and file name into the correct positions for you. De-select the {Pause} button [ALT-P], and click on {ENTER=Ok} [ENTER] to save your changes. Select {Done} [ALT-D] to return to the main menu. Click on the icon labeled {PC Tools Shell 7.1}, or press [B], to run it.

It's Your Turn

It's time for you to try some of the things you've learned. Create a software icon for The Norton Utilities 6.01, and use NORTON.EXE for the executable file name. (This version of Norton Utilities is typically stored in the C:\NU subdirectory.) You should begin by clicking on {Modify} [ALT-M]. If you have any problems, go back and review the sections on adding, editing, and copying software icons. When editing the icon, your screen should look like Figure 9.

If you create an icon by mistake, remove it by clicking on {Delete} [ALT-L]. When QuikMenu prompts, "Which item do you want to Delete:", select the icon by clicking on it, or pressing its speed key. When QuikMenu asks you to confirm the deletion, select {ENTER=Ok} [ENTER]. After creating an icon for The Norton Utilities 6.01 and selecting {Done} [ALT-D] from Modify mode, your screen should look similar to Figure 10.

In part 2, I will cover adding page, macro, and dialer icons, using advanced options to enhance their looks, and page titles.

QuikMenu III is a registered trademark of OSCS Software Development, Inc. QuikMenu 1.07 (shareware) is available on CompuServe, or other major BBS's. *(Ver. 1.06 is available on COM1 in the UTILS section of the database. -Ed)* QuikMenu III may be purchased through OSCS Software Development, Inc., or any of its dealers.

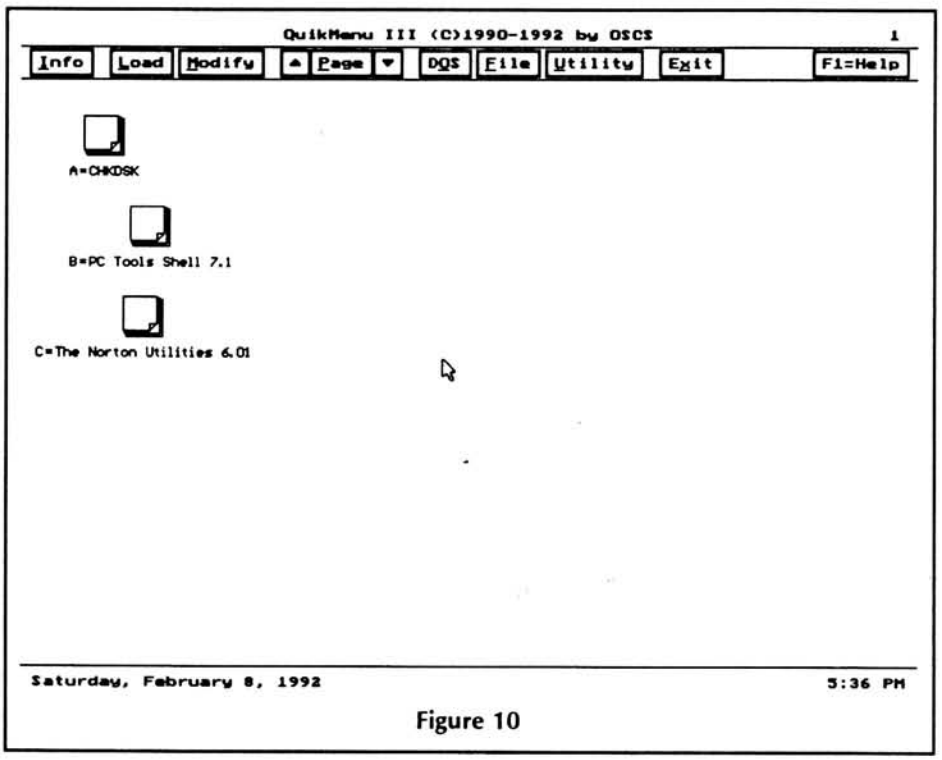


Figure 10

BORDERS.BAS

Robert W. Rasch
1504 Chickees Street
Johnson City, TN 37604

One of the features of having a personal computer is the ability to handcraft individual greeting cards and notes.

I edit my CRT screen with "Doodler" or "Showoff" to create individualized memos, stationary and gift cards.

While there are disks of "Clip Art" that can be picked from for various images and textures; they do not always meet my needs of the moment.

Frequently when I want a border or a background to add some zip to my handiwork "Clip Art" doesn't satisfy. Sometimes I struggle with designs developed by setting individual pixels on a small area of the screen. "Doodler" has a feature that "textures" (fills) an outline from a defined area of graphics. Drawing interesting borders or backgrounds can be time consuming.

Why not use numbers and geometry to do the task automatically?

BORDERS.BAS is a simple program that does just that. It makes a variety of complex and interesting graphic patterns that can be used as borders and backgrounds. The patterns can be stored in disk files and then used as additions to "Clip Art Libraries."

There are a number of formulae that make interesting patterns; all of them require repeated calculations which tend to be slow in interpreted Basic. I recommend the Basic Compiler, Quick Basic, or the C Language and a numerical co-processor to speed things up.

The illustration of a border pattern was produced by the numerical formula shown in the program listing.

Some other possibilities are:

```
PZ = SQRT(ABS(X*X + Y*Y))
PZ = SQRT(ABS(X*X + X*Y))
```

The latter formulas for "PZ" can provide changes in the "viewer window" by using variations in the starting values of X and Y. Try an XOFFSET of -320 as you explore possibile variations, you will find that you can "steer about" a window of graphic designs in a fashion similar to looking at different regions of the Mandelbrot Set.

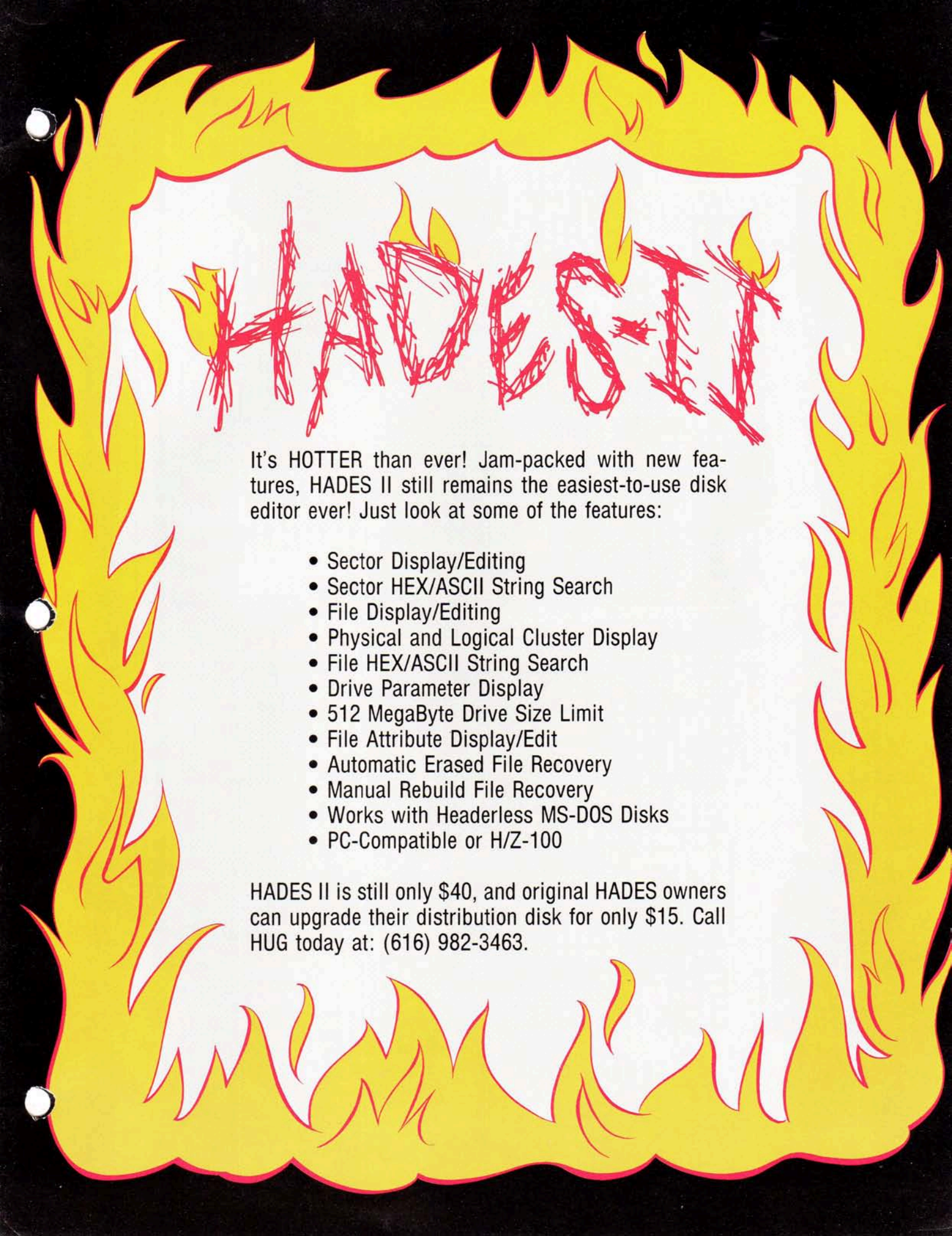
If you enjoy screen graphic design. You might find BORDERS.BAS an interesting way to enhance your ability to generate patterns.

Give it a try. There are only a few lines to type before you are generating patterns.

Showoff
HOGWARE COMPANY
470 Belleview
St. Louis, MO 63119
(314) 962-7833

Doodler
Paul F. Herman Inc.
3620 Amazon Drive
New Port Richey, FL 33553

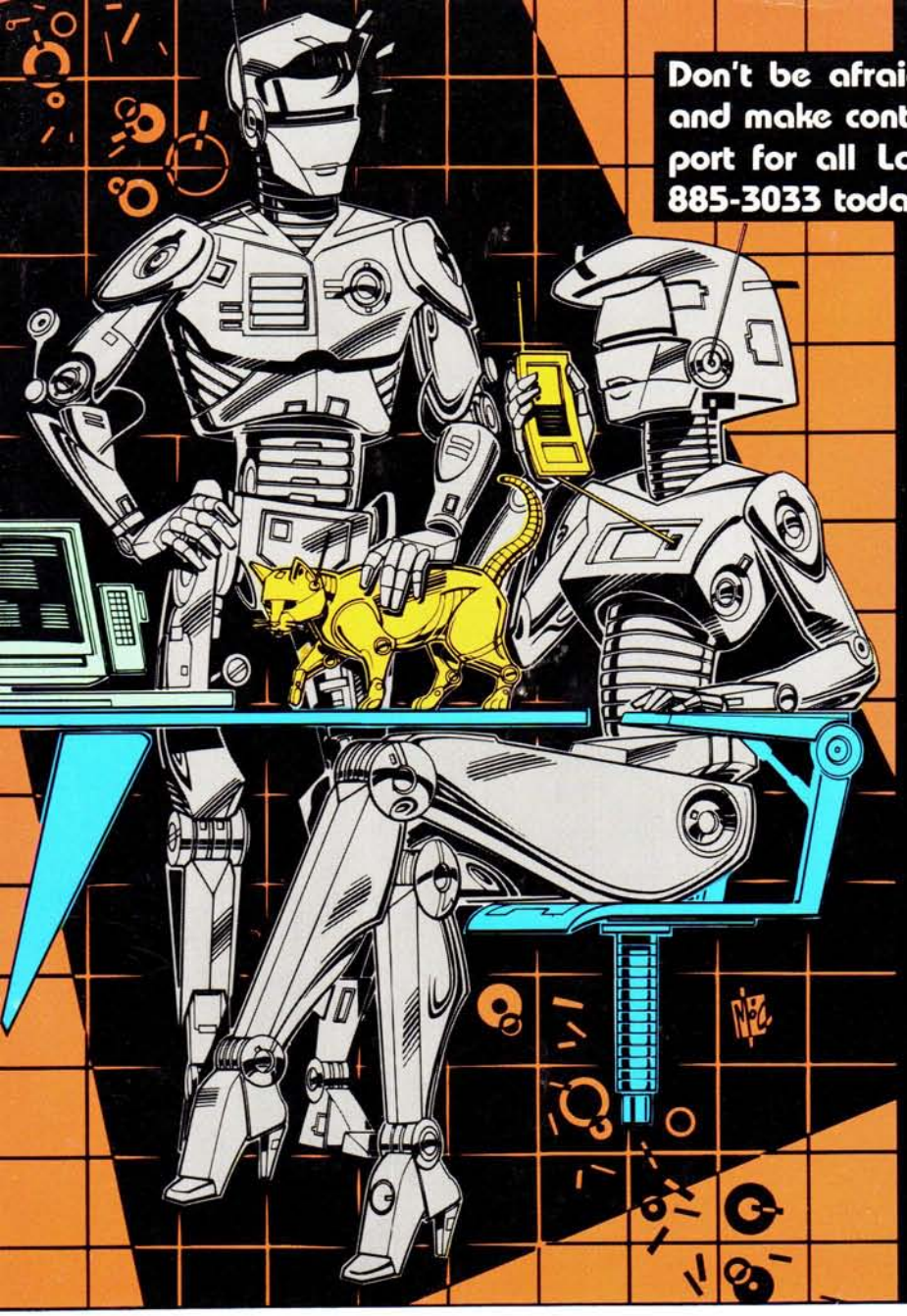
```
10 'SCREEN for whatever IBM Compatible Graphics Device
20 'If you have the H/Z100 you don't need to enter the above command
30 CLS
40 PRINT"PRESS 'Z' TO STOP"
50 INPUT"X-Offset";XOFF
60 VRESOL=220:HRESOL=640 'Enter whatever your screen has
70 REM Vertical and Horizontal Pixel Resolution
80 INPUT"Y-Offset";YOFF
90 CLS
100 LOCATE 1,1,0
110 FOR Y = 0 TO VRESOL
120 FOR X = 0 TO HRESOL
130 Y=Y+YOFF:X=X+XOFF
140 PZ=SQR(ABS(SIN(X*X)+COS(Y*Y))) 'This is the formula that you can change.
150 Y=Y-YOFF:X=X-XOFF
160 GOSUB 240
170 NEXT X
180 NEXT Y
190 BEEP
200 A$=INKEY$:IF A$="" THEN 200
210 A$=INKEY$:IF A$="" THEN 210
220 LOCATE 1,1,1
230 END
240 REM SEE IF ODD OR EVEN
250 IF (CINT(PZ) MOD 2) THEN PSET(X,Y),7
260 A$=INKEY$:IF A$="Z" THEN LOCATE,,1:END
270 RETURN
```



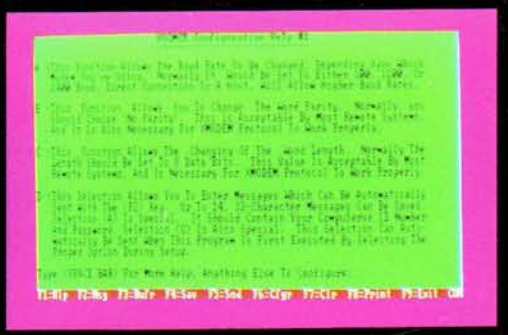
It's HOTTER than ever! Jam-packed with new features, HADES II still remains the easiest-to-use disk editor ever! Just look at some of the features:

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

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



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



ZENITH
data systems
Groupe Bull

BULK RATE
U.S. Postage
PAID
Zenith Users' Group

POSTMASTER: If undeliverable,
please do not return.

$2.50
P/N 885-2148