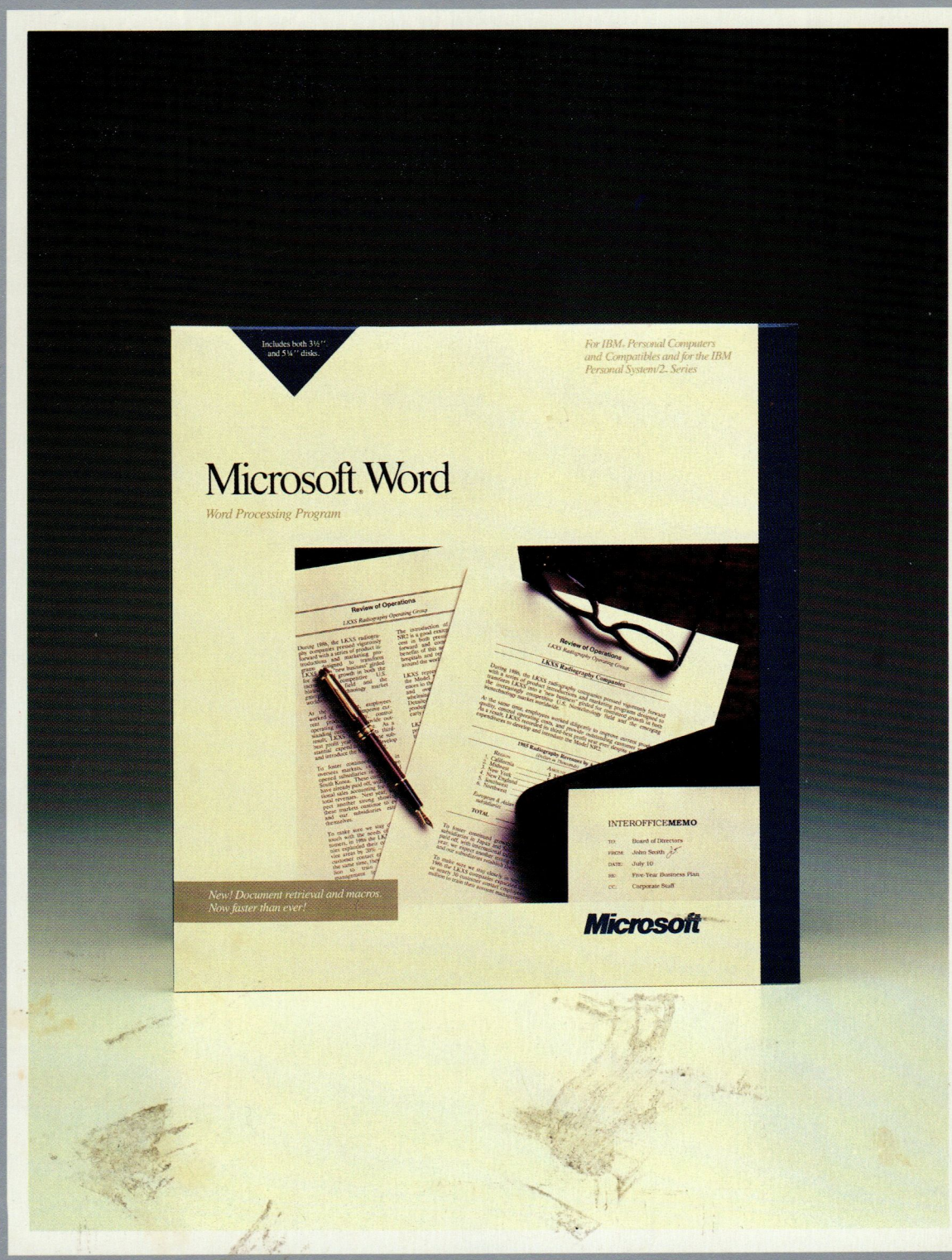
REMark®

Volume 11, Issue 8 • August 1988
P/N 885-2103 Issue 103

$2.50

Add More Than Two Floppy Drives to a Z-386
See Page 59



A Publication of Heath/Zenith


"I've Got
Your Number Now . . ."
"Now you have no reason for us to be strangers. I know each and every one of you personally! My human gave me the entire HUG Database to study and now there's no need for **any** HUG member to register. Just have your ID Number handy when you call . . . I do! I'll bet you didn't know that my Bargain Centre has discounts on HUG products, as well as other computer **and** non-computer related items. And, did you know that I can renew your HUG membership . . . at a discount?!! I can give you over 35 megabytes of free software. All you have to do is take it! Now no-one will **ever** have to know about the two of us. Just set your modem to 300, 1200, or 2400 baud, 8-data bits, no-parity, and call me anytime your computer needs someone to talk to. I can be contacted at (616) 982-3956. Although no longer required, my human can be reached at (616) 982-3837.
MdC

386 power for your 248

Now there are two sensational Upgrade Kits to give your 286-based computer kit the power of an H-386 Desktop.

The HUG-386 Upgrade Kit lets you convert your H-241 or H-248 series computer to a full H-386. **ONLY $2699**

The HUG-386-C Upgrade Kit adds a dual floppy disk/hard disk controller card to the HUG-386 to let you upgrade H-248's that don't have a Winchester controller card, or H-241's that have controllers incompatible with the HUG-386. **ONLY $2849**

If you're a HUG member and kit builder, you may be eligible for special discounts. Call for further details.

Available NOW at Heath/Zenith Computers & Electronics Centers in the U.S.

ARIZONA – Phoenix, 85017
2727 W. Indian School Rd.
602-279-6247

Tucson, 85711
5616 E. Broadway
602-745-0744

CALIFORNIA – Anaheim, 92805
330 E. Ball Rd.
714-776-9420

San Jose (Campbell), 95008
2350 S. Bascom Ave.
408-377-8920

El Cerrito, 94530
6000 Potrero Ave.
415-236-8870

San Diego (La Mesa), 92041
8363 Center Dr.
619-461-0110

Los Angeles, 90007
2309 S. Flower St.
213-749-0261

Pomona, 91767
1555 N. Orange Grove Ave.
714-623-3543

Redwood City, 94063
2001 Middlefield Rd.
415-365-8155

Sacramento, 95825
1860 Fulton Ave.
916-486-1575

Woodland Hills, 91364
22504 Ventura Blvd.
818-883-0531

COLORADO – Denver (Westminster), 80003
8725 Sheridan Blvd.
303-429-2292

FLORIDA – Hialeah, 33012
4705 W. 16th Ave.
305-823-2280

Jacksonville, 32211
9426 Arlington Expressway
904-725-4554

Orlando (Altamonte Springs), 32714
445 W. State Hwy. 436
Suite 1037
407-682-9111

Plantation, 33317
7173 W. Broward Blvd.
305-791-7300

Tampa, 33614
4019 W. Hillsborough Ave.
813-886-2541

GEORGIA – Atlanta, 30342
5285 Roswell Rd.
404-252-4341

HAWAII – Honolulu (Pearl City), 96782
98-1254 Kaahumanu St.
808-487-0029

ILLINOIS – Downers Grove, 60515
224 Ogden Ave.
312-852-1304

Skokie (Chicago), 60076
3606 W. Dempster
312-583-3920

INDIANA – Indianapolis, 46220
2112 E. 62nd. St.
317-257-4321

KANSAS – Kansas City (Mission), 66202
5960 Lamar Ave.
913-362-4486

KENTUCKY – Louisville, 40243
12401 Shelbyville Rd.
502-245-7811

LOUISIANA – New Orleans (Kenner), 70062
1900 Veterans Memorial Hwy.
504-467-6321

MARYLAND – Baltimore, 21234
1713 E. Joppa Rd.
301-661-4446

Rockville, 20852
5542 Nicholson Lane
301-881-5420

MASSACHUSETTS – Peabody, 01960
242 Andover St. (Rt. 114)
617-531-9330

Wellesley, 02181
165 Worcester Ave. (Rt. 9)
617-237-1510

MICHIGAN – Farmington Hills, 48018
29433 Orchard Lake Rd.
313-553-4171

East Detroit, 48021
18149 E. Eight Mile Rd.
313-772-0416

St. Joseph, 49085
2987 Lake Shore Drive
616-982-3215

MINNESOTA – Minneapolis (Hopkins), 55343
101 Shady Oak Rd.
612-938-6371

St. Paul, 55106
1645 White Bear Ave.
612-778-1211

MISSOURI – St. Louis (Bridgeton), 63044
3794 McKelvey Rd.
314-291-1850

NEBRASKA – Omaha, 68134
2311 N. 90th St.
402-391-2071

NEW JERSEY – Ocean, 07712
1013 State Hwy. 35
201-775-1231

Fair Lawn, 07410
35-07 Broadway (Rt. 4)
201-791-6935

NEW YORK – Amherst, 14226
3476 Sheridan Dr.
716-835-3090

Jericho, L.I. 11753
15 Jericho Turnpike
516-334-8181

Rochester, 14623
937 Jefferson Rd.
716-424-2560

N. White Plains, 10603
7 Reservoir Rd.
914-761-7690

NORTH CAROLINA – Greensboro, 27407
4620-C W. Market St.
919-299-5390

OHIO – Cincinnati (Springdale), 45246
131 West Kemper Rd.
513-671-1115

Cleveland, 44122
28100 Chagrin Blvd.
216-292-7553

Columbus, 43229
2500 Morse Rd.
614-475-7200

Toledo, 43615
48 S. Byrne Rd.
419-537-1887

OKLAHOMA – Oklahoma City, 73139
7409 South Western
405-632-6418

OREGON – Portland (Tigard), 97223
10115 S.W. Nimbus Ave.
Nimbus Center
503-684-1074

PENNSYLVANIA – Frazer, 19355
630 Lancaster Pike (Rt. 30)
215-647-5555

Philadelphia, 19149
6318 Roosevelt Blvd.
215-288-0180

Pittsburgh, 15235
3482 Wm. Penn Hwy
412-824-3564

RHODE ISLAND – Warwick, 02886
558 Greenwich Ave.
401-738-5150

TEXAS – Dallas, 75218
12022C Garland Rd.
214-327-4835

Fort Worth, 76116
6825-A Green Oaks Rd.
817-737-8822

Houston, 77008
1704 W. Loop N.
713-869-5263

San Antonio, 78216
7111 Blanco Rd.
512-341-8876

UTAH – Salt Lake City (Midvale), 84047
58 East 7200 South
801-566-4626

VIRGINIA – Alexandria, 22303
6201 Richmond Hwy
703-765-5515

Virginia Beach, 23455
1055 Independence Blvd.
804-460-0997

WASHINGTON – Seattle, 98109
505 8th Ave. N.
206-682-2172

WISCONSIN – Milwaukee (Wauwatosa), 53225
845 N. Mayfair
414-453-1161

Phone orders accepted.

Watch for our new stores in Pittsburgh, Memphis and Schaumburg, IL

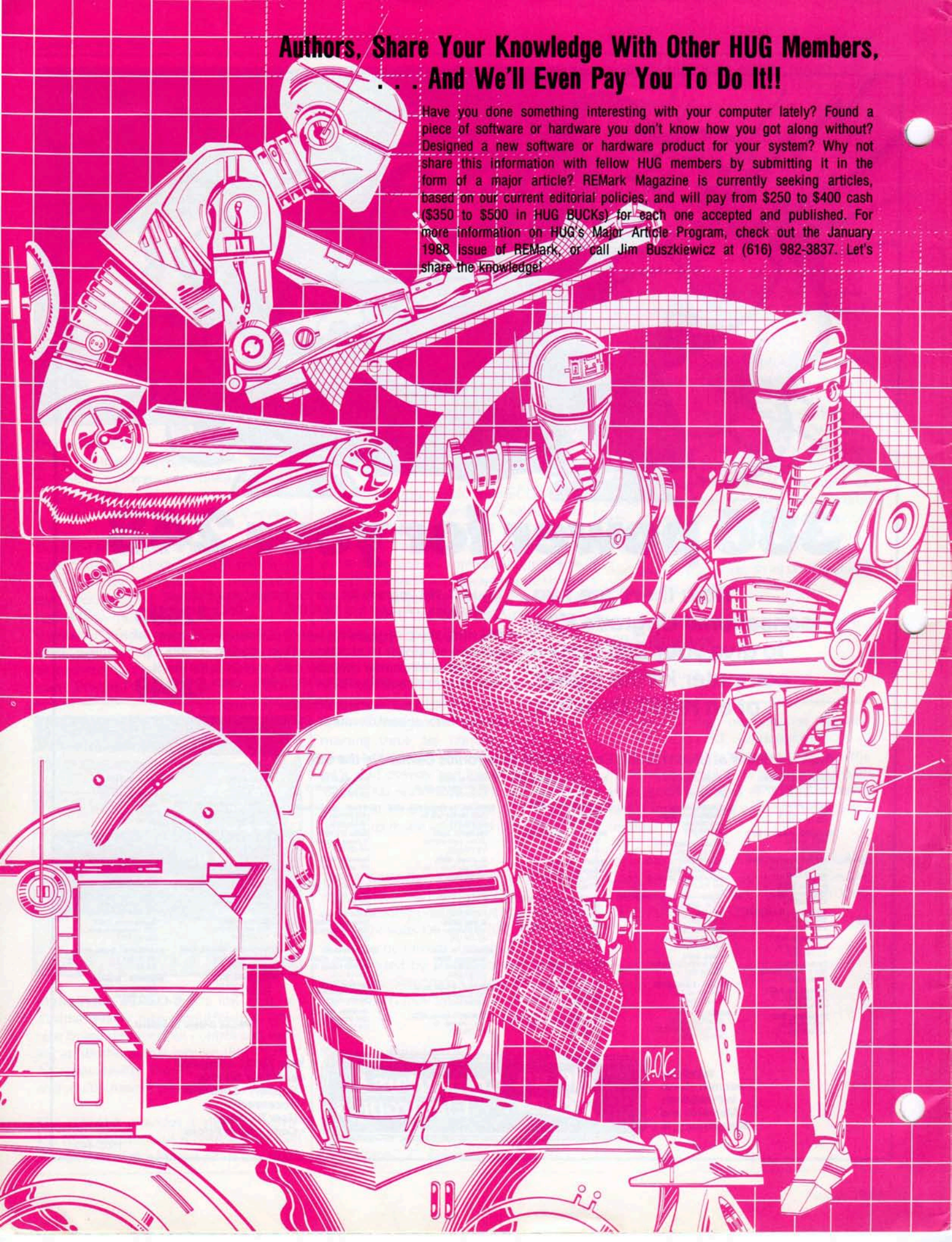
Heath / Zenith Computers & Electronics

Your TOTAL computer center for Service
- ***Support • Software***
- ***Accessories***
- ***User Training***
- ***Competitive Prices***

HZC-440

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

Have you done something interesting with your computer lately? Found a piece of software or hardware you don't know how you got along without? Designed a new software or hardware product for your system? Why not share this information with fellow HUG members by submitting it in the form of a major article? REMark Magazine is currently seeking articles, based on our current editorial policies, and will pay from $250 to $400 cash ($350 to $500 in HUG BUCKs) for each one accepted and published. For more information on HUG's Major Article Program, check out the January 1988 issue of REMark, or call Jim Buszkiewicz at (616) 982-3837. Let's share the knowledge!



On The Cover: The first having been presented in the May '88 issue of REMark, this is the second in a series of *Getting Started With . . . Microsoft® Word.* See Page 29.

REMark®

Managing Editor ..... Jim Buszkiewicz (616) 982-3837

Software Engineer ..... Pat Swayne (616) 982-3463

Production Coordinator ..... Lori Lerch (616) 982-3794

Secretary ..... Margaret Bacon (616) 982-3463

HUG Bulletin Board ..... (616) 982-3956

HUG Parts Ordering ..... (616) 982-3838

Contributing Editor .... William M. Adney

Contributing Editor ..... Joseph Katz

Printer ..... Imperial Printing St. Joseph, MI

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

* U.S. Funds

Limited back issues are available at $2.50, plus 10% shipping and handling — minimum $1.00 charge. Check HUG Product List for availability of bound volumes of past issues. Requests for magazines mailed to foreign countries should specify mailing method and appropriate added cost.

Send Payment to: Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022
(616) 982-3838

Features

Columns

Series

Resources

BUGGIN' HUG

Shared Experiences With 8-Bit Users

Dear HUG:

Browsing through the November issue of REMark encouraged me to believe that there may still be some "8-bit" computer users around (What? Maple update for the HDOS?!) and I decided to share my experience with some CP/M software which I came across recently. The piece I shall describe in more detail is the Hi-Speed C Library, from Viking C Systems. It converts your C/80 Compiler (from Software Toolworks) into, what seems to me to be, the most complete, versatile and portable C package in the CP/M world today. It requires the Mathpack if you use floats and transcendentals, and it also requires the Microsoft Assembler package of M80 and L80.

The C/80 package had built in the choice of using the M80 and L80 with supplied relocatable (.REL) libraries, and it included header files for this approach. One problem is that the C/80 in libraries are quite limited. The HiSpeed C Library uses this way of assembly, that is, it uses libraries in .REL format so that a function is included only if needed. When you combine the C/80 files with the HiSpeed Library, you end up with a flawlessly integrated system, with a main header, stdio.h, which directs traffic, plus a dozen other headers to be included as needed. The whole system contains over 300 library functions, most of which are portable. Some are not, mainly because they contain assembly code unique to the CP/M system or code for the H-89 H-19 terminal. The HiSpeed also adds many new math and UNIX style functions. It converts some "unusual" functions of the C/80 library to more standard form. It is hard to see how any new ANSI standard will contain functions not already in the package. Most important, it works without a hitch. I have written programs in BASIC and FORTRAN, but I am a novice in C. Yet, I had no really big problem compiling some C source which was gathering dust because of (up to now) non-portability.

You can buy just the basic disk of libraries in .REL and related files (no source code) plus the manual. That is about 300K. I feel, however, you would be missing much if you did not also get the source code plus some other goodies that come with it. The total package has about 1.4 megabytes!

The manual that comes with it is excellent. It does not waste any words. The best part is that it has an alphabetical listing of the functions, one per page. It lists the function name, summarizes what the function does, and includes a small, self-contained C source example which you can use to see how the function works. To the inexperienced programmer like me, the manual is an invaluable learning tool, worth the price of the whole package.

That is not all. It includes a program for generating portable source code. It also includes several executable or .COM utilities. There is an "Archiver", the best I have seen. It not only lets you put in or take out several similar files in one "drawer", but you can view all or part of a file without exiting the Archiver. I had another one which I hardly ever used because I soon forget what the file name stood for! (When will we ever get rid of the curse of the eight letter title GUESSWHA.T??) There is a comparator (LEXI) of two files to locate and mark changes in files like in revised manuscripts, sorters and a Key Word Index rotator. Besides the .COM files, there are many utility programs in C source code, ready to assemble. In BASIC and FORTRAN, I felt sure of what I had to deal with, even if I did not know how to go about it. In C, I was pretty much lost. Now, having the HiSpeed makes me feel I have a tool I can use and the possibility of learning how.

I should stop here, were it not for J. Katz' "elephant" story in his "Mainstream Computing" in the November '87 REMark, which reminded me of one problem we "8-bit" users have, that is, lack of a decent, manageable database. Yes, there is (was?) dBASE II, the "elephant". How about a mule, like the PC-file? For my needs which do not require the acres of memory of the PCs, the HUG Autofile was good for filing several hundred journal references. However, I could not selectively extract portions, like author's names sorted in a particular order, for inclusion elsewhere.

Enter the "Labeler", again from Viking C Systems. The only thing wrong with this CP/M database is its name, because it misleads one to think its purpose is to produce mailing labels or something. It does make labels of everything on anything in a flash, but above all, it is a versatile random-file database which accepts records up to about 1.5 Kbytes each. Let me hasten to add that this 1.5K is also misleadingly small, because the author was mindful of the memory limitations of the 8-bits and got around this problem in a couple of very ingenious ways. I shall not get into how it is done. Suffice to say that it can "link" two or even more databases in such a way that items with something in common can be pulled out and put together in a common "report". For example, one can have in a "separate" database the names of authors of an article and the journal, and in another a summary of the article. The advantages are obvious: one database can contain many references of authors' names on one disk and speed through locating the desired ones. Another set of commands brings in quickly the related summaries of the located articles from the "kin" database from another disk. You can do it this way only if you want it so. You have control over the format. The "language" or commands are simple and not many, about two dozen. To repeat, the 1.5K limitation is deceiving, first, because it refers to one record in one database only and does not include its "kin". Secondly, text written on an editor can be included in a "report" essentially without limit. The software includes manual and both source and object code, and again it is voluminous, over 700K. The files of the Labeler are transferable to other databases like dBASE II and III, both CP/M and MSDOS. I should also point out that there is a version of Labeler available for MSDOS.

For those curious, Dr. Grant Gustafson is the main (if not the only) author in Viking C Systems. Some "veterans" may remember his CRASH utility for HDOS and CP/M disks. "Corrupt" disks become quickly "virtuous" and readable, thanks to it. He has also written many other programs for the Heath computers in both HDOS and CP/M, various utilities, printer drivers and others.

When writing for PCs, one can have a column such as W.M. Adney's in REMark, entitled "On the Leading Edge." Recognizing our relative position in the world of fund producing adventures, we "8-bit" users can be more reserved and refer to a one time write-up entitled "On the De-

Continued on Page 44

Xenix Part 3

Matt Elwood
1670 N. Laddie Court
Beavercreek, OH 45432

Basic Commands

In the last article of my XENIX series, I showed you how to install XENIX and add users to it. In this third article, I will talk about some basic file commands in XENIX and the mail system. XENIX is the PC version of UNIX so the commands I discuss in these articles should work in UNIX. As I said in the first article, MS-DOS has a background that leans towards UNIX, so some of the commands will look familiar to most of the readers, as MS-DOS is the normal operating system.

To start, first log on as a normal user (not the super user). In the last article, the normal users were added to the system, so you must use one of these users. If you, or the name you select, have not been added by the "super user", log on will not be possible. Each normal user has their own directory which contains only the user's files. This is called your home directory. In MS-DOS, a single user system, this is the same as the root directory. The home directory is empty until you put a file in it yourself. The super user's directory is the root, for he is the only one that can change the system files. You can refer to your home directory by using '~' instead of the full name. Note that in the presentation from the XENIX login screen the password is not displayed. Also note that the terminal type is 'z100' which I create from the basic H-19 file and is part of the terminal capability file and called from the .login file. I logged in from the Z-100 using ENABLE in order to capture the screen displays.

```
Login: elwood

Password: [not echoed to the screen]

Welcome to XENIX System V/286

Terminal type is z100
```

Like in MS-DOS, you can list the contents of your directory. The directory listing command is different in XENIX, though. To display the directory, type 'ls' which is the equal of 'dir'. 'Ls' has many switches that can be specified on the command line. Typing 'ls' alone will give you a vertical one-column directory output like this:

```
account
user1
notes
savedmail
```

'Ls' uses pattern matching like most of the other commands in XENIX. Three characters are used for this, the asterisk, the brackets, and the question mark. The asterisk can mean any amount of characters after the text you have specified. For example, 'ls fl*' will list the file names of 'flank', 'fl001' and 'fl'.

The second pattern matching device is the bracket. This is used for one character. Inside of the brackets are a list of characters you want to use. You can abbreviate a list by using a dash. For example, 'ls user[45679]' would list the files 'user4', 'user5', 'user6', 'user7', 'user9', but not 'user 1, 'user2', or 'user0' You can specify the above a different way using the dash. It would look like this: 'ls user[4-79]'. This would list 'user4' through 'user7' and 'user9' and no other.

The last character is the '?' and it stands for one character. An example of using this would be 'US?'. This would include 'USE' but not 'USER'. These are the three pattern matching devices that can be used in XENIX.

If you typed 'ls' to list the files in your directory before you created any, nothing would be displayed. But for every user, regardless of what shell, there are '.' files created. They are not shown by using 'ls' but can be shown if you add a '-a' command line switch after 'ls'. If you want to know more about files, like their sizes, you can put '-1' after 'ls' for the long listing. A long listing would look like this:

```
2%:ls -la
drwxr-xr-x 1 root  default  1024 dec 26 12:52 .
dr-xr-xr-x 3 root  default  4096 dec 23 17:49 ..
-rw-r--r-- 1 matte default   173 dec 26 12:53 .profile
-rwxr--r-- 2 matte default 17660 dec 27 13:57 memo
```

The first two files are actually directories. The directory '.' is really a pointer to the current directory and the '..' is a pointer to the directory one step above this directory in the tree. You can see similar displays in MS-DOS when directories and sub-directories are present.

The first field in the long listing is the permission bit field. The first character in this is the file type bit. This would be a 'd' if the file were actually a directory, a 'b' or a 'c' if it were a device file (device files should only be found in the directory '/dev'), and just a dash if it is a normal file. The rest of the characters in the permission bit field are the actual file permissions. These are divided up into three groups of three, each for a different set of users. This is how XENIX/UNIX establishes file protection.

In the three bit sets, each one is for a different function. The permission bits are displayed when the person is allowed to complete an operation and is a dash if the person is not permitted the operation. The first bit is the read bit. This means, if it is set, the set of people in that group of three bits can read it. The next bit is write, which means the people can write to it and the last bit is executed. This can only be set if it is a script (like batch in MS-DOS) or binary executable file.

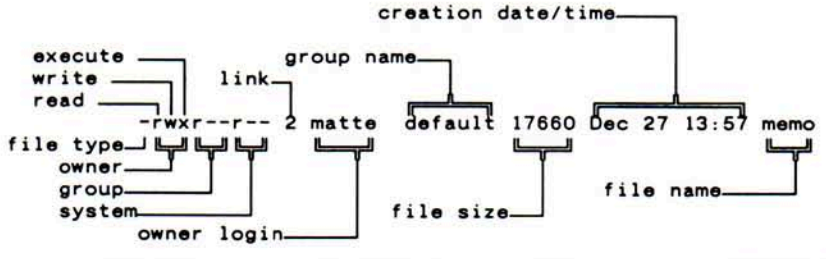
are only one file being accessed by many different names in different directories.

The next field to the right is the login name of the owner. The owner is the person who wrote or compiled the program. To the right of that is the owner's group.

One field to the right of that is the size of the file in bytes. This is the same as MS-DOS's file size, and there is nothing special to report about it for normal files and directories.

To the right of that is the date and time the file was last changed. Like MS-DOS, this is useful for keeping track of versions of files. And lastly, on the very right is the file name. File names in XENIX have no extensions, but there are some limitations. File names can only be 15 characters long.

The next commands we will talk about are the 'mv' and 'cp' commands. They are commands that do the same things as MS-DOS' 'rename' and 'copy' commands. They work with a simple syntax 'mv filefrom fileto' or 'cp filefrom fileto'. For example, to copy 'rpt1' to 'report' in the same current directory, you would type 'mv rpt1 report'. To copy this same file, you would type 'cp rpt1 report'.



Each one of the groups of three is for a different set of users, like I said before. The first group of three is the owner's permissions. You can find out the owner by looking two fields to the right. This is the owner's name and group name. The next group of three is for the group the owner belongs to and the last group is for the rest of the people on the system.

The next field to the right of it, numeric field, is the number of links to the file. You shouldn't have to worry about it for links

XENIX, as I discussed before, has a large directory structure. To take advantage of this, you can use the same commands as MS-DOS, 'mkdir', 'cd', and 'rmdir'. Mkdir makes a directory as a sub-directory to the current directory. If you don't have write permission to the current directory, you may not create the directory.

The command 'cd' changes the current directory. The directory structure in XENIX uses forward slashes '/', instead of back slashes '\' like MS-DOS. Also different from MS-DOS is when you just type 'cd'. In MS-DOS, it displays the current directory. In XENIX, it takes you back to your home directory.

The next command is a combination of 'type' and parts of 'copy' in MS-DOS. It is called 'cat', which stands for concatenated. This is used in many ways. The first way is to display a file on the screen. You just type 'cat', then the file name you wish to display. 'Cat' isn't a real good file lister for the display scrolls as fast as the computer or terminal receives data and can present it on the screen. A substitute for this is 'more' which presents the file a screen at a time. There are a lot more things you can do in 'more', and these will be discussed in later articles.

The other way you can use 'cat' is to combine files. You do this by typing 'cat', then the name of the files to be combined, then '>' and the file name you want to be the output file. The only limit for this procedure is the length of the command line. An example would be 'cat file1 file2 file3 file4 > file5'. The greater than and less than signs will give us our next subject. The screen display of a short mail message is displayed below.

```
6% cat test.ltr
From elwood Thu Apr 14 22:09:53 1988
To: matt
Subject: test of mail
Date: Thu Apr 14 22:09:53 1988

This is a test of the mail system.
```

Many commands in XENIX will get input files, but will just give output only to the screen. To output or input from the disk, you can use I/O redirection. This is exactly like MS-DOS' redirection. It is used by giving the greater than and less than signs. The greater than sign '>' and a file name after the sign is used to take the output of a program and output it to a disk file. This will give you an error if the file already exists. If you want to append to the file, you can just add another greater than sign so your command looks like this : '>>filename'. The less than sign '<' proceeded by a file name is used to take the input from the file specified in the file name. These are easily confused so watch out so you don't destroy your files!

Since XENIX is a multiuser system and people own files on the system, there has to be ways to change the permission bits and owners. The first command to do this is 'chown', which changes the owner of a file. You must be the owner of the file to

change the owner to another person. To use 'chown', you just type 'chown' and then the user ID of the person you want to be the new owner of the file. The next command is 'chmod', where you type 'chmod' then a three digit bit mask number. The first digit is for the owner, the second for the group, and the third for everyone else. Each digit is a bit mask. You add 1 for read permission, 2 for write permission and 4 for execute permission. For example, to give yourself all permission and all others execute permission, you would type 'chmod 744 filename'. The '7' is a sum of the permissions and results in the owner receiving all the permissions, while the group and system receive execute permission only.

To login to the XENIX system, you must type in the password that was assigned by the superuser when he established you on the system. XENIX provides the means to change this password so that the superuser can't access your file or for a routine change. You should change your password often with 'passwd'. You need to give your old and new password for this. To change your password, type 'passwd', XENIX will prompt for your old password and then ask for the new word. You then have to retype it a second time to ensure that you correctly typed in the new password. Following is the actual screen display for passwd.

```
1% passwd
Changing password for elwood
Old password: [not echoed to the screen]
Enter new password (minimum of 5 screen]
Please use a combination of upper and lowercase letters and numbers.
New password: [not echoed to the screen]
```

That's all of the simple commands. To use these commands, you need a shell. Even though this is automatic, I will now talk about some simple shell concepts. In the XENIX shell, you just need to type the command name. You can put two or more commands into the shell two ways.

The first way is the pipe '¦' which takes the output from the program on the left and inputs it into the program on the right. An example of a pipe command would be 'ls −1 ¦ more' which would display a long listing, but only one page at a time when you type 'ls−1'. The other way is the semicolon ';' which just executes both of them. An example would be 'vi filename ; spell'. This would run spell as soon as you quit vi.

Since XENIX is mutlitasking, you can run a long process in the background while doing something else. To do this, just add an ampersand '&' at the end of a line. This is called a background process. You can kill all of the background and foreground processes by using the 'kill 0' command. To kill a specific process, you use 'kill', then the process ID. To find the process ID, type 'ps'. The PID is the process ID. If you start a background process and then forget it and try to log out, the system will not permit you to leave until the process is completed or it is killed.

XENIX, like UNIX, has a mail system where you can send messages to other users on the system. All of the mail functions are included into one program called "mail". To start up the mail system, just type 'mail'. You can only use this to read or reply to mail. After you type "mail", if you have mail, it displays "You have mail." and a listing of the headers of each message. To read each message, type "r". To read a specific message, type "r", then the number of the message you want to read. Once you have read the messages, you may want to delete them. If you want to delete all of the messages, type "d*" or to delete only one, type "d" then the message number. To send mail type in 'mail username'. The user must be a name on the system or it will become lost mail. XENIX will then prompt for the message subject. XENIX will then permit you to enter your message. When complete, type "CTRL D" and XENIX will display (End of message). If you tried to send mail to a user that is not on the system, XENIX will save the message to a dead.letter file in your directory. If the file does not exist, it will create one for you. Every additional letter that you tried to send to a non-user will be appended to the file.. The screen display is presented below for this condition.

```
2% mail matteo
Subject: dead.ltr test
This is to show what will happen when you send mail to a nonuser.
(end of message)
No local user named "matteo"
Letter saved in "/usr/elwood/dead.letter" [Appended] 7/171
```

That concludes our discussion of basic shell commands. Next time, we will talk about vi which is the visual editor in XENIX/UNIX and one of two available with the XENIX package. Using vi, we will create short documents and use screen printouts to explain the function of the different commands. *

Don Powers
P.O. Box 969
Marion, MA 02738

Improving The H-151's Graphics

It is my desire to improve the graphics and other capabilities of my H-151 computer, but not increase its speed. Why not increase the speed? First, I feel my financial resources are needed to add capabilities, rather than throughput. (I've seen software execute at least four times faster on an H-248 than on an H-151 and agree it's a significant improvement; the Norton System Information (SI) numbers for the two systems are 8.7 and 1.8, respectively.) If I increase clock speed, I need to replace my 8087 co-processor and part of my RAM memory with faster devices. Both of these items are very expensive.

Better Graphics and Memory

While I wish to improve the graphics, and take advantage of certain features of upper 64k bytes of memory, there are certain restrictions that apply. I wanted to remove my present monitor card (a Z-309), while maintaining the ability to use it again later through board replacement. (Requirement: no other changes made to the system should have an impact on this ability.) This means I don't give up any additional slots. (I don't have any slots to spare.)

After seeing the 14", "flat" ZCM-1490 monitor, this is my first choice. The ZCM-1490 and its accompanying Z-449 full-length Video Graphics Array (VGA) board seems like the ideal combination. According to "BYTE" (Mar. 1988, pg. 112), the board has one drawback, it does not operate in mode 13h (256 colors, 320 *200 pixels). In my graphics application needs, higher resolution takes precedence over number of colors. Thus, this isn't a great loss.

However, cost is still a primary factor, so that combination has to wait until the budget can accommodate it. (The board and monitor retail for $1500, almost what I paid for my first H-151.)

The logical cost alternative for now is an Enhanced Graphics Adapter (EGA). The format is not so new that there aren't a number of manufacturers out there with boards and monitors. More importantly, with the advent of the IBM PS/2s, the standard has shifted to VGA. Like my H-100, EGA is "obsolete", and like my H-100, it certainly works well.

To help clear up my last statement, I purchased an 8 MHz low-profile H-100 long after I bought an H-151; the H-100 is still a better machine. Besides being faster with better graphics, the dual processors run almost any software and operating system. As a writer, I initially needed the H-151 for software evaluations; as an engineer, I appreciate the better S-100 bus structure and dual processors of the H-100.

I use the term "obsolete" in terms of EGA because IBM is the supposed "technology trend-setter". Actually, they are just so big that they dominate by size and reputation. Compare the PC/H-151 and H-100. The H-100 was first and better, IBM was the biggest seller. More people try to make their H-100 run PC-compatible software, than the other way around. So now the "standard" will shift from EGA to VGA.

But don't worry all you EGA users, there is a large enough number of standards, and EGA has been out long enough, so that it won't vanish from the scene. What I do see is a slight softening in EGA prices, which is starting to make EGA more affordable.

While I tried a couple of different video boards in the H-151, for simplicity I'll just refer to any of these boards as the new video board. (Note that in all cases tested the Z-309 had to be removed to prevent possible interference.) There is one mechanical problem with substituting video boards, the extra 1" backplate height on the original Z-309 board. This makes a direct replacement of boards slot-for-slot impractical, so another slot was used.

However, you can save the slot. I noticed that the mounting holes between bracket and board are similar for all standard Zenith boards. I also noticed that the bottom of the 9-pin connector on the Z-309 video board lines up perfectly with the bottom of the parallel port on the memory board. You have to expand the 9-pin cutout on the Z-309 mounting bracket. Next, drill a hole for the top screw post. The two screw mounting holes on the memory board will also have to be either drilled or slotted to the rear of the board by 1/8" for proper mounting. Either use a small round small file to make the slot, or mount the bracket only to the DB-25 connector. Drill through the mounting bracket screw holes with a very small pilot drill. Then remove the bracket and redrill with a #33 sheet metal drill. I did two boards and found the second method more satisfactory. Now you can attach the bracket to the memory card with the two plastic screws. Remember to keep the two plastic insulators between the memory board and bracket when you make the change. Otherwise, you tie the 0 volt line to the bracket. You can now mount the memory card in the Z-309 slot.

Using the Z-309 Again

OK, so what if you want to put the Z-309 back into the H-151? You can't replace the board in the same slot once you've removed the bracket and modified it for the memory board. One answer is to not use it for the memory, but instead leave the slot empty. (I, however, do not have slots to waste.) If you try to order an extra bracket from Zenith, they'll sell you a "replacement" bracket that is one inch shorter. (See below for part number and price.) It's not a direct replacement, obviously. At the same time, it does fit the Z-309 properly.

In some cases, this might not be the solution. If your H-151 has a card slot guide bracket in the front, the Z-309 board is much too long. I have more than one H-151s; the last one has no such bracket. (It was bought at close-out prices, already built.) While this exposes the speaker, normally hidden by the front bracket, it leaves the spaces on each side with enough depth to allow the insertion on the Z-309 with the shorter rear-mounting bracket. This is the method I used to solve the problem, and others may find removing the front bracket to be the solution, as well.

H-151 Memory

The H-151 provides a challenge to this proposed configuration. While the monitor EPROM is on the processor (CPU) board, the RAM memory that it requires is on the video board. The obvious solution is to have the RAM on the same board as the monitor EPROM.

Basically, the monitor/boot-ROM consists of two 16k-byte EPROMs (27128). Their addresses are decoded by a fuse-link type PROM. U208 contains the lower 16k (f800:0000 through f800:3fff) and U207 contains the upper 16k (fc00:0000 through fc00:3fff). These two EPROMs combined take up the last 32k bytes of regular memory (f800:0000 through f800:7fff).

The RAM used is 16k located from f000:0000 through f000:3fff on the video

card. (A second 16k of RAM exists on the video card from b800:0000 to b800:3fff as video memory.) Only about the first kilobyte of monitor RAM is actually used by the monitor ROM.

The following memory map of the "normal" H-151, Figure 1, may clarify this arrangement.

Start Address	Last Address	Memory Type	Memory Location
f000:0000	f000:3fff*	DRAM	VIDEO BD
f400:0000	f400:3fff	NONE	--
f800:0000	f800:3fff	EPROM	CPU U208
fc00:0000	fc00:3fff	EPROM	CPU U207

* Only the first 1k byte used.

Figure 1
Normal High 64k Memory Map

When adding an EGA video board, remember you lose some of the additional addressing capabilities, from a000:0000 through a000:ffff (64k bytes). (This may be different for other video board standards, so be certain you know the map location for your new video memory.) At the same time, be certain you don't extend your main memory decoding into this block. Some non-Zenith RAM address decoders that allow you to use 256k bit chips may map past the 640k boundary.

With this as a basis, Dave Brockman, of FBE Associates, offers one solution, a PROM that replaces U236 on the CPU board. (They also have a PAL RAM decoder that limits main memory to 640k, the MR-150T described below.) However, you have to squeeze both EPROMs on the CPU into a 32k byte (27256 or 27C256) EPROM. The other socket is then used for RAM (32k*8 types 62256 or 43256C).

To accomplish this method of video board elimination, purchase FBE's decoding PROM, part number RM-150 for $9.95 with instructions for the conversion. (See REMARK, Nov. 87, pg. 27. The price since then has increased, but only enough to cover costs.) You can get a second one that costs $3 without the instructions. It decodes the EPROM in socket U207 as starting at paragraph f800, and the first 16k bytes of RAM in socket U208 as RAM starting at f000:0000. The decoding PROM (RM-150) at U236 leaves the other 16k of RAM unaccessible. (FBE sends you instructions on how to use a 6264 (8k byte) instead; but this requires soldering to the chip.)

The next step is to change the ROM select jumpers below U208. The jumpers should be in the two outer-most pairs, (256k for ROM1 and the RAM position for ROM2. Then, all you have to do is change video cards. (Note that you can still use the old video card, Z-309, with this configuration in, since both RAMs load up at the same time with the same information.)

Since I have an EPROM programmer, that has software for reading the monitor, I initially opted for this solution. I am also aware that Dante Bencivengo offers a board solution, the VMM-150 Video Card Eliminator at $45, and for those not interested in tearing apart their computer boards this might be the ideal solution.

While I haven't seen the kit, I've been informed that it includes a punched bracket for mounting a memory card in place of the video card, so that this space doesn't go to waste.

I called Dave Brockman, at FBE, to order the parts I needed. In addition to the decoding PROM for the CPU, I bought a special PAL to "limit" the main memory board decoding to 640K. (FBE offers two others, including one that provides 704K of main memory.) The PAL I purchased, the MegaRAM-150 T (MR-150T), for $49.95, which replaces U444 on the memory board and can take as many as five banks of 256k-bit RAMs. In addition to the 640k of main memory, by loading the board fully, you can get an additional 514048 bytes in RAM disk. The rest of it, about 10k bytes, is used for FATs and the directory. This way, you insert the EGA card with no worry that the memories will overlap. And you get a 514k byte RAM disk without installing an additional EMS board, or losing all of your main memory.

My other purchases were from Zenith. I purchased the latest monitor EPROM set, Version 3.0B, for $22.10 plus shipping. The parts sport 16 suffixes (444-229-16 and 444-260-16). In addition, I tried to purchase brackets for the Z-309 board. The original part (204-2719) is no longer stocked and has been "replaced" by part number 204-2770, at $1.50 each. These "replacements", as I mentioned before, are one inch shorter.

The parts arrived from FBE, along with a set of very clear instructions. The first step is to remove all RAM and components to be switched from the memory board. At that point, replace the bracket on the Z-309 with the shorter rear bracket, and modify the longer Z-309 bracket and mount it on the memory board.

Before installing the RAM and new PAL (MR-150T) in the memory board, a wire needs to be added. FBE includes a jumper for the memory board with ball clips on each end. While the jumper is high quality, I prefer not to rely on one. Solder a wire on the back of the board to the same points that the jumper is supposed to go on the front. One end of the jumper goes to an IC, U458 Pin 10. To protect that IC, a 74LS174, remove it also before soldering the wire. Replace the components, and the memory board is complete. You can put the memory board in the video slot and put the Z-309 in any slot with which the speaker doesn't interfere. (Again, you can't do this if the card-edge guide bracket is present.)

The final step is to replace the two present 16k byte EPROMs with a 32k byte EPROM and static RAM on the processor board. Also, change the decoding PROM, U455, with the FBE RM-150 PROM. Figure 2 shows the resulting memory map.

Grab All The RAM You Can

At this point, everything works and its time for the last part of the current updates. (I plan more.) The configuration isn't totally satisfactory because it wastes too much of the static 32k byte static RAM. First, there is 15k bytes of RAM just sitting there, assuming you leave 1k for

Start Address	Last Address	Memory Type	Memory Location
f000:0000	f000:3fff*	SRAM	CPU U208
f400:0000	f400:3fff	NONE	--
f800:0000	ff00:3fff	EPROM	CPU U207

* Only the first 1k byte used.

Figure 2
High 64k Memory Map

the monitor EPROM. Secondly, the other 16k bytes can't even be accessed. The upper address bit of the 32k byte static RAM, Pin 1 of U208, is tied to +5V under the socket. (You actually only use the top half of the RAM mapped to the lower half of the address space.)

I had already read two recent articles in "Sextant" by David D. Clark, (Nov.-Dec. 1987, Issue #31, page 58 and Early Spring 1988, Issue #33, page 20). These articles deal with the use of unused memory in this area to hold memory resident programs. Among the utilities listed in "Sextant" are a keyboard buffer program (Nov.-Dec. 1987 issue) which creates a type-ahead buffer in this memory, and a keyboard break program (Early Spring 1988 issue) which allows you to invoke the MFM-150 monitor program via a "Hot-Key" combination. ("Sextant" sells these back issues of their magazine, if you don't have them, for $3.50 each.)

No matter what your use might be for this memory, the potential is there. And, not using available memory is a waste of a computer's potential resources. (David Clark sells a 5-1/4" floppy disk containing the programs listed in the articles, plus others, for $25.)

I checked the decoding on the RM-150 PROM, and found that only address paragraphs 0f000 through 0f3ff (addresses 0f0000-0f3fff) are decoded to access the static RAM. The first step requires getting a PROM that decodes paragraphs 0f4000 through 0f7fff as part of the same static RAM.

I called Dave Brockman at FBE Research Company; he said he'd make such a part if requested, changing the designation to RM-150A. (It's the same price as the RM-150. If you order one be sure to mention the RAM memory locations that you want decoded.)

The second part of the solution requires doing something about Pin 1 on the static RAM, now a solid "1" because of its +5V limit. While lifting the pin from the socket and soldering a wire between Pin 1 on the RAM and address line BA14 solves this problem, I prefer not to solder to an IC if I can avoid it.

My solution calls for a little more care, and is easily removable. Prepare a 3-inch piece of #26 or smaller wire-wrap wire (insulated, single-strand) by removing a quarter of an inch of insulation from each end. Next, take an IC solder socket pin (high-reliability type), and solder it to one end. Insulate the pin with a short piece of sleeving, ideally heat-shrinkable.

Temporarily remove the "256k ROM" jumper decoding U207 (the jumper on the left). Slip the unused end of the wire-wrap wire through the jumper. Reinsert the jumper at the same points with the wire entering the jumper from the bottom. This should make a secure connection to address line BA14.

In inserting the jumper with wire back on the "256k ROM" pins I experienced a problem because the pin entrance was too small with the wire inserted. Always keep an extra supply of 0.1" spaced jumpers on hand. I purchase them from JDR Microdevices at five for $1; they have larger openings on the ends. (JDR also has low prices on high-rel IC sockets.) It is also considerably safer to just replace the jumper than to try to cut or scrape away the jumper plastic.

Remove the static RAM from U208 and bend Pin 1 out so that when the chip is reinserted Pin 1 will be clear of the socket. Attach the socket pin to Pin 1 of the static RAM, and reinsert the RAM in the board.

Replace the PROM at U236 with an RM-150A from FBE, and this part should be ready to run. If you have Norton's SI utility, you should read that in addition to main memory (640k) and video memory (64k) there is an additional 32k bytes from 0f0000 through 0f7fff. If you use David Clark's routines with 32k bytes of static RAM fully installed, remember to change the .ASM files so that the "LastFree" equate, which defines just past the upper paragraph of static RAM, has a value of "0f800h", not "0f400h".

Naturally, you'll need a macro-assembler, such as Microsoft's MASM. This is available in the Zenith Programmer's Utility Pack (PUP) software package. While I own the PUP package, and the version of MASM in it should work, I also have the more up-to-date MASM version (5.0) which I purchased from a non-Zenith retailer.

The final memory map below, Figure 3, should now hold true.

Products Mentioned

FBE Research Company, Inc.
P.O. Box 68234
Seattle, WA 98168
(206) 246-9815

Heath Company
Parts Department
Hilltop Road
St. Joseph, MI 49085
(616) 982-3571

Continued on Page 14

Start Address	Last Address	Memory Type	Memory Location
f000:0000	f000:7fff*	SRAM	CPU U208
f800:0000	f800:7fff	EPROM	CPU U207

* First 1k byte reserved for monitor with 31k (f000:0400 - f000:7fff) for users programs.

Figure 3
Revised High 64k Memory Map

HUG Discount List

AA-3000	20%
AB-3000	20%
CB-5063-50	10%
DC-1000	10%
EN-2000	20%
HS-248-Z	10%
HS-248-ZX	10%
HS-386B	10%
HSM-100	20%
HSM-100-3	20%
HV-2000	10%
HWD-20	10%
HWD-20-AT	10%
IN-6000	20%
PA-120	20%
PD-500	10%
PM-160	20%
PM-160-3	20%
PMK-121	10%
PMK-130	10%
SK-201	10%
SK-202	10%
SK-203	10%
SK-204	10%
SK-205	10%
SK-209	10%
SK-210	10%
SK-211	10%
TM-140	10%
TM-150	10%
TM-158	10%
TM-159	10%
TM-170	10%
TM-200	10%
TM-240	10%
TM-380	10%
Z-207-7	10%
Z-304	10%
Z-315	10%
Z-316-8	10%
Z-405A	10%
Z-415	10%
Z-416-2	10%
Z-416C	10%
Z-417	10%
Z-445	10%
Z-449	10%
Z-505	10%
Z-515	10%
Z-516	10%
Z-525	10%
ZA-170-1	10%
ZA-170-3	10%
ZA-170-4	10%
ZA-170-6	10%
ZA-180-35	10%
ZA-180-39	20%
ZA-180-40	10%
ZA-180-45	10%
ZA-180-57	10%
ZA-180-62	10%
ZA-180-63	10%
ZA-180-65	10%
ZA-180-66	10%
ZA-180-67	10%
ZA-181-4	10%
ZA-181-7	10%
ZA-181-9	10%
ZA-181-17	10%
ZA-181-19	10%
ZA-181-20	10%
ZA-181-24	10%
ZA-3034-NP	20%
ZB-315-1	10%
ZBF-2526-EK	20%
ZBF-2527-EK	20%
ZBF-3339-EK	20%
ZBF-3340-EK	20%
ZBF-3339-GQ	20%
ZBF-3340-GQ	20%
ZBO-2503-EK	20%
ZCF-2326-EY	20%
ZCM-1390	20%
ZCM-1400-1	10%
ZCM-1490	20%
ZD-12	10%
ZD-14	10%
ZD-72	10%
ZD-200	10%
ZD-400	10%
ZD-800	10%
ZDF-1211-DY	20%
ZDF-1217-DY	20%
ZDF-2225-BK	20%
ZDF-2237-BK	20%
ZDF-2255-BK	20%
ZDH-1211-DE	20%
ZDH-1217-DE	20%
ZKB-2	20%
ZMM-149A	20%
ZMM-149P	20%
ZMM-1470G	20%
ZSS-184-1	20%
ZSW-184-2	20%
ZUS-386	10%
ZVM-135	20%
ZVM-1200-1	10%
ZVM-1240	20%
ZVM-1300-1	10%
ZVM-1330	20%
ZVM-1380C	20%
ZWL-200-2	20%
ZWL-200-4	20%

All software included in the software chart shown on pages 88 and 89 of catalog #213.

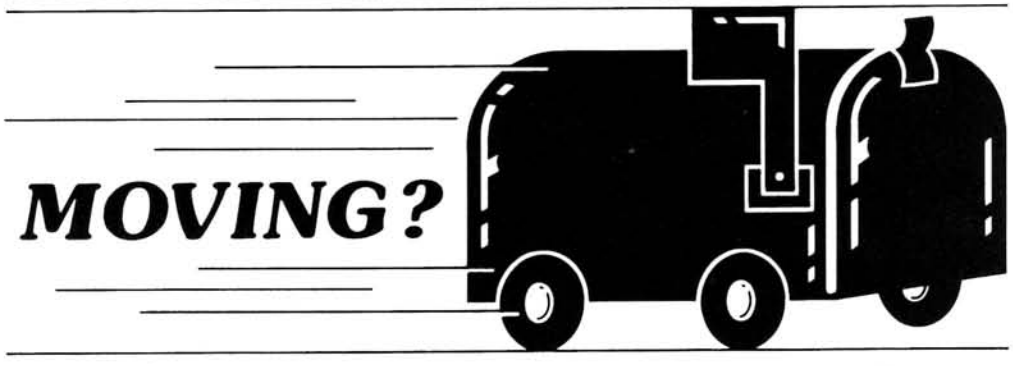
Continued from Page 13

Dante Bencivengo
P.O. Box 234
Wynandotte, MI 48192

David D. Clark
504 North Division Street
Bristol, IN 46507

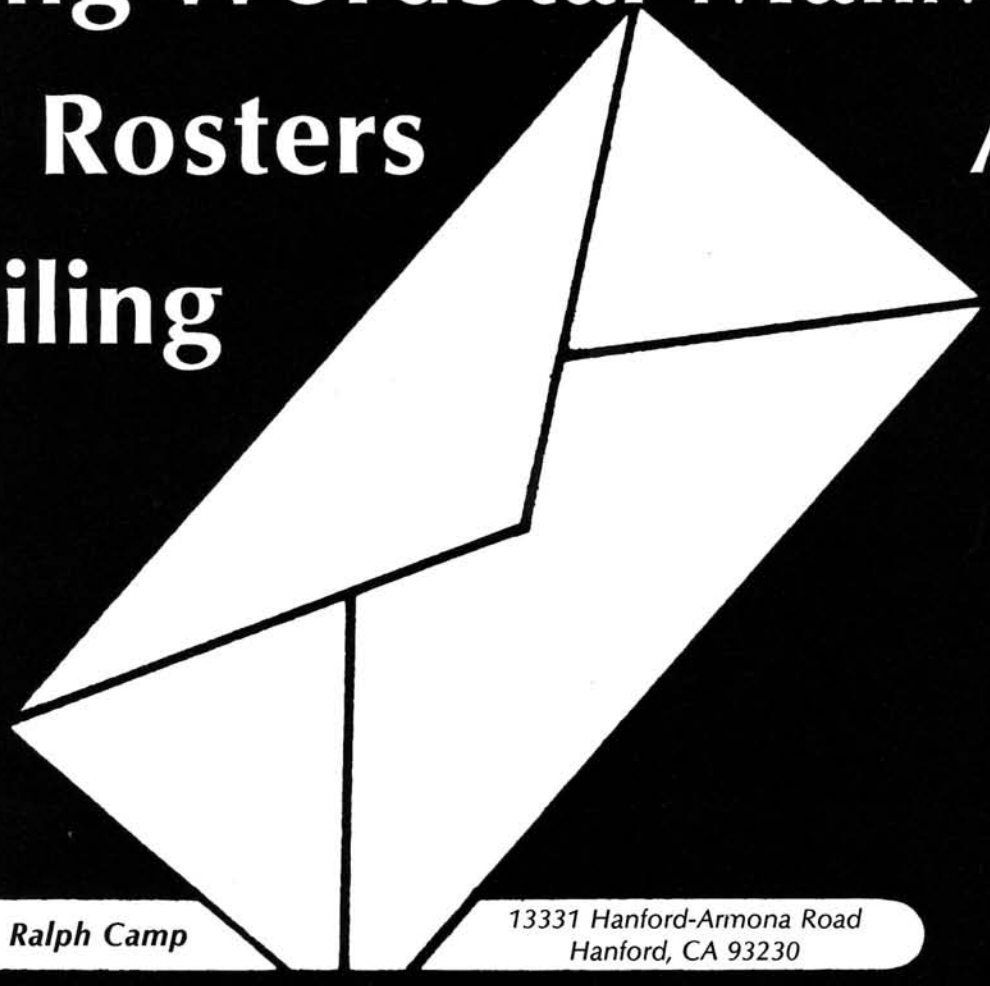
Sextant
716 E Street, S.E.
Washington, DC 20003
(205) 544-0900

JDR Microdevices
110 Knowles Drive
Los Gatos, CA 95030
(800) 538-5000 *



Please let us know 3-4 weeks in advance, so you won't miss a single issue of REMark!

Using WordStar MailMerge For Rosters And Mailing Lists



Ralph Camp

13331 Hanford-Armona Road
Hanford, CA 93230

I thought this computer would save me time and labor, not cause me more work! I can save more time and work by typing these few names and addresses manually each time I need them than to get the computer to do it for me!

If you feel this way, you are not alone. You may be frustrated, but don't bash the computer yet. There is a simple answer to the problem. The best part of all is that it won't cost you the price of another expensive software program. You will need to invest some time at the start, but it will be time well spent.

Here is an example. I have a list of churches and pastors that I need to mail information to regularly. But many of them have a mailing address different from their street address, which I also need to print out at a different time. Then I need to print out a master roster that has all of that information on the same sheet. Just to add a little spice, at times I need this information sorted alphabetically by church denomination, sometimes by church name, and sometimes by the pastor's name.

One last complication; I don't have a file program or a mailing list program that will do what I want to do with this information, and my sort program will not sort lines over 80 characters in length without eliminating everything past the 80th character. It seemed I was going to have to manually sort and type different parts of this information into the computer about six different times if I used the computer to print all of the different ways I need it printed.

However, I do have one positive factor in all of this. My sort program will sort "multi-line" entries. Also, the 80 column limitation has some advantages, as I will explain later.

The solution is simple to state: Type a data file using a multi-line format, using the key words I need to sort off of at the start of each line, and a series of print programs to produce the needed hard copy. That is the fancy way to say it. Translated, that statement says I should type the information into my computer in a non-document file with the information arranged the way I need it to be sorted and used, and then type different MailMerge Master Document files to look the way the different finished printed copies should look.

If you do not need your information sorted, you may want to ignore that section of the article. However, you may still want to use the multi-line format for your data file.

For my own thinking, I find it best to think of this as a SYSTEM for printing all of my information.

The very first step is to look closely at the information you have to use. Don't worry about how somebody else might be using this information. You are the one using it; you are the one who has to live with the outcome, so look at it to see what information is important to you.

Here is some of the information I have to use. For this article, I have changed names and addresses. Only my own name and address and the Lemoore, CA zip code are correct. The rest of the information is altered or completely fictitious. These examples are used to cover the different problems encountered in my own experience.

Not all of the "church names" are of congregations. Some of them are of church related organizations and the "pastor" has a different designation, so I found I needed to add another category of information, and that was "office held." I also needed a way to separate the churches from the other organizations.

The total information for each "entry" is called a RECORD. Each part of that record is called a FIELD.

You will need to determine how you will divide up your information. Then assign what the MailMerge manual calls a GENERIC NAME to each field of your record.

Figure 1
The Raw Information

St. Peter's Roman Catholic Church (mail to) P.O. Box 19 1001 D Street Lemoore, CA 93245	Priest: Father Simon Cephas church phone 924-1234 home phone 924-1234
Assembly of God 2992 Apple St Lemoore, CA 93245	Pastor: Rev. George Goodfellow church phone 924-2345 home phone 924-3456
Grace Baptist Church (mail to) P.O. Box 62 345 C Street Lemoore, CA 93245	Pastor: Pastor Mike Miracle church phone 924-4567 home phone 924- 5678
First Baptist Church 101 Salvation Ave., North Lemoore, CA 93245	Pastor: Dr. Sam Shepherd church phone 924-6789 home phone 924-7890
Church of the Nazarene (mail to) P.O. Box 3579 4321 The Way Road Lemoore, CA 93245	Pastor: (none at this time) church phone 924-8901 home phone
Lemorre Christian Servicemen's Center 13331 Hanford-Armona Road Hanford, CA 93230	Director: Rev. Ralph Camp office phone 584-0436 home phone 584-0436
Lemoore Christian Aid Society 1001 East West Avenue Lemoore, CA 93245	Director: Mrs. Jane Doe office phone 924-8765 home phone 924-4321

As I looked at the information I had and how I wanted to use it, I found that I needed these categories: church's denomination, church's name, church's mailing address if different from the street address, church's street address, church's phone number, pastor's name, pastor's title, and pastor's home phone number.

I found that for me, these worked best; SORTNAME, CHURCH, ADDMAIL, ADDSTREET, CITY, STATE, ZIP, OFFICEPH, HOMEPH, LASTNAME, FIRSTNAME, POSITION and TITLE. This generic name is the key to using your information in each of your MailMerge Master Documents. For now, look at your information and assign your generic names.

The MailMerge rules for Generic Names are: • Use one to forty letters in upper or lowercase and digits or hyphens. Do not use any other characters. • Do not use a digit or a hyphen as the first character. • Select generic names for each field in a record in a data file. (Page 3-11 in 3.3 manual.)

To determine what the most important information is, it is helpful, or maybe I should say, necessary, to draw out on a scratch pad what you want the printed copy to look like. If you expect to have several different forms of this information, you need to draw out each one of them. After all, the whole reason you are doing this is because there is a specific job to be done. What do you want to do, and what do you want it to look like?

By now you should have some idea of how many lines each record will use. You should also have some idea of how many different MailMerge Master Documents you will need to make your printer give you the information back on paper. You are ALMOST ready to start typing the information into your computer.

There is one more preparation you need to make before you actually start to type. That preparation is mental. Understand that you are doing this to meet your specific needs. Probably the hardest for me to remember is KISS; Keep It Simple, Sam! Use as many lines for each record as you need, but use as few as possible and still get the job done. Most of all, don't be afraid to experiment. Even though I have built these data files and MailMerge Master Document systems before, each one is unique, and each one takes some "fine tuning."

For my own thinking, I find it easier to refer to these MailMerge Master Documents as print programs, and that is how I will refer to them for the rest of this article.

My suggestion is to first type into your data file only five or six complete records of your information, and then use that file to test your print programs. This saves you a lot of typing if you find that you need to enter your information in a different arrangement. Then type the print program (MailMerge Master Document) you expect to use most. When that program is working OK, type your next print program and "fine tune" it. Keep going until all of your print programs work. After you have your system built and working, it is very

simple to add the rest of the information to your data file.

Your finished system will consist of your master data file, your sorted data files (if any), and your print programs (MailMerge Master Documents).

I decided that three lines for each record would do what I needed to do. The first line is SORTNAME and ADDSTREET (street address). I put the street address on the first line because it is normally longer than a post office mailing address. The second line is CHURCH, ADDMAIL (mail address, and not every record had one), CITY, STATE, ZIP, and OFFICEPH (office phone number). The third line is the person's LASTNAME, FIRSTNAME, POSITION, TITLE, AND HOMEPH (home phone number). This allows me to sort by the three needs I have, by denomination, by church name, or by pastor's name.

To separate the congregations from the church related organizations, I added "Z-" to the front of the sort name of the organizations. When sorted, these all appear alphabetically at the bottom of the list.

data file. I used "CHURCH.DAT".

Type your information as you have planned it on your scratch pad. If you do not have information for one field of your record, type a comma and go to the next field. You will see this in the illustration where there is not a pastor for the Nazarene church. You must have as many fields in your record as you have generic names in your print program (MailMerge Master Document). Do not put a comma after the last field of the line. Each line of type is to be ended with a carriage return. You will see "<" on the right side of your screen when you put a carriage return.

If you have to put a comma inside your text in the data file, as in the address "101 Salvation Avenue, North" in the illustration, you must enclose the text with quotation marks to make the computer print the comma as text and not read it as the end of the text for that field.

If you are going to ask your print program to determine if it is to print a specific field of information, you must have something in that field. I need to print the mailing address IF it is different from the street address. In those records where there was not a separate mailing address, it was necessary for me to enter "(none)" in that field. More about this when we get to the print programs.

As I suggested before, type only a few complete records of your data file for now. When you are finished, type CONTROL K D (^KD) to return to the OPENING MENU.

If you need to sort your information with a sort program, try it now with your short data file. I suggest you make a copy of your file and test it. That way if you do something wrong, you have not lost all of your work to this point. With your WordStar program at the OPENING MENU, type "O" to COPY a file. The screen will ask, "Name of file to copy from?" You type the name of your data file and RETURN. For our illustration, the name is CHURCH.DAT. The screen will then ask, "Name of file to copy to?" You type the name of your new file and RETURN. My suggestion is TEST.DAT.

Follow the instructions for your own sort program. When you are finished with your sort program, you have to make a choice. Either you have to delete your original file and rename your sorted file to the name of your original file, or you must have your print program read the sorted file, not the original file. This will be more meaningful when you are ready to run your first print program. I rename my sorted "master file" as .DAT and each of the other sorted files (resorted from the master file) as .SRT.

You may be able to run your sort file while still in WordStar. At least try it. If you have made a copy of your original file, the most that can happen is that you will electronically lock up your computer and have to reboot, and the most you have to lose is the copy of your file and a little time. With your WordStar program at the OPENING MENU, type "R" to "Run a program." The screen will ask, "COMMAND?" You type the name of your sort program and RETURN. If it works, when your sort program has finished sorting, you will see a command to "Hit any key to return to WordStar:"

With your WordStar program at the OPENING MENU, type "D" to open a DOCUMENT FILE. This will be your print program (MailMerge Master Document) to command the computer how to use the information in your data file. For each of these Master Documents, I use the file name extension of ".PGM" to denote that this file commands action as opposed to storing data or text. Since what I want to

Figure 2
The information as entered into the computer and sorted by SORTNAME.

(NOTE: The working non-document file is 80 characters wide. The indented lines in this illustration are the continuation of the preceeding line in the working file. Observe the carriage return character at the end of the lines.)

```
ASSEMBLYOFGOD,2992 Apple Street                                    <
Assembly of God,(none),Lemoore,CA,93245,924-2345                   <
Goodfellow,George,Pastor,Rev.,924-3456                             <
BAPTISTFIRST,101 Salvation Avenue, North                           <
First Baptist Church,(none)Lemoore,CA,93245,924-6789               <
Shepherd,Sam,Pastor,Dr.,924-7890                                   <
BAPTISTGRACE,345 C Street                                          <
Grace Baptist Church,P. O. Box 62,Lemoore,CA,93245,924-4567        <
Miracle,Mike,Pastor,Pastor,924-5678                                <
CATHOLIC,1001 D Street                                             <
St. Peter s Roman Catholic Church,P. O. Box 19,Lemoore,CA,93245,
     924-1234                                                      <
Cephas,Simon,Priest,Father,924-1234                                <
NAZARENE,4321 The Way Road                                         <
Church of the Nazarene,P. O. Box 3579,Lemoore,CA,93245,924-8901    <
,,Pastor,,                                                         <
Z-AIDSOCIETY,1001 East West Avenue                                 <
Lemoore Christian Aid Society,(none),Lemoore,CA,93245,924-5432     <
Doe,Jane,Director,Mrs.,924-6543                                    <
Z-SERVICEMENSCENTER,13331 Hanford-Armona Road                      <
Lemoore Christian Servicemen s Center,(none),Hanford,CA,93230,
     584-0436                                                      <
Camp,Ralph,Director,Rev.,584-0436                                  <
```

With your WordStar program at the OPENING MENU, type "N" to open a NON-DOCUMENT FILE. The name of this file should end with ".DAT" because it is a

do most with my information is to address labels and post cards, I will type and test these print programs before going on to the rest. My first program I called "ADDLABEL.PGM" for Address Labels Program. You decide what file name is best for you.

Figure 3
The Print Program to Print Address Labels

```
..This file is called ADDLABEL.PGM to print CHRUCH.DAT          <
..This is to print return address and church address labels     <
..Typed (date)                                                  <
.PL 12                                                          <
.MT 0                                                           <
.MB 0                                                           <
.PO 0                                                           <
.DF CHURCH.DAT                                                  M
.RV SORTNAME, ADDSTREET                                         M
.RV CHURCH, ADDMAIL, CITY, STATE, ZIP, OFFICEPH                 M
.RV LASTNAME, FIRSTNAME, POSITION, TITLE, HOMEPH                M
Lemoore Ministerial Association                                 <
P. O. Box 646                                                   <
Lemoore, CA  93245                                              <
                                                                <
                                                                <
                                                                <
&TITLE& &FIRSTNAME& &LASTNAME&                                  <
&CHURCH&                                                        <
IF &ADDMAIL& = "(none)" GOTO STREET                             M
&ADDMAIL&                                                       <
.EF STREET                                                      M
.EX &ADDMAIL& = "(none)" GOTO CITY                              M
&ADDSTREET&                                                     <
.EF CITY                                                        M
&CITY&, &STATE&,   &ZIP&                                        <
.PA                                                             <
----------------------------------------------------------------P
```

I start each of my program files with non-printing comments to record the file name, what the program is to do, and the date it is typed. I will explain the purpose of this in detail later. Non-printing comments are entered by typing a period or "dot" in the first two columns of the line, then your comment.

The first four program lines of your file after the non-printing comment lines will be DOT COMMANDS to override or superceed the page formatting established with WordStar Install. They are PAGE LENGTH (.PL), MARGIN TOP (.MT), MARGIN BOTTOM (.MB), and PAGE OFFSET (.PO). Most continuous feed address labels are one inch from top of label to top of label, or six lines of type. Since I am making a return address label for each address I am printing, I use .PL 12. If you are printing addresses without a return label, use .PL 6. PAGE LENGTH should be the very first dot command, on the first line after any non-printing comments. (You get a "nasty-gram" on your screen from WordStar if it is not.) Next, MARGIN TOP and MARGIN BOTTOM are each set to zero. Otherwise, you would not be able to print, because WordStar normally has a total of 11 lines of top and bottom margins on a printed page. PAGE OFFSET can be the hardest one to deal with. This sets the position where the printer will start printing on your label. I find it is best for me to use zero (.PO 0) and adjust the paper feed on my printer to match the column one print position, but you may find it better to use a different print start position to match your printer. This is where the "fine tuning" comes in.

Now it is time to add the MailMerge dot commands. DEFINE FILE (.DF) tells the computer where to look for your information to READ the VARIABLES (.RV).

There are three .DF rules that must be followed to make your program work. • Always put .DF before .RV, and both of them above text. • Do not use more than one .DF in a master document. • Use the file name in your .DF command exactly as it is used to name your data file on the disk menu.

There are five .RV rules that must be followed to make your program work. • Always provide all of the generic names in the Master Document for all the fields in a data file even though you may not want to use all of them. • Never vary the order of the generic names in the .RV command from the order of the fields in the data file. • An .RV command may be no longer than 240 characters. • Never use ampersands (&) around the generic names in an .RV command. • There must be a SPACE between the .RV command and the first generic name. After that, generic names may be separated by only a comma or by a comma and a space.

Whenever a MailMerge dot command is used, an "M" will appear on the right side of the screen where the carriage return "<" character normally appears, even though you end every dot command with a carriage return.

If you look at the ruler line at the top of the screen, the line indicator will still read "LINE 1" if you have typed everything correctly to this point. In the illustration of Figure 3, 11 typed lines of non-printing comments and dot commands preceed the first line of text.

You are now ready to enter your first line of text. If you are printing a return address label, enter your return address as you want it to be printed, and then add enough carriage returns to move the cursor to line seven. This will be the first line of type using the information from your data file. If you are not printing a return label, the first line of type will be on line one.

Type the generic name for the information you want on the first line of the address, in the order that you want it to be printed. All generic names are enclosed with an ampersand (&). The first line of my address label contains the pastor's title and name — first and last. If there is not a pastor, as in the case of the Nazarene church in the illustration, the computer will leave this line blank, but advance the printer one line and print the second line of information on the second line of type, the same as if line one had been printed. The second line is the church's name.

The third printed line of the address is a little more difficult. I need the computer to decide between two different addresses and print the proper one. Six program lines are used to get the one correct address printed on the label. Here is where CONDITIONAL PRINTING is used. If you do not need a choice made for your program, you may want to move to the last line.

I want to use the mailing address (ADDMAIL) IF there is one. The dot command IF (.IF) tells the computer to print the mail-

ing address if there is one and to go to the street address if there is not a mailing address. To be more technically precise, the command tells the computer to go to the street address if there is not a mailing address, otherwise, print the mailing address. The command is written this way: .IF &ADDMAIL& = "(none)" GOTO.

It is necessary to have text in the field being questioned to make the program work correctly all of the time. That is why it was necessary to put "(none)" in the place of a mailing address in the data file if there was not one. For those who are interested, there is a technical note at the end concerning empty fields and IF commands.

There is no need to add anything to the GOTO command, because MailMerge already understands to GOTO the END IF (.EF) command. Every IF (.IF) and EXCEPT (.EX) command must have an END command. However, in this case, there is a need for a second END IF command in the same program. Therefore, it is helpful to add an OPTIONAL LABEL to the GOTO and END IF commands to help you follow the program in your own thinking. Whatever optional label is used with the GOTO should also be used with the END IF command. Be sure to put a space between the command statement and the optional label. Here I used GOTO STREET.

The generic name &ADDMAIL& is placed between the .IF dot command and the .EF dot command, each one on its own program line. Notice again that the line count does not advance and an "M" appears on the right side of the screen when you use these dot commands.

The first half of the conditional printing is set. Now for the second half. Again, three statements are used. However, this time I want to bypass the street address EXCEPT when there is not a separate mailing address. The command is written this way: .EX &ADDMAIL& = "(none)" GOTO. A "rule of thumb" for using IF and EXCEPT: whenever possible, use the one that will allow you to use the equal sign (=) and avoid using the unequal sign (<>). The command can be written .IF ADDMAIL& <> "none" GOTO and it will work, but the better way in this case is to use .EX and =.

The generic name *ADDSTREET& is placed between the .IF dot command the .EF dot command, each one on its own program line. The conditional printing section is finished.

The last line is simple. &CITY& is followed by a comma and a space, &STATE& is followed by two or three spaces, and &ZIP&, and the address is complete.

One last command to go. Every MailMerge Master Document should end with a PAGE (.PA) command. This is vital to a print program like this one. After you enter .PA and a carriage return, a "P" will appear on the right side of the screen and a row of dashes will go across the screen. The ruler line at the top of the screen will now show PAGE 2, LINE 1, COL 01. Type CONTROL K D (^KD) and return to the OPENING MENU.

Now it is test time. Turn your printer "ON" if it is not already. With your WordStar program at the OPENING MENU, type "M" to Run MailMerge. The screen will ask, "Name of file to MailMerge?" You type the name of your print program. You can escape the rest of the print questions by pressing the ESCAPE key instead of the RETURN key.

If all went well, you have your addresses printed the way you want them. If not, go back and check each step of your work. Hopefully, the most you might have to do is adjust the PAGE OFFSET (.PA). You should have this program working the way you want it to before you go on to the next programs.

I said early in this article that being limited to 80 columns per line had advantages as well as disadvantages. One big advantage is that you can have your system type out your data file on an 80 columm printer so you can look for mistakes in your typing or arrangement of information. Since I use CP/M, I will have to give you the CP/M commands, but I am assured by MS-DOS friends that the commands work "about the same."

With your printer turned "ON" and your computer in the command mode, at the "A>" prompt, type CONTROL P (^P) and RETURN to set the computer to send information to the printer, as well as to the screen. Next, type the command TYPE and the FILENAME and RETURN. In the case of the illustration we have been using, it will look like this: A> TYPE CHURCH.DAT followed by a RETURN. You should see the information printed on both your screen and your printer. If the lines in your data file are longer than 80 characters, this will not work properly unless you have a wide carriage printer. When the printer has finished printing, enter a second ^P to turn the print command to "OFF" or everything you type on the keyboard will also go to the printer.

If you have a DESPOOL program, you can use it to print out your data file. Follow the instructions for your specific program.

When your print program is working the way you want it to, use this same method to print a HARD COPY (paper copy) of your program. Keep a file folder or notebook of all of your print programs. There are two reasons for this. The first is that as you develop your use of the computer, you will use a lot of this information in different ways. Sometimes all you have to do is to adjust one program by a few lines and have it do another job for you. The second reason to keep a hard copy file is in the event of a computer disaster (which I have had), you do not have to reconstruct all of your work. Sure it is a lot of typing to enter all of the information and programs again, but at least you do not have to do all of the research again. When you have non-printing comments at the top of your print program, it helps you understand and remember which program does what job with what data file. I even keep a second file folder at another location. One disaster is enough for me.

A variation of the address label program is a program to address continuous feed post cards, if you are interested. With your WordStar program at the OPENING MENU, type "O" to COPY a file. The screen will ask, "Name of file to copy from?" You type the name of your print label program and RETURN. For our illustration, the name is ADDLABEL.PGM. The screen will then ask, "Name of file to copy to?" You type the name of your new program and RETURN. For our illustration, the name is ADDCARD.PGM to address post cards. When the computer is finished copying your program, type "D" to open the document file and type the name you have called your new program.

Look at Figure 4. The first two comment lines have been changed to reflect the new program. You will also see that PAGE LENGTH (.PL) is now 24 because the post cards are 4 inches or 24 lines high. The only other difference is that there is more space between the return address and the address, and the address has been moved more to the center of the postcard. In this case, there are six blank lines between the

Figure 4
The print program to address post cards.

```
..This file is called ADDCARD.PGM to print CHURCH.DAT          <
..This is to address postcards to pastors                      <
..Typed (date)                                                 <
.PL 24                                                         <
.MT Ø                                                          <
.MB Ø                                                          <
.PO Ø                                                          <
.DF CHURCH.DAT                                                 M
.RV SORTNAME, ADDSTREET                                        M
.RV CHURCH, ADDMAIL, CITY, STATE, ZIP, OFFICEPH                M
.RV LASTNAME, FIRSTNAME, POSITION, TITLE, HOMEPH               M
Lemoore Ministerial Association                                <
P. O. Box 646                                                  <
Lemoore, CA  93245                                             <
                                                               <
                                                               <
                                                               <
                                                               <
                                                               <
                                                               <
                                                               <
                &TITLE& &FIRSTNAME& &LASTNAME&                 <
                &CHURCH&                                       <
.IF &ADDMAIL& = "(none)" GOTO STREET                           M
                &ADDMAIL&                                      <
.EF STREET                                                     M
.EX &ADDMAIL& = "(none)" GOTO CITY                             M
                &ADDSTREET&                                    <
.EF CITY                                                       M
                &CITY&, &STATE&   &ZIP&                        <
.PA                                                            <
---------------------------------------------------------------P
```

return address and the address, and the generic names have been moved over 15 spaces. Be sure to leave the dot commands in column one, or they will not work. Now you have two print programs. Give this one a try on your computer. Test it the same way you did the first program.

There are three more programs that are helpful when printing rosters. Again, the second program is a variation of the first.

(The remainder of the program prints a roster by church and a second roster by pastor, and a program for file cards.)

Technical note concerning empty fields and IF commands. It is possible to make the program work under certain conditions by using only four program lines and an empty field. However, the program will not work under all conditions. If you want to experiment, the four lines are 1) .EX &ADDMAIL& = "" GOTO, 2) &ADDSTREET&, 3) .EF and 4) &ADDMAIL/O&.

*

ENABLE Part 9

A Tutorial

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

In this, the ninth in a series on ENABLE, we return to the spreadsheet. In the previous two spreadsheet articles, the basic capabilities have been discussed. Basic graphics capability will be explored in this article. This capability was of great use to me during the development of a business plan last year. I started this plan using LOTUS 1-2-3 but because of the amount of graphics involved, changed to ENABLE when it became available. This provided the means to complete the plan with less effort. Graphics is one of the best features in the ENABLE spreadsheet module.

While I continue to focus on the Z-100 version of ENABLE, the PC version 2.0 has an additional graphics package called Perspective. This is an add on program written by 3D Graphics and requires EGA or greater capability so it will not run on the Z-100. If you select the SIGMA 400 graphics board during installation, some limited screen displays are possible on the Z-100 using Perspective, ZPC and the PC version of ENABLE. You can not run the Z-100 version of ENABLE using ZPC as it checks for an IBM interrupt and if it finds it, it will not load. The Perspective printer outputs, not the screen dumps, will provide a usable product, but you need to know the function keys that will provide this output. I will go into Perspective in greater detail in a later article.

I will focus on the Z-100 version of ENABLE but will continue to provide the PC keystrokes to accomplish the same task. ENABLE Version 2.15 should be available when you read this. It has been developed as a follow on product for the Department of Defense Z-248 contract. It provides added capabilities over version 2.0 and many improvements over version 1.15 which is currently on the Z-248 requirements contract. These improvements should carry through to version 3.0 which, according to the media, should be released this summer.

ENABLE's spreadsheet capability in total cells is somewhat limited when compared to products like LOTUS 1-2-3. Because of the integration requirement of the product, limits had to be set, which resulted in the 65,000 total cell limit. For most users, this should provide more than enough to complete the task at hand. Larger applications, if studied closely, are probably database applications.

Before moving into the basic graphics capability of ENABLE, a little more discussion is in order on the printing capability of ENABLE's spreadsheet. Unlike LOTUS, print macros can not be added to the spreadsheet for printing of groups of parts or to change the character per inch (CPI). ENABLE uses the basic print capability of the integrated package to provide printing. In one large application I developed, 16 different parts of the spreadsheet had to be printed with a change of CPI between some parts. In order to get ENABLE to accomplish this task, a macro had to be written. Because this requirement occurred soon after I received the package, I did not feel up to this requirement. To make it easy, I would write the ENABLE spreadsheet out to LOTUS format and print it from there using a macro I had written. I had been using LOTUS on my Z-100 for several years at that time and this was easier to do. This showed the ability of ENABLE to write out and read in LOTUS files without problem.

Using ENABLE's normal spreadsheet print capability, the above problem could have been solved by printing the bulk of the report in the usual way. This involves selecting the fields and then printing them in 12 CPI. The last part of the report was printed in 17 CPI. This part of the spreadsheet is selected and the printer command changed to 17 CPI. See Figure 1 for a graphics presentation of the printer requirement. Because this report is generated using several different spreadsheets, a macro could be used. ENABLE has two ways to develop macros. The first way is to type in the commands to be used by using the main menu macro commands.

This method will be discussed in a later article. The easiest method is to "teach" the computer what you want it to do. For this example, the macro will be called "P". In order to teach the computer, you must perform the tasks to be accomplished which are then recorded. Having the spreadsheet created, start the "learning" by pressing "F0 F9" (ALT/F9 for the PC) to start the macro procedure. Press the backslash (\) to tell ENABLE you want to record the keystrokes. Now press the "P" key to name the macro. Any key, except for shifted, ALT or CTRL keys or the backslash and minus sign can be used as macro names. ENABLE will display "MAC" on the status line to indicate that a macro procedure is in progress. ENABLE is now ready to learn what to do when the macro "P" is selected. Any key that you press from this point until you turn off the "learn" will be record and can be called up at any time. For this example we will press "F10 (P)rint" for the print menu in the spreadsheet. Then select (R)ange and type in the first range, i.e., aa3..am43 and then a <RETURN>. Next press (S)etup, and assuming that the first three screens are correct, press SHIFT/3 (PgDn for the PC) three times. The print screen that sets the CPI is now displayed. Press the down arrow key twice and (E)lite, one more down arrow and the F0 F10 (ALT/F10 for the PC). The last set is to press (E)xecute (P)rinter. This page will then print. When it is completed, press F10 (P)rint (R)ange and select the next range to be printed in the spreadsheet. Press (E)xecute (P)rinter and this page will print. Continue this until all pages have been printed. Don't worry about mistakes that are made during this "learn" procedure as they will be repeated but will cause no more problems than when they were input. When you have completed this procedure, press "F0 F9 SHIFT/1" (ALT/F9 END for the PC) and the entire procedure will be saved under the name "9.SSM". Macros that are developed in LOTUS are saved in a cell(s) and are saved with the spreadsheet. You have to reinput or copy them to other spreadsheets in order to use them. To use this macro again, press "F0 F9 P" (ALT/F9 P for the PC) and the entire procedure will run. This is the first time that macros have been discussed in this series on ENABLE. I will expand on these capabilities in upcoming articles.

The basic graphics capability of ENABLE is one of its strongest points. As a long time user of LOTUS 1-2-3, the added capabilities were a welcome resource. In the first example I used above, printing of the financial data from a business plan, extensive graphing was also required. LOTUS graphing is done within the spreadsheet but the actual printing is done from another module in the program. You must build the graph, including labels and then save this to a file that will be printed. If you have another graph, you are required to delete the graph setting and build the new graph. You must then save this graph. When you are finished with all of the graphs, you save the spreadsheet and the last graph setting and go to the "Printgraph" part of the program. You can then print all of the graphs that have been saved using the fonts that are selected during this procedure. You can select all graphs and they will print one after another. This provides good graphs of the material that was developed from the spreadsheet. Unfortunately, if you need to change the LOTUS spreadsheet, all of the graphs will have to be rebuilt, saved and reprinted. The version of LOTUS I am most familiar with is 1.1 for the Z-100 so this is what I base my comparison on.

The same exercise using ENABLE will be several steps easier to start and many steps easier for the change that always occurs in spreadsheets. ENABLE graphs are printed from the spreadsheet as one of the top line menu options. This is one less step than LOTUS. With ENABLE, you can have different fonts for different labels. When you display the graph from the spreadsheet, these fonts will be displayed on the screen so you can immediately see what the graph will look like. More graph types are available in ENABLE than in LOTUS and using PC version with Perspective even more. The feature that provides the greatest time savings is that ENABLE will save up to eight different graph settings with the spreadsheet. So if a change is necessary, all graphs will be changed to reflect these changes and the rebuilding of all graphs will not be necessary. This one capability saved many hours when the spreadsheets for the business plan were changed. Another graphic capability was discussed in article six. This capability permits both the spreadsheet and the graph to exist on the same screen. With this interactive capability, changes on the spreadsheet can be displayed graphically on the screen, providing immediate feedback on the change. The ENABLE graphing capability is not limited to spreadsheets but will also extend to the database function.

To use the graphing capability of ENABLE, you must be in the spreadsheet with data available to graph. From the opening

A ... AJ

10

Freight Month One 17 CPI	Dinner Train Month One 17 CPI	Excursion Month One 17 CPI	Summary Month One 12 CPI ** PRINT **
Freight Month Two	Dinner Train Month Two	Excursion Month Two	Summary Month Two 12 CPI ** PRINT **
↓	↓	↓	↓ ** PRINT **
Freight Month Twelve	Dinner Train Month Twelve	Excursion Month Twelve	Summary Month Twelve ** PRINT **
Freight Summary Year One ** PRINT **	Dinner Train Summary Year One ** PRINT **	Excursion Summary Year One ** PRINT **	Summary Year One ** PRINT **

1100

FIGURE 1 - Spreadsheet printer requirement

menu select (U)se system (S)preadsheet/ graphics (R)evise for this graphics demonstration. I will use some of the data from the business plan to provide data which will be presented. Some of the data will be used in article eleven where the graphs will be moved into the textural part of the business plan. The name for the spreadsheet is entered as "YEARONE". The real spreadsheet for this year provides a month by month projection of sales/services which are used to secure loans and other financial backing. The 13th block of data is the summary, and this is the one we will use.

After the data has been created and the summary page has been developed, the graphs will be created to graphically present the data. The graphs will provide a quick look at the data and will point out errors or areas that need further research or review. From the spreadsheet, three profit/loss centers are identified: freight operations, dinner train, and passenger excursion. The first chart to be made will be a pie chart showing the percentage of income from each to the centers. Press "F10" or backslash (\) to display the top line menu. Press (G)raph to enter the graphics package. The next menu permits you to (S)elect (C)reate or (D)elete a graph. If you press <RETURN> at this time, the only graph name will be DEFAULT. There is no data associated with this graph, only a name. For this first chart press (C)reate and name the chart on the next screen where prompted "INCOMPIE". The next screen prompts for more information on the graph by selecting (O)ptions. This is the screen where you can (D)isplay the completed graph on the screen, (P)rint the graph to a printer, or (L) output to a plotter. On the PC version, another selection (E), gives access to PERSPECTIVE. Also note in the upper right corner of the screen is a display "X12345678". This is the total number of graphs that can be saved in ENABLE. The number of graphs saved will be displayed by a line under the number starting under the "1". At this time the display will reflect "X12345678" as two graph names are available, "DEFAULT" and "INCOMPIE". If you select (D)isplay, nothing will be displayed as no data has been selected.

Select (O)ptions to permit data points to be inputted into the graph. The next menu is where you select the data points. This first chart is to reflect the income of the company with each profit/loss center being a segment. For the first point, press

	A	B	C	D	E	F	G	H	I	J	K
1	r		150	11	10		120	38		75	6
2											
3		Frt	Frt	Trkage	DT	Dinner	Dinner	Dinner	Excursion	Excusion	Excursion
4		Car	Income	Right	Movement	Train	Train	Train	Train	Train	Train
5		Loading		Income		Trip	PAX	Income	Trips	PAX	Income
6											
7	Jan	35	5250	770	16800	14	1680	63840	4	300	1800
8	Feb	25	3750	1320	14400	12	1440	54720	2	150	900
9	Mar	30	4500	660	19200	16	1920	72960	2	150	900
10	Apr	15	2250	220	30000	25	3000	114000	4	300	1800
11	May	12	1800	220	36000	30	3600	136800	12	900	5400
12	Jun	15	2250	275	60000	50	6000	228000	24	1800	10800
13	Jul	9	1350	385	68400	57	6840	259920	24	1800	10800
14	Aug	32	4800	550	74400	62	7440	282720	24	1800	10800
15	Sep	26	3900	880	72000	60	7200	273600	24	1800	10800
16	Oct	47	7050	1045	54000	45	5400	205200	6	450	2700
17	Nov	38	5700	1320	36000	30	3600	136800	10	750	4500
18	Dec	42	6300	1210	36000	30	3600	136800	10	750	4500
19											
20	Total	326	48900	8855	517200	431	51720	1965360	146	10950	65700
21											
22	Freight		574955								
23	DT		1965360								
24	Excursion		65700								
25	Total		2606015								
26											
27			Freight Expenses				DT Expenses				
28											
29											
30		Loan	Insur	Tax	Labor	MX/Supp	Food	Labor	Tax	Insur	RR Move
31	Jan	17000	4000	1500	15000	2000	20160	16128	2000	8500	16800
32	Feb	17000	4000	1500	15000	2000	17280	13824	2000	8500	14400
33	Mar	17000	4000	1500	15000	2000	23040	18432	2000	8500	19200
34	Apr	17000	4000	1500	15000	2000	36000	28800	2000	8500	30000
35	May	17000	4000	1500	15000	2000	43200	34560	2000	8500	36000
36	Jun	17000	4000	1500	15000	2000	72000	57600	2000	8500	60000
37	Jul	17000	4000	1500	15000	2000	82080	65664	2000	8500	68400
38	Aug	17000	4000	1500	15000	2000	89280	71424	2000	8500	74400
39	Sep	17000	4000	1500	15000	2000	86400	69120	2000	8500	72000
40	Oct	17000	4000	1500	15000	2000	64800	51840	2000	8500	54000
41	Nov	17000	4000	1500	15000	2000	43200	34560	2000	8500	36000
42	Dec	17000	4000	1500	15000	2000	43200	34560	2000	8500	36000
43											
44	Total	204000	48000	18000	180000	24000	620640	496512	24000	102000	517200
45											
46	Total	474000					Total	1760352			
47											
48	Freight	474000									
49	Dinner Tr	1760352									
50	Excursion	64136									
51	Total	2298488									

	Insur	Labor	Supp
Jan	1500	864	400
Feb	1500	432	200
Mar	1500	432	200
Apr	1500	864	400
May	1500	2592	1200
Jun	1500	5184	2400
Jul	1500	5184	2400
Aug	1500	5184	2400
Sep	1500	5184	2400
Oct	1500	1296	600
Nov	1500	2160	1000
Dec	1500	2160	1000
Total	18000	31536	14600
		64136	

FIGURE 2 - YEARONE Spreadsheet - Summary page

```
A1: r                                                          MENU
Worksheet   MCM   Print   Graph   Save   Combine   DBMS   Quit
Display range(s) of cell values in a graph
     A          B          C          D          E          F          G          H
  1  r                     150         11         10                    120         38
  2
  3              Frt        Frt      Trkage      DT      Dinner     Dinner     Dinner
  4              Car        Income   Right     Movement  Train      Train      Train
  5           Loading                Income             Trip       PAX        Income
  6
  7  Jan          35        5250       770      16800       14       1680       63840
  8  Feb          25        3750      1320      14400       12       1440       54720
  9  Mar          30        4500       660      19200       16       1920       72960
 10  Apr          15        2250       220      30000       25       3000      114000
 11  May          12        1800       220      36000       30       3600      136800
 12  Jun          15        2250       275      60000       50       6000      228000
 13  Jul           9        1350       385      68400       57       6840      259920
 14  Aug          32        4800       550      74400       62       7440      282720
 15  Sep          26        3900       880      72000       60       7200      273600
 16  Oct          47        7050      1045      54000       45       5400      205200
 17  Nov          38        5700      1320      36000       30       3600      136800
 18  Dec          42        6300      1210      36000       30       3600      136800
 19
 20  Total       326       48900      8855     517200      431      51720     1965360
#2   C:\EN200\YEARONE.SSF                                                     Cap   N51
```

FIGURE 3 - Spreadsheet Top Line Menu

```
Create
 Enter name of graph setting you wish to CREATE:
     A          B          C          D          E          F          G          H
  1  r                     150         11         10                    120         38
  2
  3              Frt        Frt      Trkage      DT      Dinner     Dinner     Dinner
  4              Car        Income   Right     Movement  Train      Train      Train
  5           Loading                Income             Trip       PAX        Income
  6
  7  Jan          35        5250       770      16800       14       1680       63840
  8  Feb          25        3750      1320      14400       12       1440       54720
  9  Mar          30        4500       660      19200       16       1920       72960
 10  Apr          15        2250       220      30000       25       3000      114000
 11  May          12        1800       220      36000       30       3600      136800
 12  Jun          15        2250       275      60000       50       6000      228000
 13  Jul           9        1350       385      68400       57       6840      259920
 14  Aug          32        4800       550      74400       62       7440      282720
 15  Sep          26        3900       880      72000       60       7200      273600
 16  Oct          47        7050      1045      54000       45       5400      205200
 17  Nov          38        5700      1320      36000       30       3600      136800
 18  Dec          42        6300      1210      36000       30       3600      136800
 19
 20  Total       326       48900      8855     517200      431      51720     1965360
#2   C:\EN200\YEARONE.SSF                                                     Cap   N51
```

FIGURE 4 - Graph Create Menu

```
Select DEFAULT                                                 X12345678
Options  Display  Print  L=Plot  E=Perspective(tm)
Change current graph settings
     A          B          C          D          E          F          G          H
  1  r                     150         11         10                    120         38
  2
  3              Frt        Frt      Trkage      DT      Dinner     Dinner     Dinner
  4              Car        Income   Right     Movement  Train      Train      Train
  5           Loading                Income             Trip       PAX        Income
  6
  7  Jan          35        5250       770      16800       14       1680       63840
  8  Feb          25        3750      1320      14400       12       1440       54720
  9  Mar          30        4500       660      19200       16       1920       72960
 10  Apr          15        2250       220      30000       25       3000      114000
 11  May          12        1800       220      36000       30       3600      136800
 12  Jun          15        2250       275      60000       50       6000      228000
 13  Jul           9        1350       385      68400       57       6840      259920
 14  Aug          32        4800       550      74400       62       7440      282720
 15  Sep          26        3900       880      72000       60       7200      273600
 16  Oct          47        7050      1045      54000       45       5400      205200
 17  Nov          38        5700      1320      36000       30       3600      136800
 18  Dec          42        6300      1210      36000       30       3600      136800
 19
 20  Total       326       48900      8855     517200      431      51720     1965360
#2   C:\EN200\YEARONE.SSF                                                     Cap   N51
```

FIGURE 5 - Graphics Task Menu

(1) or move the highlight bar using the cursor keys to the "1" in the data group and press <RETURN>. The Create menu is then presented. Note that ENABLE will provide a line in the top line menu to let you know what graph you are working on and at what level. The first item to be selected is the data for the graph. Press (D)ata and then another <RETURN> to point to the data range. For this example, select field "C22" either by typing in the field or by moving the highlighted cell to "C22" and pressing <RETURN>. If you have a range of cells, you point to the first cell insert ".." and then move to the last cell in the range and press <RETURN>. ENABLE will highlight all cells in the range. Again for this example, type (L)egend and for the ENABLE prompt type in "Freight" and a <RETURN>. Because this will be a pie chart, we will use another capability of ENABLE which is exploded segments. To select this option, press (E)xplode and respond (Y)es to the prompt. If you want color for the display, to plot the result, or have a color printer, you can select the color for this segment by pressing (C)olor and then selecting the color of your choice. When you have completed these tasks, press {ESC} and you will be returned to the Create Menu. Press (2) for the next data point. The second data point will be the dinner train income located in cell "H20" with the legend "Dinner Train"; again select Exploded. Note that in the top right hand corner, ENABLE will provide the total number of points selected for the graph with the number displayed in the manner as the graph number. For the second point, the display will be "X12345678". Press {ESC} when completed with data point "2", returning to the Create Menu. Select "3" for the last point in this graph. This data point resides in cell "L20" and the legend should be "Excursion" and exploded should again be selected. This completes the data point selection for the pie chart.

After pressing {ESC} you will return to the Create Menu again. Press (G)lobal for the next level of menus. The first selection is the type of graph. Press <RETURN> as (T)ype is the default setting and then select (P)ie as the type. After selecting the pie chart, another menu will appear in which you will select the Type of pie chart. Press <RETURN> and you will be prompted if you want exploded, to which you respond (Y)es. The next option is the Format. This menu item will permit you to select how you want the data labels displayed and where to find them. (D)ata labels will require that the labels be typed in on a line and then "X-Axis Data" be selected in another option. Since we have input legends with the data points, this will be selected. The last option permits you to select the shading or colors of the segments of the graph. Again, we have selected this option with the data points so it is not required. You can now move back through the menus to display the graph by pressing {ESC} four times until the (D)isplay options are shown and then se-

```
Select DEFAULT Options                                          X12345678
Global        Data Group:  1  2  3  4  5  6  7  8      Device
Change options that apply to entire graph
      A          B          C          D          E          F          G          H
  1   r                     150         11         10                    120         38
  2
  3              Frt        Frt      Trkage       DT      Dinner     Dinner     Dinner
  4              Car     Income       Right  Movement      Train      Train      Train
  5          Loading             Income                   Trip        PAX     Income
  6
  7   Jan         35       5250        770      16800         14       1680      63840
  8   Feb         25       3750       1320      14400         12       1440      54720
  9   Mar         30       4500        660      19200         16       1920      72960
 10   Apr         15       2250        220      30000         25       3000     114000
 11   May         12       1800        220      36000         30       3600     136800
 12   Jun         15       2250        275      60000         50       6000     228000
 13   Jul          9       1350        385      68400         57       6840     259920
 14   Aug         32       4800        550      74400         62       7440     282720
 15   Sep         26       3900        880      72000         60       7200     273600
 16   Oct         47       7050       1045      54000         45       5400     205200
 17   Nov         38       5700       1320      36000         30       3600     136800
 18   Dec         42       6300       1210      36000         30       3600     136800
 19
 20   Total      326      48900       8855     517200        431      51720    1965360
#2    C:\EN200\YEARONE.SSF                                              Cap        N51
```

FIGURE 6 - Create Options Menu Display

```
Select DEFAULT Options Group                                    X12345678
Data  Scale  Color  1=Shade  Legend  Reset-data  Explode  2=Symbol
Specify Spreadsheet range or DBMS field for this group
      A          B          C          D          E          F          G          H
  1   r                     150         11         10                    120         38
  2
  3              Frt        Frt      Trkage       DT      Dinner     Dinner     Dinner
  4              Car     Income       Right  Movement      Train      Train      Train
  5          Loading             Income                   Trip        PAX     Income
  6
  7   Jan         35       5250        770      16800         14       1680      63840
  8   Feb         25       3750       1320      14400         12       1440      54720
  9   Mar         30       4500        660      19200         16       1920      72960
 10   Apr         15       2250        220      30000         25       3000     114000
 11   May         12       1800        220      36000         30       3600     136800
 12   Jun         15       2250        275      60000         50       6000     228000
 13   Jul          9       1350        385      68400         57       6840     259920
 14   Aug         32       4800        550      74400         62       7440     282720
 15   Sep         26       3900        880      72000         60       7200     273600
 16   Oct         47       7050       1045      54000         45       5400     205200
 17   Nov         38       5700       1320      36000         30       3600     136800
 18   Dec         42       6300       1210      36000         30       3600     136800
 19
 20   Total      326      48900       8855     517200        431      51720    1965360
#2    C:\EN200\YEARONE.SSF                                              Cap        N51
```

FIGURE 7 - Create Options Group Data Menu Display

```
Select DEFAULT Options Global Type                              X12345678
Vertical-bar  Pie  Line  XY  HLC
3D-Standard  2D-Standard  S=3D-Stacked  T=2D-Stacked
      A          B          C          D          E          F          G          H
  1   r                     150         11         10                    120         38
  2
  3              Frt        Frt      Trkage       DT      Dinner     Dinner     Dinner
  4              Car     Income       Right  Movement      Train      Train      Train
  5          Loading             Income                   Trip        PAX     Income
  6
  7   Jan         35       5250        770      16800         14       1680      63840
  8   Feb         25       3750       1320      14400         12       1440      54720
  9   Mar         30       4500        660      19200         16       1920      72960
 10   Apr         15       2250        220      30000         25       3000     114000
 11   May         12       1800        220      36000         30       3600     136800
 12   Jun         15       2250        275      60000         50       6000     228000
 13   Jul          9       1350        385      68400         57       6840     259920
 14   Aug         32       4800        550      74400         62       7440     282720
 15   Sep         26       3900        880      72000         60       7200     273600
 16   Oct         47       7050       1045      54000         45       5400     205200
```

FIGURE 8 - Graph Type Menu Displays

lect it. The basic pie chart with exploded segments will be shown with the legend on the right side of the display.

Press {ESC} to return to the menu. Press (O)ptions and then <RETURN> to return to the global options menu. The chart that was displayed above did not have headings which are required to make it complete. To add the headings press (H)eadings. You are then presented with more menu options for the headings, both for headings and axis headings. ENABLE permits both a primary and secondary heading for each choice. For our example, select (M)ain and type in "XENIX WESTERN RAILROAD" and a <RETURN>. Then select (1) - Main-sub and type in "Profit Center Income" and a <RETURN>. Press {ESC} to move back to the Global Options menu. The default font that ENABLE uses for all headings and labels is a simple block font. Although this is acceptable, ENABLE has four other fonts and an enhancement of the added basic four fonts. From the Options Global menu select (F)onts. You can then select the font for each group of titles, labels, and legends. You select the location and then the type font you need. With this capability, it is possible to have four different fonts on one graph. For this example I have selected (2)-Roman for the (M)ain-titles and (R)oman1 for the (S)econdary-titles and default for the legends. Small printing on the screen does not look as good as with the other fonts when it comes to legends. Press {ESC} three times and you can now view your chart with the fonts by pressing (D)isplay. You can also print the graph by selecting (P)rinter. This completes the first graph for the project. As I said earlier, if the numbers should change, the graph will change without having to reinput the graph settings.

Although the pie charts does a good job, it does not present the side-by-side comparison of income vs. expenses that is required by financial managers. The bar chart with both the income and expenses would show this information, starting from the spreadsheet again. (You get there by using the {ESC} key to back out.) Note that on the spreadsheet that the total data for the income and expenses are aligned in adjacent cells with the basic legend provided. This data will be used to develop the bar graph for the presentation. Select the graphics module of ENABLE from the spreadsheet by pressing "F10 (G)raph". This is a new graph so (C)reate will be selected and the name "INCEXP1" will be typed and a <RETURN>. Press <RETURN> until the "Create INCEXP1 Options" menu is displayed. Press "1" for the first data set and then another <RETURN> to select the data set. In our example the data elements are located in cells C22 to C25. Move the highlighted cell marker to cell C22 and press ".." to lock this cell and then move the highlighted area down with the arrow keys to cell C25. The cells between C22 and C25 should now be highlighted and with a <RETURN> will be accepted. (L)egend is selected and "Income" is typed in. Press {ESC} to back out of the first data set and select the second data point by pressing "2". The second set of

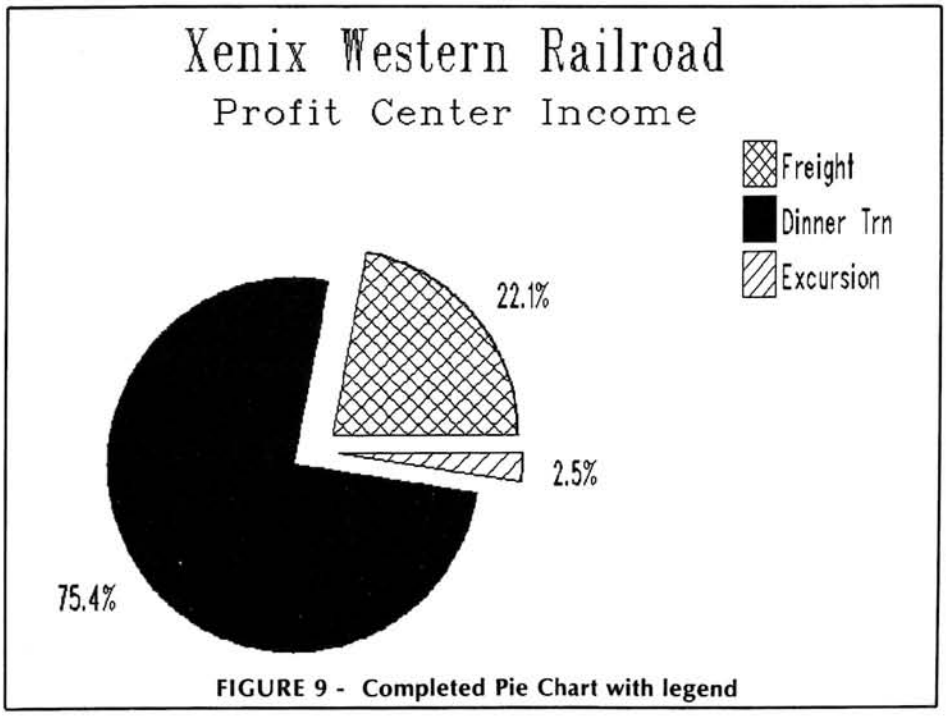


FIGURE 9 - Completed Pie Chart with legend

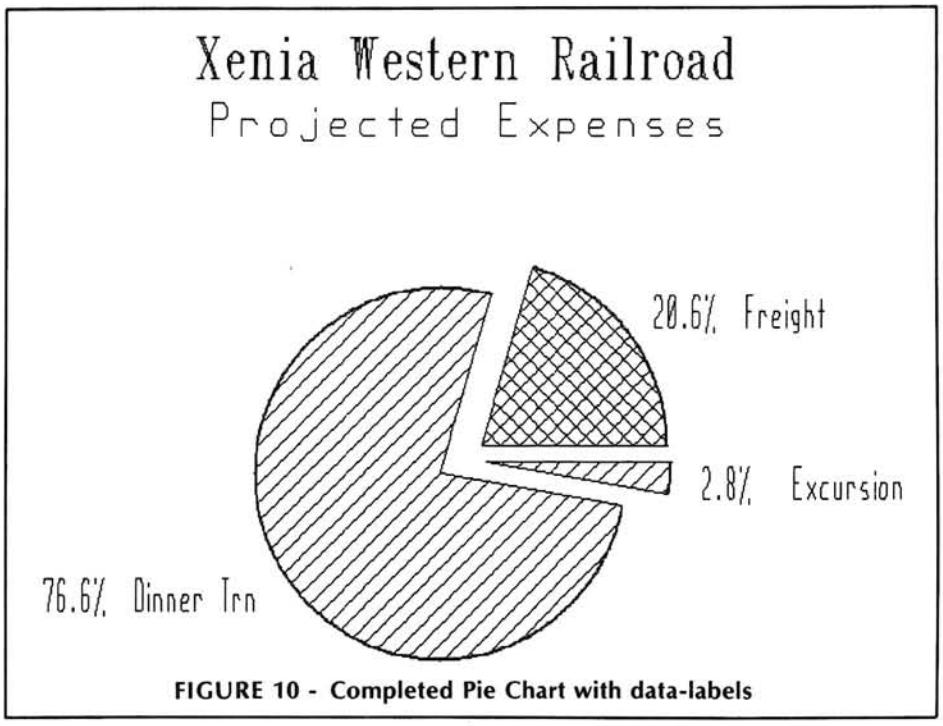


FIGURE 10 - Completed Pie Chart with data-labels

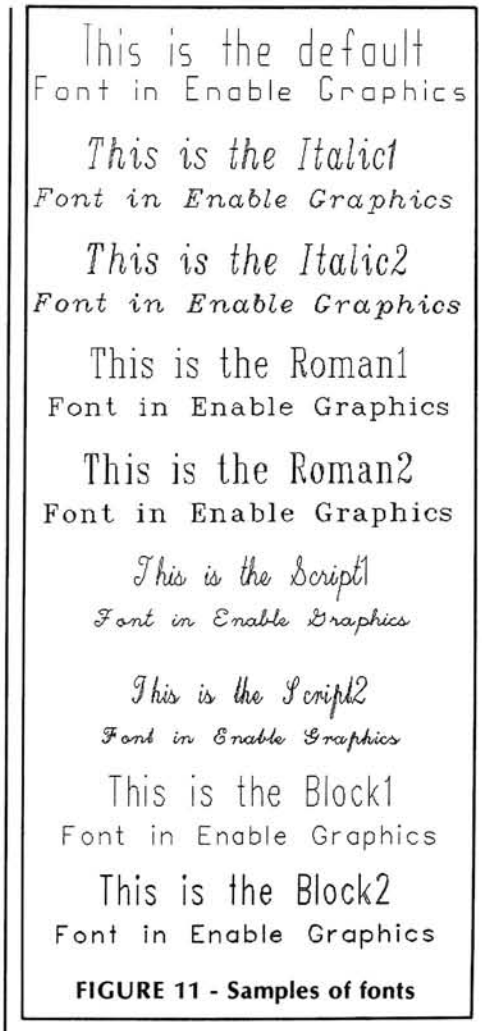


FIGURE 11 - Samples of fonts

numbers lay between cells B48 and B51. Select these numbers in the same manner as the income cells. The legend for this data set is "Expenses". Press {ESC} again to back out of the menu and press (G)lobal to select the (T)ype of graph. Select (V)ertical-bar and again another line will appear with four types of bar graphs. The default graph is a standard 2-D Standard graph that is like LOTUS and other spreadsheet graphics packages. ENABLE also provides a 3-D Standard bar graph that will be used in this example. The 3-D bar provides a very good visual image for small numbers of bars. When the number of bars gets large, the visual impact is lost as they appear almost like the 2-D bar. After selecting the "3D-Standard" graph, ENABLE prompts for the method of displaying the bars, either in (S)ets or (G)roup. Set will display the bars from the first data with the bars from the second data set (and additional data sets if selected) together. The second set of data elements will be displayed together and so on. If (G)roup is selected, all data elements from the first group will be displayed together and then the second group and so on. Press {ESC} to return to the "Create INCEXP1 Options Global".

I have found that adding a horizontal grid on the 3-D bar makes them easier to read. To add this function, select (G)rid and then (H)ortizontal. Grid will also add vertical, both 2 and vertical grids, or Clear the grids.

Moving along the menu, select (A)xis and (Y)-axis-format. From the options, select ($) and 0 decimal places. Both the X and Y axis have the same selection. The other options are fixed decimal points which if selected will display the numbers on the axis with the number of decimal points selected. The default is two. (I)nteger will only display whole numbers, rounding if necessary. ($) will display numbers with a "$" and with commas separating the numbers at the correct locations. You have the option to select the number of decimal places up to six. Negative numbers will be displayed in parentheses. (C)omma displays numbers with commas at the correct location but without the "$". (P)ercent will display the numbers

with the "%" sign, multiplying by 100. (D)ate will display the date selected in the format selected from five options. If you have large numbers, select (S)cientific and the result will be a display using scientific notation. The (G)eneral options adjusts the axis labels to fit available space. Also select (1)=X-axis-data. Move the cursor to cell A48 in the example and press ".." and move the cursor down to cell A51 and then a <RETURN>. This will display the contents of cells A48..A51 along the X-Axis of the chart, with the cell data below the bar of data.

The next area is the headings. The bars chart headings are constructed in the same manner as the pie chart. In addition to the Main and Main-sub headings, a Y-Axis heading called "Dollars" is also used. After the headings are constructed, the fonts are selected. Again, Roman is selected for the main heading and the default is maintained for the remainder of the headings. The Next option on the Global menu line is color. If a color is selected, it will be used for headings and other lines relating to the basic graph except for the bars. Colors for the bars are selected when you input the data points. Pressing the {ESC} key three times will return you to the menu where you can look at your creation.

One more bar chart is needed to display the Projected Freight Car Loadings. The data for this chart is obtained in cells A7..A18 and B7..B18. Construction of the chart is done in the same manner as the Income/Expenses chart. Because there are 12 bar entries and 12 X-Axis labels, the bottom of the chart could get very crowded. To make this easier to read, from the Create CARLDGS Options Global menu line, select (I)gn-pts. This ENABLE feature will permit you to select the number of points to be displayed on the X-Axis by responding to the number of points to ignore. This number can be between 1 and 255.

The last option on this menu line is (L)ayout. This option permits the user to adjust the chart for more of an impact. Three options along with the default value can be selected. Option 2 adds height to the graph while option 3 adds width. Option 4 increases the height and width of the graph while reducing the size of the lettering.

Using the {ESC} key, you can back out to the options menu and display the resulting graph. You can print the graph on a dot matrix or laser printer or send it to a plotter. The plotters supported by ENABLE are the HP 7470A, 7475A, and ColorPro 7440A; IBM 7371, 7372, and XY/749, Houston Instruments DMP-29-P and DMP-29-3, Graphtec MP-2000, and the Epson HI-80. These plotters are selected in the profile. In the Option Device menu, you can select or change the device setting. The three options are screen setting, printer options and plotter options. The screen device option permits changing the resolution. The Z-100 is a high resolution display and will not accept the medium resolution option. The printer option permits selecting the print density, either Single, Double, Triple, or Quad. Using the Format option, ENABLE will permit you to select either Land-

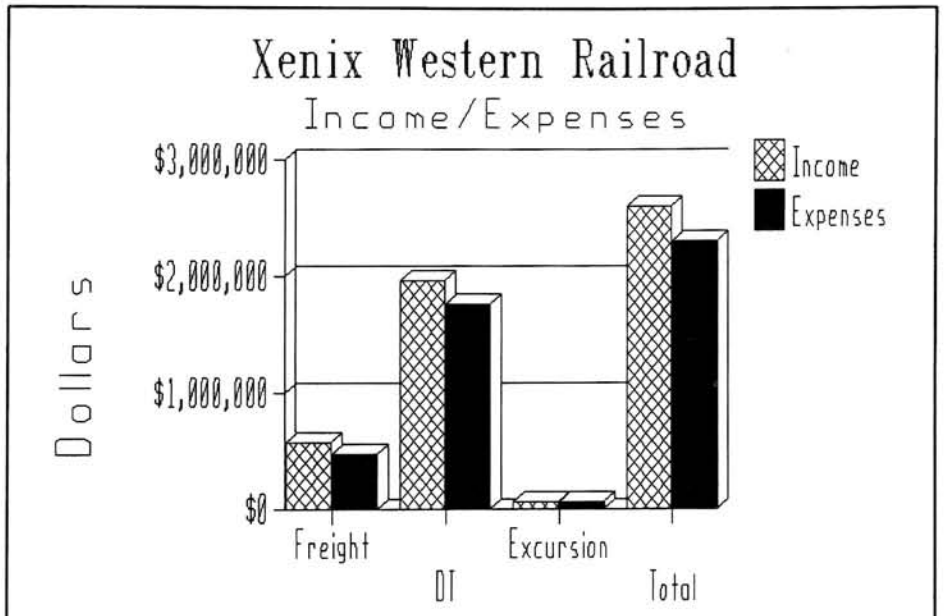


FIGURE 12 - Example 3-D Bar Chart

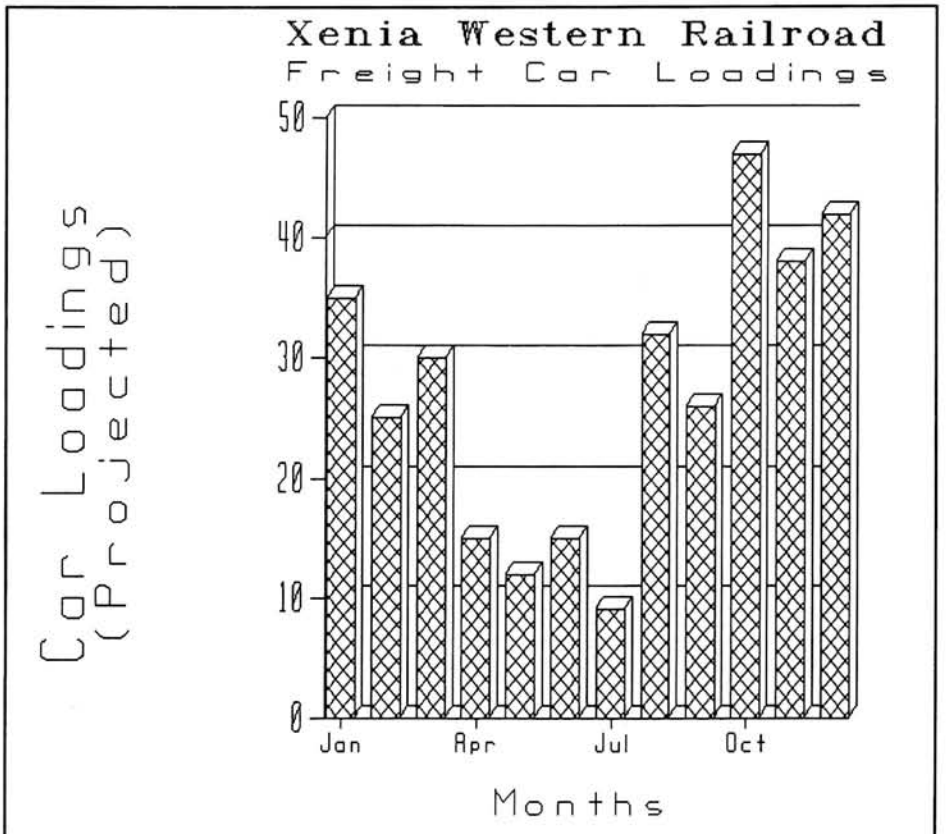


FIGURE 13 - Example 3-D Bar Chart with Ignored Points.

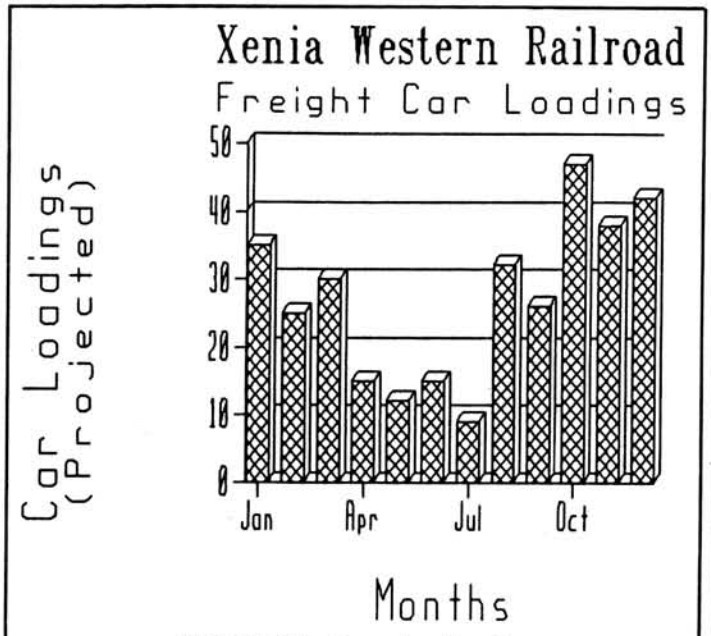


FIGURE 14 - Layout option two

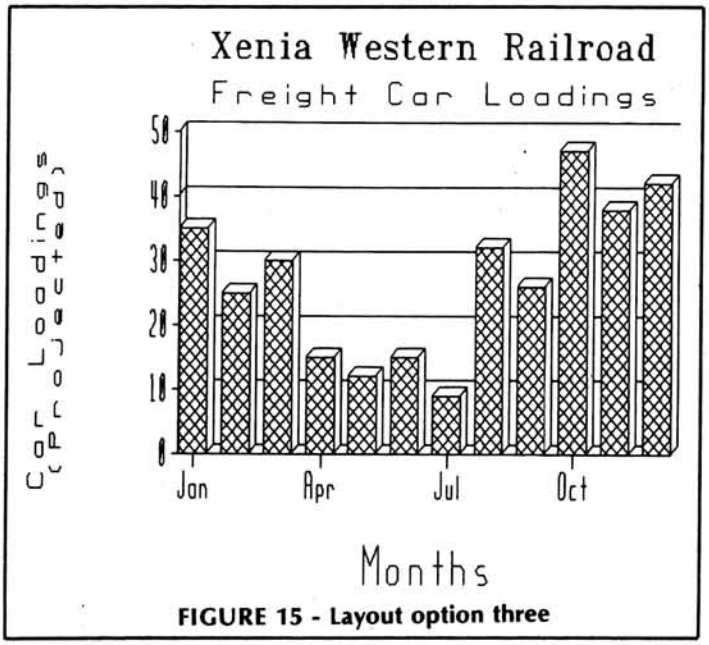


FIGURE 15 - Layout option three

scape, the default, or Portrait mode for printing. Portrait will permit the printing of two graphs on the same page. If Portrait is selected, you can select the top or bottom of the page for the location of the graph. Like in normal text printing, SHIFT/F0 F2 (CTRL/F2 for the PC) will stop the printing or plotting in progress. The plotter option permits the setting of Penwidth (Narrow or Wide), Speed (Fast or Slow), and to select the Comm-port (1 or 2).

This has been a start into the ENABLE graphics capability. In the next spreadsheet article we will cover the line, XY and Hi-Lo graphs along with an introduction to Perspective graphics. The Hi-Lo graph is the type that is used to display stock market information. The next spreadsheet article will be in three months. Next month, I will present the third article on databases with the start of the dot commands and report reformat.

*

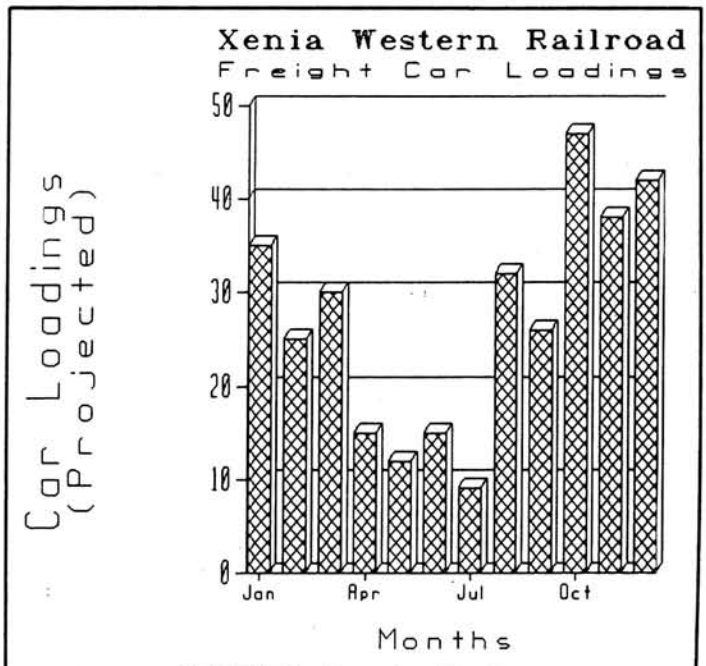


FIGURE 16 - Layout option four

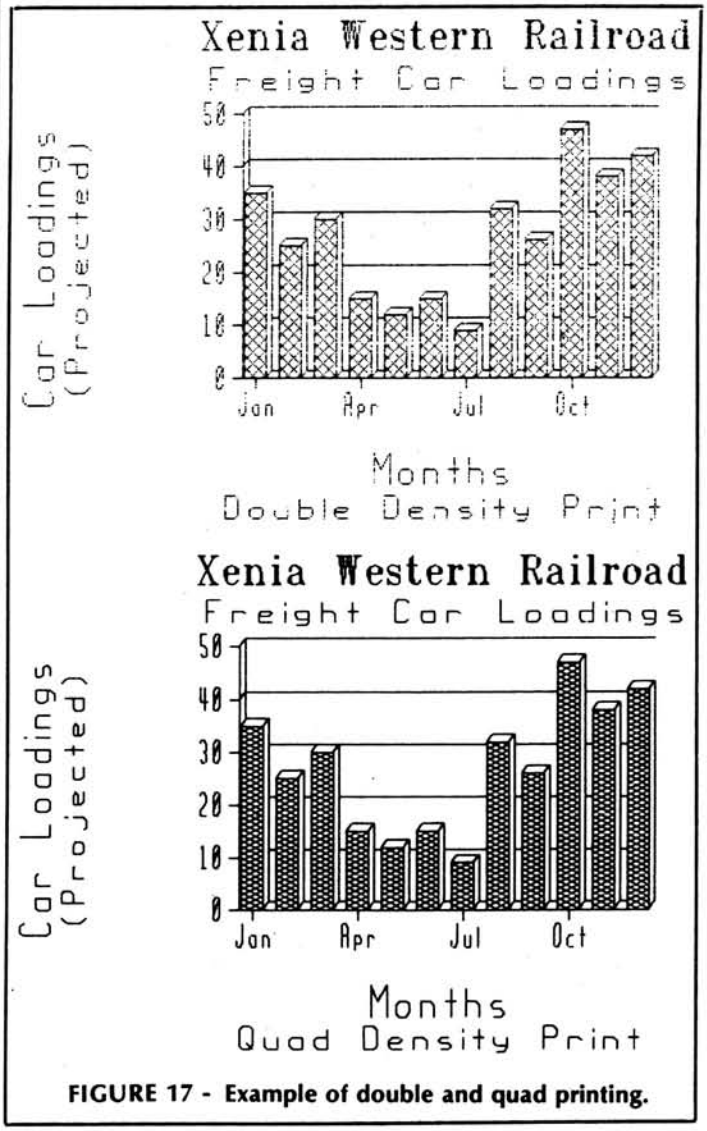


FIGURE 17 - Example of double and quad printing.

Getting Started With . . . Microsoft® Word

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

In a previous article (May 1988 issue of REMark), I mentioned how character formatting in Word is the most printer-dependent of the three formatting functions; character, paragraph and division. As printers become more powerful and with more features it becomes more of a challenge to get the most out of the machine. Fortunately, Word has always had excellent printer support. If you have been a Word user for any length of time and have been reading all the press about version 5 of Word Perfect, you recognize that it is just now getting some of the features that Word users have enjoyed since the start.

Since laser printers are becoming more affordable and represent a complexity at least one order of magnitude more than a dot-matrix printer, let's take a look at using one with Word. Specifically, we will talk about the Hewlett Packard LaserJet II; within their restrictions, the information will also apply to the older LaserJet Plus, LaserJet and many of the compatible printers.

The first problem I had was getting the silly printer to draw boxes with the format borders command. Instead of the line drawing character set, it was printing foreign language characters. An "easy fix" I thought: HP has it set up to use the wrong symbol set. Just use that neat little LCD readout and front panel switches to change the symbol set to one that has the PC graphics characters (IBM-US).

Changing the symbol set didn't do the trick so I asked Microsoft. The answer they gave was to modify the printer driver so that the correct symbol set was called for.

So I ran the utility makeprd.exe on hppccour.prd, and changed the 8Q on the beginmod line to 10U, as Microsoft had suggested, and it worked. But later I discovered that change was not needed. Seems the LaserJet manual doesn't say you have to turn the printer off before the changes you make with the control panel become effective. All I really needed to do to print the IBM graphic characters was follow the instructions on page 2-29 of the LaserJet-II manual and then turn the printer off and on again.

After using the printer for a bit, I discovered I could not get the format borders functions to print a bold box around a paragraph. Again, I talked to Microsoft and they kindly (and quickly) sent me a disk of print drivers just for the Series II that was supposed to cure the problem. Alas, still no bold box even with the new drivers. After printing a test file to disk, the light began to dawn when I saw the codes for bold were, in fact, being sent to the printer. I grabbed my font printout and discovered that the line drawing characters in the courier bold font were the same as the regular font. Progress is always slow, but never sure.

The simplest printer driver to use with the bare LaserJet II is hppccour.prd. In fact, you may use any one of the LaserJet drivers as they all support the basic printer. Except if you use any of the other drivers, you will be tantalized by the display of all the fonts you do not have. Regardless of which driver you use only the courier and lineprinter fonts will do you any good until you add cartridge fonts or download soft fonts.

At this point, you realize you have a very quiet and fast replacement for a daisy-wheel printer. By pressing alt+f8, you may select either the courier font at 12 point or the lineprinter at 8.5 point. Change the print driver to one of the landscape ones and you can print sideways, but still in the same dull monospaced fonts.

There are two choices for adding fonts to the printer: plugging in cartridge fonts or downloading soft fonts.

Cartridge fonts are expensive and their range of fonts is limited. They have the advantages of always being available and not requiring additional memory in the printer.

Soft fonts are much more versatile. Again, we find two choices: purchasing the bit-mappings from HP (or third party) or buying the font outlines from a third party, such as Bitstream, and generating your own bit-mappings.

The fonts HP provides are limited to 30 points in size and to the exact sizes they have selected. With an outline package and the program to generate the needed bit-mappings, any size you select may be generated, including fractional sizes, up to 120 points (a point is 1/72 of an inch). In addition, you can create your own custom symbol set.

My upgrade to Word 4.0 included an offer from SoftCraft for a free copy of their font installation program with the purchase of their Laser Fonts program. Laser Fonts included several bit-mappings of fonts, but no outlines. In retrospect, I might have done without Laser Fonts and used Word's utilities to create the print drivers and download the fonts. I understand Microsoft plans to include some fonts with new shipments of Word. This will enable you to get more use out of the printer right away and without extra cost.

Laser Fonts seems easier to use and more versatile. Version 3 enhances Word's ability to import graphic images, including side-by-side paragraphs, either text and graphics or two graphic images, despite the admonition in the Word manual (page 410) that it can not be done.

The font installation program and a font outline are used together to generate the bit-maps for each size you want to make. For each typeface you want you will need a font outline for that typeface. Prepare to spend a fair amount of time generating your bit-map files and set aside adequate space to store them on your fixed disk. A 10 point typeface will take about 15 minutes on my stock H/Z-150 and just over 21 kilobytes of disk space; a 48 point face requires over 324 kilobytes of storage and over half an hour of time. Some time and disk space may be saved by generating 7-bit fonts with only 128 characters instead of the full set of 256 for 8-bit fonts. The LaserJet II allows characters in the ASCII range 0-31 and 128-159 to be downloaded; other laser printers do not. However, Microsoft Word cannot access characters 0, 8-13 and 196.

Besides generating the sizes you want, you also need to generate separate bit-maps for those sizes in bold and italics and bold italics that you feel will be needed. Whereas a Postscript printer will use the outline to generate the font each time it is used, taking longer to print, the LaserJet requires the font to be generated and stored in advance.

At the time I generated my first set of fonts, I was not sure which symbol set to create. The symbol set allows you to print such characters as the IBM line drawing characters, the copyright symbol (©) and fractions (⅞). After a couple of false starts I modified a symbol set SoftCraft created. Their set includes many of the commonly used bullets, fractions and other symbols. I added the em and en space and the proper set of double quotes (“ ”).

Listing 1
fonts.bat

```
:fonts.bat
   echo off
:START
   cls
   type c:\batch\fontmenu.dat
   reply
   if errorlevel 101 if not errorlevel 102 goto CF8
   if errorlevel 91 if not errorlevel 92 goto SF8
   if errorlevel 90 if not errorlevel 91 goto SF7
   if errorlevel 85 if not errorlevel 86 goto SF2
   if errorlevel 68 if not errorlevel 69 goto F10
   if errorlevel 67 if not errorlevel 68 goto F9
   if errorlevel 66 if not errorlevel 67 goto F8
   if errorlevel 65 if not errorlevel 66 goto F7
   if errorlevel 64 if not errorlevel 65 goto F6
   if errorlevel 63 if not errorlevel 64 goto F5
   if errorlevel 62 if not errorlevel 63 goto F4
   if errorlevel 61 if not errorlevel 62 goto F3
   if errorlevel 60 if not errorlevel 61 goto F2
   if errorlevel 59 if not errorlevel 60 goto F1
   goto START
:F10
   copy c:\lf\reset.hp *lpt1:
   goto START
:F9
   copy c:\lf\delfonts.hp *lpt1:
   goto START
:F8
   copy c:\lf\formfeed.hp *lpt1:
   goto START
:SF8
   copy c:\lf\tray.hp *lpt1:
   goto START
:CF8
   copy c:\lf\manual.hp *lpt1:
   goto START
:SF7
   command /c downl olde
   goto START
:F7
   command /c downl script
   goto START
:F6
   command /c downl pccour
   goto START
:F5
   command /c downl helvet
   goto START
:F4
   command /c downl letters
   goto START
:F3
   command /c downl times_r
   goto START
:SF2
   command /c downl lf-2
   goto START
:F2
   command /c downl lf
   goto START
:F1
   help
```

Once you have generated the fonts, you will need to create a print driver and download the fonts. Laser Fonts makes it very easy to create a custom print driver and download the fonts that driver will use. Be sure you use the print driver that corresponds with the fonts you downloaded, unless you like surprises!

Normally, I load a set of fonts that will handle 90% of what I'll be doing. It contains Times Roman (Bitstream's Dutch) from 6 to 36 points, including some bold italics in the smaller sizes and a few sizes of a sans-serif font, including one size in reverse (white on black). Of course, the built-in Courier and Lineprinter fonts are available, too. In addition, I have a half-dozen or so other drivers and downloading files for specific tasks, such as the one requiring 48 point type.

I created a batch file, fonts.bat, that presents a menu of fonts to download, plus some printer control functions. Listing 1 is the batch file and Listing 2 is the data file, fontmenu.dat, that is typed out to the screen. The batch file uses the file reply.com to generate an errorlevel code to determine which key was pressed. There are several versions of such a function available in public domain. The printer control functions are small files, some furnished with Laser Fonts, that contain the necessary control codes to send to the printer, lpt1:. ANSI codes are used in fontmenu.dat to show different colors for the function keys shifted/control selections. Downl.bat is shown in Listing 3.

Even the small set of fonts that were furnished with Laser Fonts were more than enough to fill the memory of the basic LaserJet II. About 390 kilobytes of the 512 kB furnished is available for downloading and you have to leave the room to fit the document in memory, too.

Needless to say, I've had to add memory to the printer. One document I prepared required 48, 14 and 10 point size type, leaving no room for the document without added memory. Instead of the HP board, which is not upgradeable, I purchased a board from Pacific Data Products. The PDP board came with 1 Meg of memory and can be upgraded to 2 Meg or 4 Meg by replacing the 32-256K chips with 16 or 32-1 Meg chips.

With the fonts in and the right print driver selected, Word works just as always, except you have a stable of high-quality fonts to choose from. Hold on, there is more.

The printer will also do some nice things like cross-hatching (referred to in the manual as "HP Pattern"), grey scale and rules. I've had success using these features by embedding the printer commands directly in the document. To really work with the LaserJet II, you will need the Technical Reference Manual from HP. The part number is 33440-90905 and the cost is around $30 (call 800-538-8787).

Here are a couple of examples, one using the push/pop method. In these examples, the escape is shown by a ^ and each command sequence is shown on a separate line to improve the clarity. The actual command should be all one line with no spaces. Do not substitute lower-case letters for upper-case or vice versa.

Grey scale can be used to add a nice accent to a title, line drawing or just to break up the page. This LaserJet command paints a 15% grey scale 1200 dots wide (at

Listing 2 — fontmenu.dat

```
←[32;40m╔═══════════════════════════════════════╗
        ╚═ Download Fonts to LaserJet? ═╝
Function Key      ←[34;40mShifted      ←[31;40mControl←[0m
←[32;40m╔═══╗                          All Fonts
╚ F1 ╝ Return to DOS          ╔═══╗ ←[34;40mAll Fonts-2←[32;40m
                              ╚ F2 ╝
                              ←[m

←[32;40m╔═══╗                          ╔═══╗
╚ F3 ╝ Times Roman            ╚ F4 ╝ For Letters
                              ←[m

←[32;40m╔═══╗                          ╔═══╗
╚ F5 ╝ Helvetica              ╚ F6 ╝ PCCourier
                              ←[m

←[32;40m╔═══╗ Script                        ╔═══╗ Form Feed
╚ F7 ╝ ←[34;40mOld English←[32;40m          ╚ F8 ╝ ←[34;40mTray
Feed←[32;40m
                              ←[31;40mManual Feed←[m

←[32;40m╔═══╗                          ╔═══╗
╚ F9 ╝ Delete Fonts           ╚ F10╝ Reset Printer
                              ←[m
```

For clarity, here is fontmenu.dat without the ANSI codes:

```
        Download Fonts to LaserJet?
Function Key      Shifted      Control
                                All Fonts
F1  Return to DOS           F2  All Fonts-2

F3  Times Roman             F4  For Letters

F5  Helvetica               F6  PCCourier

F7  Script                  F8  Form Feed
    Old English                 Tray Feed
                                Manual Feed

F9  Delete Fonts            F10 Reset Printer
```

Listing 3
down1.bat

```
echo off
cls
if "%1" == "" %0 lf
echo ------------------------------------------------

echo Downloading %1.prs to printer
echo ------------------------------------------------

cd \lf
lasfonts %1.prs %2 %3 %4 %5
cd\
```

300 dots per inch that is 4 inches wide) and 425 dots high. It starts 10 dots below the current cursor position:

```
<--*p+10Y
   (moves 10 dots down the page)
<--*c1200a425b15g2P
   (the actual graphics)
<--*p-10Y
   (moves 10 dots up to starting point)
```

It took some fiddling to get the right numbers for the correct size of the pattern and its starting point in order to cover the desired spot on the page. This command was placed on the line before the area to be covered with the grey scale.

This line uses the push/pop method to generate two vertical rules or lines, 4 dots wide and 2800 dots long. Both rules start 315 dots from the top of the page; the first is 825 dots from the left side and the other is 1555 dots:

```
<--&f0S
   (store the cursor position-push)
<--*p825x315Y
   (re-position the cursor)
<--*c4a2800b0P
   (the first rule)
<--*p1555x315Y
   (re-position the cursor)
<--*c4a2800b0P
   (the second rule)
<--&f1S
   (restore the cursor position-pop)
```

The technique is to use the push command to store or push the current location of the cursor into memory, give the commands to re-position the cursor to where the graphic is to be printed and draw the graphic, then pop the stored cursor position out of memory and move back to where you started.

One advantage of using push/pop is the command may be placed anywhere on the page. Regardless of what happens to the text these two rules will always appear in the same spot. If the entry is formatted as a header, then the rules will appear on any page the header is formatted to print.

The LaserJet command strings are logical, but complicated. Using Word's glossary will help speed the process up and reduce errors (well, maybe).

To make the push entry, type the following in your document: ESCAPE&f0S. Type the escape key by pressing and holding the alt key while you press 027 on the numeric keypad (not the numbers on the top row of the keyboard). Then press the ampersand (& or shifted 7), then the lower case "f", then "0" (zero) and then uppercase "S". When you are sure this is correct, highlight what you have typed, press escape C to bring up the Copy to: menu. For a glossary name, type "push" and press return.

Making the pop entry requires replacing the zero with a one (ESCAPE&f1S) and copying to the glossary entry "pop".

Now you will be able to type push (or pop), press f3 and have the LaserJet command entered. In a similar fashion, you can create additional glossary entries for your other common graphics commands.

For simple rules and boxes, Word's Format Borders command and Line Draw features are easier to implement than embedding printer codes. Sometimes though, it is easier using the printer codes. For example, if you wish to draw a box around the entire page, here are the printer codes which can be placed anywhere on the page. The page may have any kind of text and formatting applied. You try it and see what it takes using Word's built-in features:

```
<--&f0S          (push)
<--*p0x60Y       (upper left corner)
<--*c4a3180b0P   (left side of box)
<--*c2400a4b0P   (top of box)
<--*p2396x60Y    (upper right corner)
<--*c4a3180b0P   (right side of box)
<--*p0x3236Y     (lower left corner)
<--*c2400a4b0P   (bottom of box)
<--&f1S          (pop)
```

If you find embedding printer codes too much of a throwback to WordStar, Laser Fonts 3.0 also has provisions for working with the printer's rules and patterns.

The Public (Software) Library has several utilities available for laser printers. Disks 1-PR-864, -1225, -1226, -1262, and -1263 contain programs to print a display of all font files on your disk, including all of the symbol set, to print files four pages to a sheet, and some fonts, plus several other utilities, including a very handy one for envelope printing. Unfortunately, most of the fonts lack symbol sets and are not as good a quality as Bitstream's.

And speaking of envelopes — the fact that the LaserJet does not have to be set in manual feed mode or envelope or executive feed in order to print them, totally escaped me at first. I tried all sorts of ways to get printer codes into a Word document to switch to manual mode and executive size paper in order to print on that size paper. All efforts failed. Then I discovered anything you place into the manual feed slot zips right through for printing! How simple. Someone please tell me I was not the only one in the world to find this out the hard way.

SoftCraft, Inc.
16 N. Carroll Street, Suite 500
Madison, WI 53703
(800) 351-0500

The Public (Software) Library
P.O. Box 35705
Houston, TX 77235-5705
(713) 721-5205

Pacific Data Products
8525 Arjons Drive, Suite M
San Diego, CA 92126
(619) 549-0922

*

Getting Started With WORDSTAR

Peter E. Walberg
390 Loma Drive
Forsyth, IL 62535-0057

If you have access to Wordstar and have been afraid to try it, or if you've been wanting to learn it but haven't found any easy instructions, this article is for you. This article won't tell you how to get your computer started (there are lots of variations on this), but if you know how to boot your computer and get a directory of a disk, you know enough to start with this article. This article also will not make you an expert on the many features of Wordstar, but it will, in a very short time, make you productive. It will get you to the point where you can type, edit, and print letters or reports. Most of the instructions given here also apply to Newword, if you have that. In fact many of the basic commands are common for many editors, such as the editor with Turbo Pascal, Sidekick, and even the dBase editor.

Conventions

For purposes of this article, I will enclose the keys to be typed in brackets, e.g. [WS]. This will mean to push the "w" key, and then the "s" key. In most instances, the case of the letter will be insignificant: you can type a capital letter or a lower case letter, whichever is most convenient. In some cases, you will have to type a carriage return (may be marked "RETURN" or "ENTER" or have a bent arrow, starting down, and then turning left (<DDY), depending on your machine) after the entries. Normally, if there is only one letter or control character to type, or if something happens as soon as you hit a key, you probably won't need a CR. If you are responding to a prompt or a question that could require a response of several keys, you will need a CR to tell the computer that you've entered all you want, and it should go ahead. I will normally not specifically mention the carriage return. Where I do, I will identify it as CR, with no quotation marks or brackets. Somewhere on the left side of your keyboard will be a control key (usually marked "CTRL"). This key is very important for Wordstar commands. I'll refer to it by a caret. For example, [^k] means to hold down the CTRL key while you push the "k" key.

Getting Started

To start, turn on your computer and boot it. When you get the drive prompt (such as "A>"), then you are ready to start. The next steps depend on what drive configuration you have.

For computers with two or more drives:

Find your Wordstar disk and put it in a drive other than the one with the system disk that you booted from (probably drive B) and make that drive the default drive.

For computers with a hard disk: Find your Wordstar directory and make it the default directory.

For computers with only one disk drive: Find your Wordstar disk. Then you'll have to take your system disk that you booted from out of drive A, and then put your Wordstar disk in that drive.

From here on, the instructions are the same for all systems. Type [WS]CR. You should see a sign-on screen identifying the program and providing an opening menu. Just look at it for now and remember that it's there. We'll discuss some of the items on it as we go along. You may also see a list of files that are on the disk. Ignore them for now.

Typing

Type [d]. This tells Wordstar that you want to start typing something. The menu will change, and the program will ask you for a document to open. Type any legal filename, in other words, a maximum of eight letters. I suggest [test]. Then hit CR. Normally, Wordstar will tell you it can't find that file and ask if you want to make a new one. Type [Y]. In the case where you've made a file by that name before, you can work on it in the following instructions, or you can type [^KQ] and start with a new file name.

Now you'll get another screen, normally with a menu at the top of the screen, a line of hyphens with an "L", an "R", and several explanation points below the menu, and blank space on the screen below that. The cursor will be at the upper left corner of this blank space, and that's where you will be typing. There may also be some function key definitions at the bottom of the screen, depending on the version of the program you are using. Ignore them for now. Start typing. You may type meaningful sentences, or just random words, separated by spaces. Don't worry about spelling or punctuation. If you make a mistake, just keep going. We'll come back to them later. Type at least four lines full of words, but don't hit the CR. After you have at least four lines of words, then hit the CR.

Now look back at what you've typed. Notice that, although you didn't hit the CR, the words have become lines. Secondly, notice the right margin. It's all even. Then you may notice that, in some cases, there is more than one space between the words you typed, even if you typed only one. This is the way Wordstar forms paragraphs. If the word you are typing would run past the margin, Wordstar moves it down to the next line. Then it adds spaces between words to make the right margin come out even. These features can be turned off and on as you choose, but we'll come back to that.

Moving The Cursor

Now, try moving around on the screen. Use the CTRL key and the "A" "S", "D", "F", "E",and "X" keys. Note that ^E moves the cursor up, ^X moves it down. ^S moves it left, and D moves it right. ^A and ^F move it left and right, respectively, but a whole word at a time. These keys are not mnemonic -- "D" doesn't seem to stand for right, or "S" for left, for example. But look at the location of these keys on the keyboard and notice the pattern. Play around for awhile with these keys and notice what they can do and what they can't. Notice that you can't move beyond the end of what you've typed, such as to go to a new line. Notice, too, that you can't go back beyond the first word you typed, such as if you wanted to put something in front of what you already typed. These are characteristics of Wordstar, but don't worry, I'll tell you later how you can go either way from what you've already typed.

Deleting

Next, let's try something that word processors are good for, and typewriters aren't. Let's change something. Move the cursor to the top line, to a word near the middle of the line. Now type [^t]. What happened? Did the word the cursor was on disappear? Now type [^u]. What happened? Did the word come back? There are several keys you can use to erase a mistake. [^t] is the word erase command. It will erase from where the cursor is to the end of a word. [^g] is the character delete function. It will delete the character that the cursor is under. The backspace key, or [^h], will delete the character that the cursor is just past. In other words, it will backspace and delete the character that you backspace to. The delete key, depending on how your Wordstar is configured, will either act like the backspace key or like [^g]. [^y] also erases, but it is a little more radical. It will erase a whole line. Try it. Then type [^u]. What happened? [^u] is the unerase feature. It will not unerase a single character deleted with [^g], backspace, or delete, but it will unerase almost everything else.

Inserting Text

Now, you've learned to erase and replace what you erased. How about adding things. Easy! Just put the cursor where you want to add, and start typing. This is the way to add to the beginning or end of the short file you made earlier, too. Wait a minute, you say. What if I want to type over something? [^V] will toggle (that is turn on if off or turn off if on) the insert mode. On some computers, the "Insert" key will do the same thing. If the insert mode is on, whatever you type will just move the other stuff to the right. While insert is off, whatever you type will replace whatever was at the cursor before. Try it a little. Add a few words to your document. Then type over a few others to replace them.

How do you know whether the insert mode is on or off? Now it's time to look at the status line at the top of the screen. It tells you several things. Starting from the left, it tells you what file you're working on. Then it tells you what page you're on, what line, and what column. Then it may say "Insert", or there may be a space. Note as you hit [^v] a few times that the "Insert" comes and goes. That's how you tell if when you type something, it will move everything over or whether it will type over it. The status line will tell you other things, too, as you advance further, but we won't cover that right now.

Paragraph Forming

Now that you've erased, recovered, and added a few things to your file, look at the right margin. Is it still even, or is it jagged? Move the cursor to the beginning of your little file. Now hit [^B]. What happened? Did the paragraph get reformed and wind up with an even right margin? Now let's try something fancy. Type [^OR] then [35] CR. Do you see the line with all the hyphens and exclamation points below the menu and just above the document you typed? Did the "R" at the right end of the line move? This line is called the "ruler line". The "R" stands for the right margin. The "L" stands for the left margin. And all the exclamation points stand for tab

stops. The [^OR] command sets the right margin. A [^OL] command sets the left margin. Move the cursor to the beginning of your document, and type [^B]. What happened? Try changing the margins a few times and reformatting the paragraph. That's where word processors shine. If you typed a letter, and then decided you wanted different margins, your only option would be to start over from scratch. Here you can change them in the twinkling of an eye.

Block Functions

Now lets take a quantum leap. Move the cursor to the beginning of your "paragraph". Then type [^KB]. Then move the cursor to the end of your "paragraph" and type a CR. Then type [^KK]. Now type [^KC]. Repeat the [^KC] a few times. You should now have more than a screenfull of copies of your "paragraph". Now we can try something more. Remember the [^E] and the [^X]? Now try a [^R] and a [^C]. These two keys move a whole screen at a time, don't they? Now try a [^W] and a [^Z]. What happened? Did the whole document move a line at a time? Notice these keys are still in the pattern around the "SEDX" diamond. Now, you may ask, what were the [^KB], [^KK], and [^KC] for. [^K] is a command prefix for a group of "block" functions. [^KB] marks the beginning of a block. [^KK] marks the end of a block. Notice the mnemonics — Block. Then [^KC] Copies the marked block, or [^KV] moves it to wherever the cursor is. With these block functions, you can move a word, phrase, sentence, paragraph, or move wherever you want it. Try moving a few things to see how it works. You may want to reformat your paragraphs ([^B]) after you get done moving them.

Now, you may see that [^B] works only on the paragraph that contains the cursor at the time, and only for the part of the paragraph after the cursor. This is something to keep in mind. There is a way to reformat all of your paragraphs with one command, but we'll only touch that briefly at the end of the article. You may wonder, "What does Wordstar understand to be a paragraph?" As you were typing your original few lines of words, remember that I told you not to use the CR? Then when we marked and copied that block of text, I had you put in a CR before you marked the end of the block? Wordstar understands that a paragraph is from one CR to another. When you typed words that would go past the right margin and Wordstar moved them down to the next line, it didn't put in a real CR, so it can tell the difference between where it moved words down to a new line and where you hit the CR. Sometimes the places where Wordstar goes to a new line are called "soft" CR's and the ones you put in are called "hard" CR's, because as you reformat paragraphs, the "soft" ones may be moved, but the "hard" ones will always stay where you put them in the text.

Disk Files

So how can I print my letter? We're just getting to that. Wordstar works with files in two places--in the computer memory and on the disk. What you see on the screen is what's in the computer memory, but Wordstar can't print what's in memory. You might think this is a limitation, but actually, as you get more advanced, you will find that this feature allows you to be printing one document while you work on another and save time. So how do I get my file from the computer memory onto my disk? The block ([^K]) command prefix does more things than we've done up until now. Type [^K] now. Then look at the menu display*. It shows four options under "SAVE": "S" for "save & resume edit", "D" for "save document", "X" for "save & exit Wordstar", and "Q" for "quit without saving". Depending on your version, the exact words may differ. [^KS] will save the document onto the disk and then come back to the screen so you can continue editing or writing it. You should use this every once in awhile, especially when editing long files. This will save your file onto the disk so it's safe in case of power failure or any of a number of other tragedies, and you don't lose everything you've been working on. [^KD] will save the file and bring back the opening menu and a blank area to start all over. If you decide you just want to trash what you've been working on, you can type [^KQ], or if you've run out of time, and you just want to save what you've done and quit, you can type [^KX]. If you type [D] now (remember, you've already typed the [^K]), you'll save your file and get back to the beginning of Wordstar, the opening menu.

Printing

Now that you've saved your file to your disk, you can print it. Look at the opening menu, while it's there. We sort of brushed over it last time. Notice, especially, the "D", the "N", and the "P". The "D" is what we used when we started for working on a Document file. "N" starts you on a file, too. But the difference is that with "N", for Non-document, you don't get automatic new lines when you get to the margin, you don't get paragraph reforming, or any of the other fancy text formatting capability. You would use this for things that you want to format yourself, generally programs that require specific line arrangements.

Let's print the file. First make sure your printer is on and ready to go. Then type [P]. We won't go in to the other options on the opening menu in this article. When you typed "P", it told Wordstar that you want to print something. It asks you what document you want to print. Remember that you can only print something that's on your disk. Since we saved "TEST" before, let's print that. Type in [test] CR. Wordstar then asks you a series of questions about how you want to print, such as how many copies you want, whether to stop between pages so you can put in another sheet of paper, whether to use form feeds to go to the next page (some printers don't understand form feeds), what page you want to start or end on in case you don't want to print your whole file, and, depending on your version, maybe one or two more. Normally, you can just hit ESC (the Escape key on your keyboard), and Wordstar will print your file with all the standard defaults. Hit ESC now to print your file. If you didn't get your file printed, and your printer was connected and ready, then you may have to go into WSCHANGE or WINSTALL to change the printer that Wordstar is expecting. This is beyond the scope of this lesson.

Menus

You're almost ready to head out on your own, but we have one more thing to cover, so that you can better understand how to explore the capabilities of your Wordstar. We've already looked at the opening menu a little, and we've looked at part of the Block (^K) menu. Remember the move and copy block commands? This is also the menu you use to do things like save part of a file or merge two files together, copy, delete, or change the names of files, and other similar things.

Another menu, called On-screen, called

by typing [^O] (notice the mnemonics), is for things like changing the margins, tabs, and line spacing, centering titles, and changing things like whether the right margin is justified or not. In other words, what your file will look like ON the SCREEN. In order to see what it does, you have to be editing a document. If you just saved and printed your file, call it back again ([d], [filename]CR), or open another one. Then you can try the options on this menu. If you call up a menu, and then decide that it wasn't the menu you wanted, hit the SPACE bar to cancel that menu.

A third menu is for Printing options. You get it with [^P]. With this menu you can specify things like whether you want words underlined, or in bold, or in a different font or pitch. Depending on your particular computer and/or version of Wordstar, you may only see the control characters (such as ^B, ^S) inserted into your file. With other versions, the words that you've marked for underlining, or bold, or some other special printing attribute may be in color or inverse video on the screen.

The next menu is the Quick menu, called up with [^Q]. Remember the cursor and text moving commands? [^S], [^D], [^R], and so forth. You'll like this one. If you type [^QD], for example, the cursor will move to the right, but all the way to the end of the line. Similarly, [^QS] will go all the way to the left. Try the "SEDX" diamond with each key preceded by [^q]. Try [^qc] and [^qr], too. Are you impressed? The Quick menu also has commands to align all the paragraphs in your document, so you don't have to go to each one and use [^B]. This is also the menu you use for searching through your document for finding a particular word, and if you want, automatically replacing it throughout.

The last menu is the Help menu. You get this by typing [^J] (not ^H). If you forget the different control characters we've covered in this article, type [^J] for help. This menu will also tell you how you can turn off some of the menus after you've learned them, so you have more room on your screen for editing.

I've tried to introduce you to where to find things on the menus so you have some idea where to look when you say "I know I can do that, but I can't remember how". It's up to you to try the different things on the menus to become proficient with them. Even word processing is just like most anything else -- you have to practice to perfect your skills.

*This article assumes that Wordstar has not been customized already. If the menus don't show up, type (SPACE^j ^j3)CR. Then type (^K) again. *

How to Get the Most from a Zenith Laptop Computer

by Joseph Katz

Part 3: Traveling Software's LapLink transfers files across incompatible disk media in either direction. It's faster than a speeding bullet, more powerful than a locomotive.

Travelling Software's LapLink is a super program for laptop owners. In fact it's a complete package for transferring files between any two IBM compatible computers connected directly by a null modem cable. You'll want it if you often need to transfer files between one computer that has only 3.5-inch floppy diskette drives and another computer that has only 5.25-inch floppy diskette drives.

Don't limit your thinking to just Zenith's Z-183, Turbosport 386, Supersport 286, and Supersport laptop computers. Of course LapLink is what you need to transfer files from their disks to an IBM PC, XT, or AT compatible desktop computer. But it's what you need too if you own a Z-171 laptop computer and want to extend its life into the age of desktop computers with 3.5-inch floppy diskette drives.

Ignore the advice of anyone who says it's just as easy to use a standard communications program on each computer to do the job of transferring files between them. Sure you can do it. But standard communications programs are neither as easy nor as fast for this kind of work. A specialized program such as LapLink is so much faster and better in every way that the comparison really is invalid. If you have to transfer files between incompatible media on a regular basis, you want this kind of program. And if you want this kind of program, chances are you want LapLink.

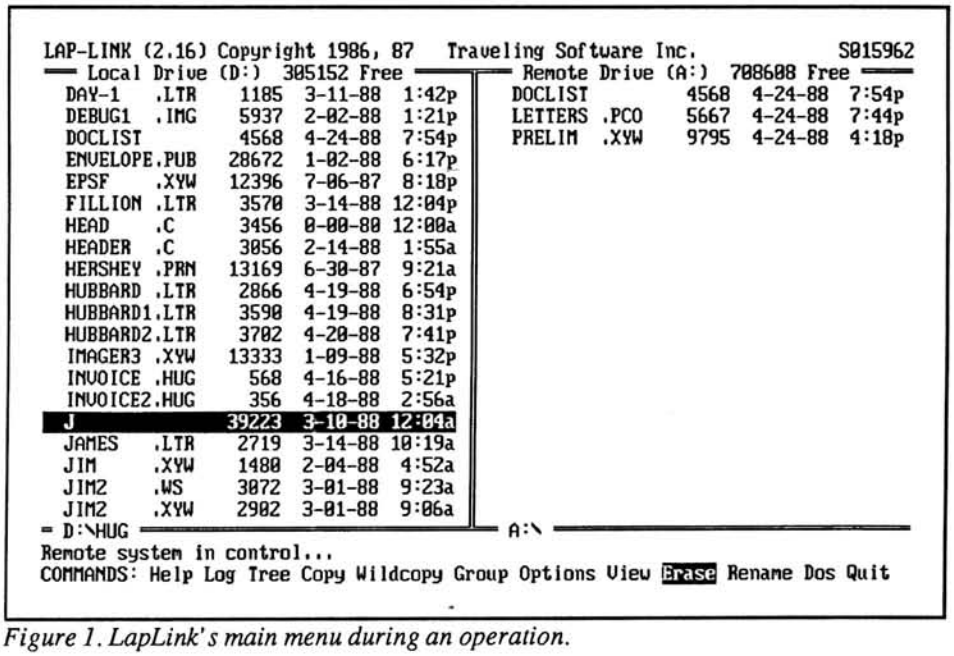


Figure 1. LapLink's main menu during an operation.

The package includes everything you need: one 3.5-inch program disk and one

Address all correspondence to Dr. Joseph Katz, 103 South Edisto Avenue, Columbia, South Carolina 29205.

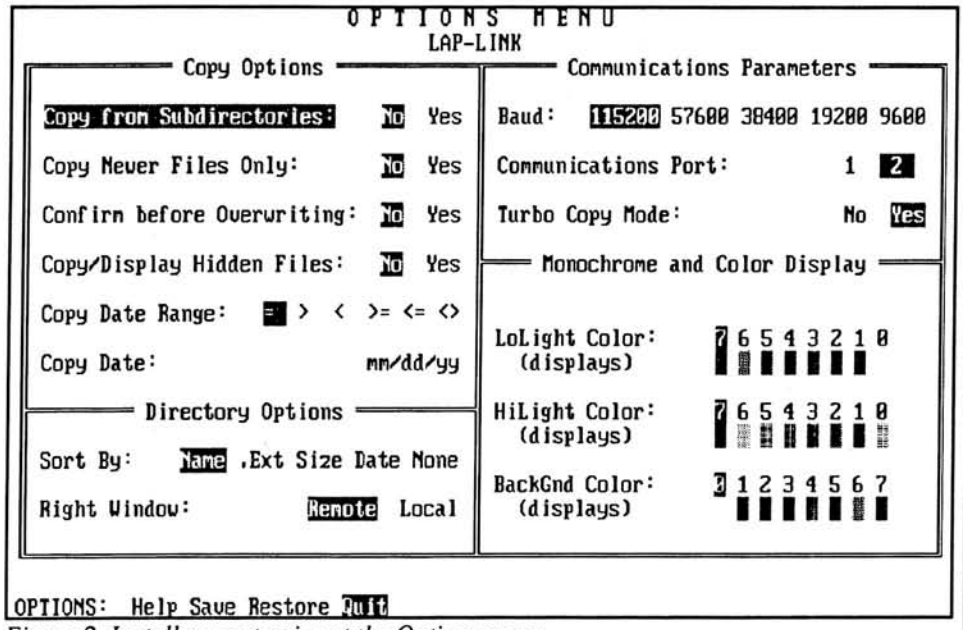


Figure 2. Install or customize at the Options menu.

5.25-inch program disk, a universal null modem cable that has both a DB-9 and a DB-25 connector on each end, and a manual. That universal null modem cable is well designed and well made. It's molded to prevent the cable from breaking or the wires coming loose from frequent flexing, and it has thumbscrews instead of slotted screws on every connector so you can lock the cable easily during a transfer session and unlock it just as easily when the session is over and you're ready to move on. That manual is equally well done. The only obvious omission is screen shots to illustrate what the words are saying. You'll therefore find useful the illustrations in this article.

A one minute manual

You might not even need the manual if you're at all familiar with disk management programs such as WindowDos or Xtree. You'll see in Figure 1 that the LapLink program looks like them. It also works like them. In use it's like having the same disk management program on each of two computers linked by the universal cable so that major disk management functions—such as copying and erasing files—work across the link. If you've ever used a disk management program, all you really need to do right now is cable the two computers together at their serial ports, run LapLink on each computer, and start transferring files.

If "disk management program" is a foreign term to you, you'll need to work a little harder at understanding LapLink. But not much harder, because disk management programs are a way to make easier the basic MS-DOS housekeeping functions—such as copying and erasing files—for people who find crucial MS-DOS commands too primitive. The typical disk management program therefore reads a disk's directory, displays a list of the files in it, and provides a menu of operations that can be performed on the files in that list. You select the file or files on which you want to operate, select the operation from the menu, and press the ENTER key to do what you want done. For anyone who has mastered the crucial MS-DOS housekeeping commands, therefore, LapLink's way of handling things will be a piece of cake. (If you need Help, choose it from the menu. It's not context sensitive help, but as you can see from Figure 3 it's effective and detailed.)

The major difference between what you see in a typical disk management program and what you see in LapLink's display is that there are two lists of filenames in the latter. The filenames on the left side of the display always are the contents of the current directory in the "local" computer, the one on which that copy of LapLink is running. The filenames on the right side of the display always are the contents of the current directory in the "remote" computer, the "other" computer. The orientation is always the same: left is local, right is remote, no matter which the computer. Nor is there's any other significance to the terms "local" and "remote": both computers are absolutely equal in a LapLink connection, at least until an operation (such as transferring files) that affects both computers is initiated at the keyboard of one computer. Then you'll see the message "Remote system in control" on the other. When the operation is finished, control is again available to either computer. You probably won't care about which computer is in control if you work alone, but you'll find it helpful to know on occasions when someone's at the other keyboard helping. Or trying to. Most times you'll probably want to work alone. LapLink is that easy.

You can work on either side of the screen from either computer: use the right or left arrow key to move a highlighted bar from one side to the other. Then, if what you want to do involves only one file at a time, use the up or left arrow key to scroll through the filenames. They'll continue "down" if there are more than twenty files in the directory and scroll "up" as you move through the list to its end. Of course you can always move backwards to see what you've passed. You select an operation by pressing the key for the first letter of the menu choice you want. If it's an operation that affects a single file, the operation will be performed on the filename highlighted by the cursor bar controlled by the arrow keys. So, as you can see from the main menu, you can Erase, Rename, or View (type to the screen) files on either computer. Those are handy options, the first two for routine housekeeping and the third for peeking inside a data file to make sure it's what you want to copy.

More powerful than a locomotive

"Copy"—following the terminology of those disk management programs it emulates—is LapLink's term for the transfer of files from one computer to the other. You have three ways to do it, in either direction, from the keyboard of either computer. All three ways to copy files are truly useful. The choice to use one or another method begins by deciding how many files in each directory you want to copy at one time.

If you want to copy one file at a time, move the highlighted bar to the filename and press "C" for "Copy."

If you want to copy a number of files that can be represented by a template using the standard MS-DOS * and ? symbols for ambiguous filenames, press "W" for "Wild-Copy.". Then, without regard to the file highlighted by the cursor bar, you'll be asked to type in the template you want used. Just as at the MS-DOS command line, the template BLITZ?.* would result in copying BLITZ1.COM, BLITZ2.DOC, and BLITZ3.TXT as well as any other files whose filenames fit the pattern. And, just

as at the MS-DOS command line, the template *.* would copy the entire current directory.

If you want to copy a number of files that can be represented by a template and you want to add or subtract a few as well, press "G" for "Group." Immediately you'll see a submenu for defining a group of files in just about every conceivable way. There are two things to remember about the Group option. First, it allows you to define a group of files by *whether or not* its members fit the template. Second, it works by tagging the files to be affected by an operation. That tag is represented by an arrow next to the filename. You can add tags or remove them on an individual basis, file by file, as you move the highlighted cursor bar and make the appropriate menu choice.

In the usual course of events you begin by typing "W" for "WildTag" to establish the template. If you next select the Copy option, the result is just about the same as if you had selected Wildcopy from the main menu—which is what you ought to have done if all you wanted to do was produce that same result.

Where WildTag begins being the more powerful option is in allowing you to scroll through the list of filenames to see if you really do want all those files included in the group. If you come upon any that you really don't want, Untag them while the highlighted cursor rests on their filenames. Or if you come upon any files you want copied even though they don't match the template, Tag them while the highlighted cursor rests on those filenames. When you've finished, the files tagged are those affected by the operation you choose. It can be the Copy option or, if you want to do the same kind of mass deletion possible with an ordinary disk management program, the Erase option.

WildTag's power is heightened by the Reverse option. You go into Reverse after you've established the template with WildTag. Then the tags are removed from the names of files that fit the template and are displayed next to the files that do not fit it. So if you used WildTag to establish the template BLITZ?.*, arrows would appear next to filenames such as BLITZ1.COM, BLITZ2.DOC, and BLITZ3.TXT. Then, if you typed "R" for "Reverse," those tags would disappear and only all the other files in the directory would be tagged.

Of course you then can go on to Tag or Untag individual files to your heart's content, adding them to or subtracting them from the group until you've molded it into precisely the shape you want.

```
---- REMINDERS IN MESSAGE LINE ----

The message line of the LAP-LINK screen always tells you which serial port and
which transfer rate is selected on each computer.  It also displays reminders
whenever you change the settings of any of the six Copy Options or the Turbo
Copy Mode:

< 12/05/88  SMOHT COM1: 115200
                        └─Baud set at 115200
                      └─Communications Port set at 1 (COM1)
                    └─Turbo Copy Mode set at Yes
                  └─Copy/Display Hidden Files set at Yes
                └─Confirm before Overwriting set at Yes
              └─Copy Newer Files Only set at Yes
            └─Copy from Subdirectories set at Yes
          └─Copy Date set at 12/05/88
        └─Copy Date Range set at < (for "before 12/05/88")


          Press any key to exit this menu.
```

Figure 3. Help explains a configuration message line.

But you don't have to start with WildTag to form a group. When you want to sculpt instead of mold, to chip away a few files from a heap of files with various kinds of filenames, the easier way to begin is by typing "A" for "All." Immediately you'll find a tag next to every filename in the current directory, at which point you can use Untag to pare away what you don't want included in the group.

All this is easier to do than to explain, which is why LapLink is so desirable. And don't worry about becoming confused and causing inadvertent damage. LapLink requires you to confirm erasures and, if you trip over your fancy footwork tagging and untagging, you can type "N" for "None" to untag everything and clear the decks for a new beginning.

Faster than a speeding bullet

It's fast because LapLink is designed specifically for transferring files between two computers cabled directly together. You don't have to go through any silliness of telling LapLink where to copy what. It's intelligent enough to know that you mean the copy to go to the "other" computer, whichever that happens to be.

LapLink is fast in absolute terms too. It's default communications speed is a dazzling 115200 bits per second. That's fast. You can cut the communications speed if one of the two computers can't handle the fast lane. You should have no trouble on this score with any Zenith IBM compatible computer in good repair. But if you have to negotiate a transfer with a computer that can't keep up, type "O" for "Options" and reduce the communications speed. The increments are 115200, 57600, 38400, 19200, and 9600 bits per second. Always work at optimum speed—the fastest speed both computers can manage while producing reliable file transfers—and always make sure both computers are set to the same speed.

At any speed you can go faster still by selecting LapLink's "Turbo" copy mode. It gulps in bigger blocks of data from the disk each time it makes a disk read. Even though the communications speed remains the same as in the normal mode, things go faster because there are fewer disk accesses. Disk accesses on both ends, of course, eat time. If you're using LapLink to transfer files between your laptop and another while both computers are operating on battery power, therefore, you'll get more out of the charge in Turbo mode. (If you're a spy doing a data drop between flights in an airport waiting room, or while in flight on the plane, you ought to use Turbo mode. Ignore the crowd that will gather to find out what you're doing.) Turbo mode is supposed to be more prone to transmission errors than the normal mode, but LapLink's error handling is so good you might not notice. If you do, switch to normal mode.

X-Ray vision

When you're finished transferring files from and to the current directories of each computer, LapLink makes it easy to change directories on either or both. Type "L" for "Log" and respond to the prompt with the

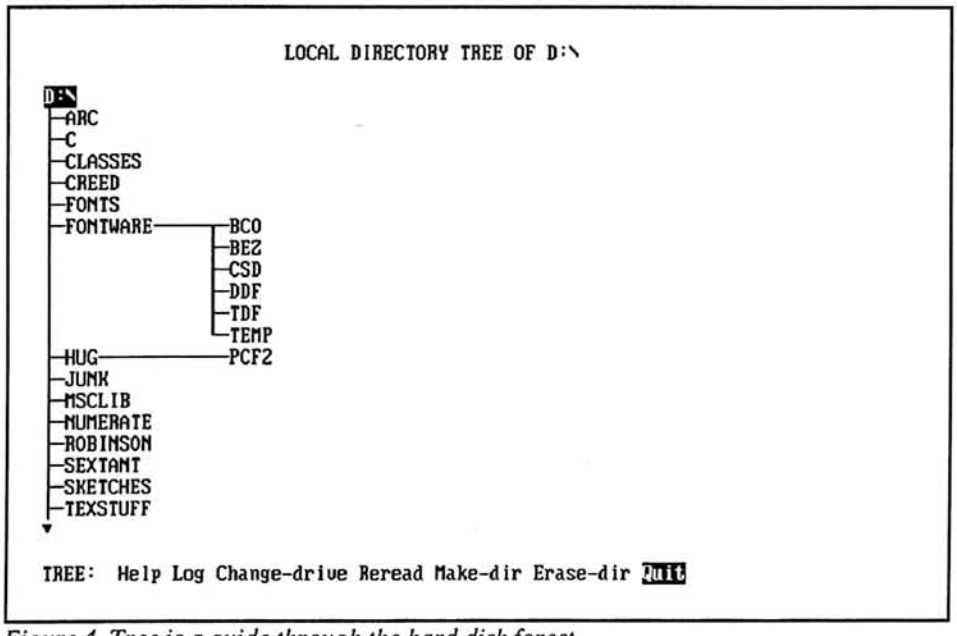


Figure 4. Tree is a guide through the hard disk forest.

exact place you want to reach. You may use the complete pathname, including the drive. So you don't have to change drives first, then change directories.

If you've lost your place on a big hard disk, or one that has a complex organization, type "T" for "Tree" and you'll be rewarded with a map of the terra incognita. Of course you can use Tree also to find your way around the local computer in case you get lost on your own turf and are embarrassed to admit it.

If you have to attend to other business while you're linked to a remote computer but don't want to exit LapLink for a few moments, type "D" for DOS command and you can drop down to the MS-DOS command line. You then can perform normal MS-DOS operations and even run other programs. When you've finished, type Exit at the command line and you're back in LapLink and still connected to the remote computer. You'll need sufficient RAM to accomodate both programs and an extra load of the MS-DOS command processor all at once, of course, but if your computer is a Zenith Z-183 laptop computer or some other that has 640 KB of RAM, and if you don't routinely run RAM-resident programs that hog memory, you can do it. What you have to remember, though, is to not do things such as rename or remove programs or directories that were tagged in LapLink before you used the DOS command. If they're not there when you return to LapLink, it will be confused. Who wouldn't be when they rules are changed? Use common sense and LapLink will reward you with fast, faithful service.

"It's a bird, it's a plane, it's . . ."

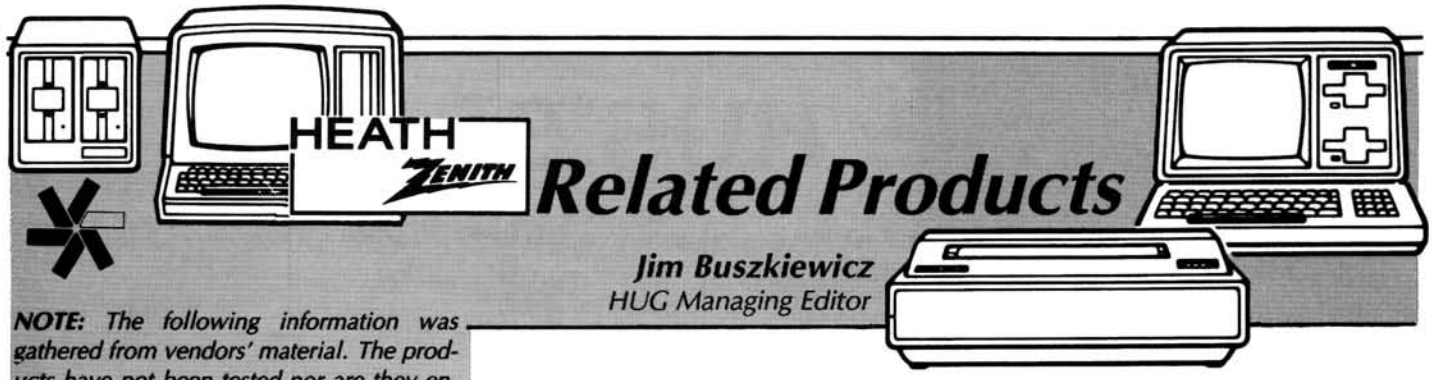
It's almost too easy to call "installation." LapLink is one program, LL.EXE. Copy it to the hard disk or put it on a working floppy diskette (you're better off if the diskette has the MS-DOS system on it) and plug a computer into each end of the cable. Then run the program. It's already installed for communications at 115200 bits per second through COM1. You can change those values, and others, at any time with the Options menu illustrated in Figure 2. Simply move the highlighted cursor bar to your selection and press the ENTER key. If the change is one you want kept for the next time you use that copy of the program, make sure to Save it.

You should be aware of the implications to one installation option you might miss. You can configure LapLink so the right window is "Local" instead of remote. Then, when you return to the main menu, you can Log that window to a local drive and directory. And what do you have then? You have a disk management program you can use on a single computer until the next time you need to transfer files with another computer. As I've said, LapLink is a super program..

Product Information

LapLink. $129.95
Traveling Software, Inc.
North Corporate Center
19310 North Creek Parkway
Bothell, WA 98011
206/483-8088

*



Related Products

Jim Buszkiewicz
HUG Managing Editor

***NOTE:** The following information was gathered from vendors' material. The products have not been tested nor are they endorsed by HUG. We are not responsible for errors in descriptions or prices.*

St. Louis, MO . . . **Hogware Company** announces the release of version 2 of ShowOff, the high resolution graphics editor for the Z-100.

This is a major revision of the popular Z-100 program that includes the addition of the following features:

- ShowOff pictures can be used directly by all popular desktop publishing software.
- ShowOff can read clip art files available from many software publishers.
- Text functions can now generate perfectly formed characters of variable width, of variable height, up to full screen size and at any angle.
- Individual pixels can be magnified for viewing and editing with zoom edit.
- Smooth curves can be generated for fitting experimental data or for intricate drawings.
- Any area of the screen can be resized — reduced down to 10% of original or enlarged to full screen. The change in size is continuously variable in both x and y directions, so that an area can be stretched in one direction and compressed in the other.
- Any area of the screen can be resized vertically and tilted to any angle.

ShowOff $95, with Logitech mouse $185. The update to this new version of ShowOff for current ShowOff owners is $15. (Shipping for software only $2, mouse $4)

Available from Hogware Company, 470 Belleview, St. Louis, MO 63119, (314) 962-7833.

For more information, contact Janet M. Hirsch (314) 962-7833.

St. Louis, MO . . . **Hogware Company**, the publisher of ShowOff, the high resolution graphics editor for the Z-100, has introduced the ShowOff Art Gallery.

ShowOff Art Gallery is a collection of high resolution graphic images that can be used with ShowOff on the Z-100, or with popular desktop publishing software running on any MS-DOS computer.

The Art Gallery will include disks of pictures in the following categories: medicine, business, computers, chemistry, electronics, transportation, animals, and art deco.

The initial entry in the Art Gallery is the Sampler Disk priced at $15 ($2 s/h) available from Hogware Company, 470 Belleview, St. Louis, MO 62119 (314) 962-7833.

For more information, contact Janet M. Hirsch (314) 962-7833.

MetraBytes' MBC-BASIC is a high level programming language that is virtually 100% compatible with GWBASIC and BASICA. MBC-BASIC has been specifically designed to provide the user an easy method of developing sophisticated, custom software for industrial data acquisition applications.

Unlike many forms of BASIC, MBC-BASIC includes a structured programming format (e.g., C, Fortran, etc.) which allows the creation of well organized programs with procedures and functions that are easily understood and identified. Variables may be declared as local (within a procedure) or global (external). Procedures and Functions are created simply and may be stored as modules for re-use in future programs. Libraries may be created to store pre-tested, error-free code for repeated use.

MBC-BASIC also allows the creation of up to 5 windows. Each window includes independent scrolling, borders, titles, foreground and background colors, and character attributes. Windows may display up to 8 colors on the screen at one time.

A powerful feature of MBC-BASIC is the ability to create User Defined Key Words. This allows the user to tailor the language to the specific needs of a particular application. MetraByte has developed a number of special Keywords to optimize MBC-BASIC for data acquisition applications. These keywords support the MetraBus; a family of industrial and process control monitoring products offered by MetraByte.

Application flexibility is also increased through the programmers ability to access the full 640 kBytes of memory that is available to DOS. Unlike MBC-BASIC, most BASIC languages are limited to 64 kBytes.

For more information, contact MetraByte Corp., 440 Myles Standish Boulevard, Taunton, MA 02780. (508) 880-3000. Price: MBC-BASIC — $195.00.

Continued on Page 82

The following HUG Price List contains a list of all products in the HUG Software Catalog and Software Catalog Update #1. For a detailed abstract of these products, refer to the HUG Software Catalog, Software Catalog Update #1, or previous issues of REMark.

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

HEPCAT is here! *HEPCAT* is here! *HEPCAT* is here! So what is *HEPCAT*, you may ask? Why it's just another Pat Swayne SUPER-UTILITY. *HEPCAT* is an acronym for *HUG Engineer's and Programmer's CAlculation Tool*. Just what we don't need, another memory resident calculator, right? Wrong! With *HEPCAT*, you can throw away the rest and use the best. *HEPCAT* only uses two partial lines on your screen, and best of all, does NOT cause existing programs to stop executing! That means, while your computer is grinding numbers internally, you can be grinding them externally. Order ***HUG P/N 885-3045-37***.

Can't remember how to use the MS-DOS 'COPY' command? Forget the exact command line format for 'ASGNPART'. Too far to go for the MS-DOS manuals on the shelf on the other side of the room? Why not just type 'HELP' on the keyboard? You say it comes back with "Bad command or file name"? It wouldn't if you had HUG's ***HELP*** program. With *HELP* installed on your hard disk, all you need to do is type 'HELP' for a complete list of MS-DOS commands and transients along with a brief explanation of how each command works, as well as the format for its use. *HELP*, ***HUG P/N 885-8040-37***, works on *ALL* Heath/Zenith computers that run MS-DOS!

HUG Price List

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
H8 — H/Z-89/90				
ACCOUNTING SYSTEM	885-8047-37	CPM	BUSINESS	20.00
ACTION GAMES	885-1220-[37]	CPM	GAME	20.00
ADVENTURE	885-1010	HDOS	GAME	10.00
ASCIRITY	885-1238-[37]	CPM	AMATEUR RADIO	20.00
AUTOFILE (Z80 ONLY)	885-1110	HDOS	DBMS	30.00
BHBASIC SUPPORT PACKAGE	885-1119-[37]	HDOS	UTILITY	20.00
CASTLE	885-8032-[37]	HDOS	ENTERTAINMENT	20.00
CHEAPCALC	885-1131-[37]	HDOS	SPREADSHEET	20.00
CHECKOFF	885-8010	HDOS	CHECKBOOK SOFTWARE	25.00
DEVICE DRIVERS	885-1105	HDOS	UTILITY	20.00
DISK UTILITIES	885-1213-[37]	CPM	UTILITY	20.00
DUNGEONS & DRAGONS	885-1093-[37]	HDOS	GAME	20.00
FLOATING POINT PACKAGE	885-1063	HDOS	UTILITY	18.00
GALACTIC WARRIORS	885-8009-[37]	HDOS	GAME	20.00
GALACTIC WARRIORS	885-8009-[37]	CPM	GAME	20.00
GAMES 1	885-1029-[37]	HDOS	GAMES	18.00
HARD SECTOR SUPPORT PACKAGE	885-1121	HDOS	UTILITY	30.00
HDOS PROGRAMMERS HELPER	885-8017	HDOS	UTILITY	16.00
HOME FINANCE	885-1070	HDOS	BUSINESS	18.00
HUG DISK DUPLICATION UTILITIES	885-1217-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG	885-4500	VARIOUS	PRODUCTS THRU 1982	9.75
HUGMAN & MOVIE ANIMATION	885-1124	HDOS	ENTERTAINMENT	20.00
INFO. SYSTEM AND TEL. & MAIL SYSTEM	885-1108-[37]	HDOS	DBMS	30.00
LOGBOOK	885-1107-[37]	HDOS	AMATEUR RADIO	30.00
MAPLE	885-8005	HDOS	COMMUNICATION	35.00
MAPLE	885-8012-[37]	CPM	COMMUNICATION	35.00
MICRONET CONNECTION	885-1122-[37]	HDOS	COMMUNICATION	20.00
MISCELLANEOUS UTILITIES	885-1089-[37]	HDOS	UTILITY	20.00
MORSE CODE TRANSCEIVER	885-8016	HDOS	AMATEUR RADIO	20.00
MORSE CODE TRANSCEIVER	885-8031-[37]	CPM	AMATEUR RADIO	20.00
PAGE EDITOR	885-1079-[37]	HDOS	UTILITY	25.00
PROGRAMS FOR PRINTERS	885-1082	HDOS	UTILITY	20.00
REMARK VOL 1 ISSUES 1-13	885-4001	N/A	1978 TO DECEMBER 1980	20.00
RUNOFF	885-1025	HDOS	TEXT PROCESSOR	35.00
SCICALC	885-8027	HDOS	UTILITY	20.00
SMALL BUSINESS PACKAGE	885-1071-[37]	HDOS	BUSINESS	75.00
SMALL-C COMPILER	885-1134	HDOS	LANGUAGE	30.00
SOFT SECTOR SUPPORT PACKAGE	885-1127-[37]	HDOS	UTILITY	20.00
STUDENT'S STATISTICS PACKAGE	885-8021	HDOS	EDUCATION	20.00
SUBMIT (Z80 ONLY)	885-8006	HDOS	UTILITY	20.00
TERM & HTOC	885-1207-[37]	CPM	COMMUNICATION & UTILITY	20.00
TINY BASIC COMPILER	885-1132-[37]	HDOS	LANGUAGE	25.00
TINY PASCAL	885-1086-[37]	HDOS	LANGUAGE	20.00
UDUMP	885-8004	HDOS	UTILITY	35.00
UTILITIES	885-1212-[37]	CPM	UTILITY	20.00
UTILITIES BY PS	885-1126	HDOS	UTILITY	20.00
VARIETY PACKAGE	885-1135-[37]	HDOS	UTILITY & GAMES	20.00
VOLUME I	885-1008	N/A	SOFTWARE LISTINGS	9.00
VOLUME II	**SOLD OUT**		SOFTWARE LISTINGS	12.00
VOLUME III			SOFTWARE LISTINGS	9.00
VOLUME IV	885-1037	N/A	SOFTWARE LISTINGS	12.00
WATZMAN ROM SOURCE & DOC	885-1221-[37]	CPM	H19 FIRMWARE	30.00
WATZMAN ROM	885-4600	N/A	H19 FIRMWARE	45.00
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	CHECKBOOK SOFTWARE	25.00
COPYDOS	885-1235-37	CPM	UTILITY	20.00
DISK DUMP & EDIT UTILITY	885-1225-[37]	CPM	UTILITY	30.00
DUNGEONS & DRAGONS	885-1209-[37]	CPM	GAMES	20.00
FAST ACTION GAMES	885-1228-[37]	CPM	GAME	20.00
FUN DISK I	885-1236-[37]	CPM	GAMES	20.00
FUN DISK II	885-1248-[37]	CPM	GAMES	35.00
GAMES DISK	885-1206-[37]	CPM	GAMES	20.00
GRADE	885-8036-[37]	CPM	GRADE BOOK	20.00
HRUN	885-1223-[37]	CPM	HDOS EMULATOR	40.00
HUG FILE MANAGER & UTILITIES	885-1246-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PRODUCTS 1983 THRU 1985	9.75
KEYMAP CPM-80	885-1230-[37]	CPM	UTILITY	20.00
MBASIC PAYROLL	885-1218-[37]	CPM	BUSINESS	60.00
MICRONET CONNECTION	885-1224-[37]	CPM	COMMUNICATION	16.00
NAVPROGSEVEN	885-1219-[37]	CPM	FLIGHT UTILITY	20.00
REMARK VOL 3 ISSUES 24-35	885-4003	N/A	1982	20.00
REMARK VOL 4 ISSUES 36-47	885-4004	N/A	1983	20.00
REMARK VOL 5 ISSUES 48-59	885-4005	N/A	1984	25.00
REMARK VOL 6 ISSUES 60-71	885-4006	N/A	1985	25.00
REMARK VOL 7 ISSUES 72-83	885-4007	N/A	1986	25.00

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
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 UTILITIES FOR MBASIC	885-1231-[37]	CPM	UTILITY	20.00
ZTERM	885-3003-[37]	CPM	COMMUNICATION	20.00
H/Z-100 (Not PC) Only				
ACCOUNTING SYSTEM	885-8048-37	MSDOS	BUSINESS	20.00
CALC	885-8043-37	MSDOS	UTILITY	20.00
CARDCAT	885-3021-37	MSDOS	BUSINESS	20.00
CHEAPCALC	885-3006-37	MSDOS	SPREADSHEET	20.00
CHECKBOOK MANAGER	885-3013-37	MSDOS	BUSINESS	20.00
CP/EMULATOR	885-3007-37	MSDOS	CPM EMULATOR	20.00
DBZ	885-8034-37	MSDOS	DBMS	25.00
ETCHDUMP	885-3005-37	MSDOS	UTILITY	20.00
EZPLOT II	885-3049-37	MSODS	PRINTER PLOTTING UTILITY	25.00
GAMES CONTEST PACKAGE	885-3017-37	MSDOS	GAMES	25.00
GAMES PACKAGE II	885-3044-37	MSDOS	GAMES	25.00
GRAPHICS	885-3031-37	MSDOS	ENTERTAINMENT	20.00
HELPSCREEN	885-3039-37	MSDOS	UTILITY	20.00
HUG BACKGROUND PRINT SPOOLER	885-1247-37	CPM	UTILITY	20.00
KEYMAC	885-3046-37	MSDOS	UTILITY	20.00
KEYMAP	885-3010-37	MSDOS	UTILITY	20.00
KEYMAP CPM-85	885-1245-37	CPM	UTILITY	20.00
MAPLE	885-8023-37	CPM	COMMUNICATION	35.00
MATHFLASH	885-8030-37	MSDOS	EDUCATION	20.00
ORBITS	885-8041-37	MSDOS	EDUCATION	25.00
POKER PARTY	885-8042-37	MSDOS	ENTERTAINMENT	20.00
SCICALC	885-8028-37	MSDOS	UTILITY	20.00
SKYVIEWS	885-3015-37	MSDOS	ASTRONOMY UTILITY	20.00
SMALL-C COMPILER	885-3026-37	MSDOS	LANGUAGE	30.00
SPELL5	885-3035-37	MSDOS	SPELLING CHECKER	20.00
SPREADSHEET CONTEST PACKAGE	885-3017-37	MSDOS	VARIOUS SPREADSHEETS	25.00
TREE-ID	885-3036-37	MSDOS	TREE IDENTIFIER	20.00
USEFUL PROGRAMS I	885-3022-37	MSDOS	UTILITIES	30.00
UTILITIES	885-3008-37	MSDOS	UTILITY	20.00
Z100 WORDSTAR CONNECTION	885-3047-37	MSDOS	UTILITY	20.00
ZBASIC DUNGEONS & DRAGONS	885-3009-37	MSDOS	GAME	20.00
ZBASIC GRAPHIC GAMES	885-3004-37	MSDOS	GAMES	20.00
ZBASIC GAMES	885-3011-37	MSDOS	GAMES	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	885-3016-37	MSDOS	GAME	10.00
ASSEMBLY LANGUAGE UTILITIES	885-8046-37	MSDOS	UTILITY	20.00
BOTH SIDES PRINTER UTILITY	885-3048-37	MSDOS	UTILITY	20.00
DEBUG SUPPORT UTILITIES	885-3038-37	MSDOS	UTILITY	20.00
DPATH	885-8039-37	MSDOS	UTILITY	20.00
FLIPFAST GUIDE TO GW-BASIC	885-4601	GWBASIC	MANUAL	20.00
FLIPFAST GUIDE TO MS-DOS	885-4602	MSDOS	MANUAL	23.00
FLIPFAST GUIDE COMBO OFFER	885-4603	VARIOUS	MANUALS	41.00
HADES	885-3040-37	MSDOS	UTILITY	40.00
HELP	885-8040-37	MSDOS	CAI	20.00
HEPCAT	885-3045-37	MSDOS	UTILITY	35.00
HUG BACKGROUND PRINT SPOOLER	885-3029-37	MSDOS	UTILITY	20.00
HUG EDITOR	885-3012-37	MSDOS	TEXT PROCESSOR	20.00
HUG MENU SYSTEM	885-3020-37	MSDOS	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PROD 1983 THRU 1985	9.75
HUGMCP	885-3033-37	MSDOS	COMMUNICATION	40.00
HUGPBBS SOURCE LISTING	885-3028-37	MSDOS	COMMUNICATION	60.00
HUGPBBS	885-3027-37	MSDOS	COMMUNICATION	40.00
ICT 8080 TO 8088 TRANSLATOR	885-3024-37	MSDOS	UTILITY	20.00
MAGBASE	885-3050-37	VARIOUS	MAGAZINE DATABASE	25.00
MATT	885-8045-37	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025-37	MSDOS	UTILITIES	20.00
REMARK VOL 5 ISSUES 48-59	885-4005	N/A	1984	25.00
REMARK VOL 6 ISSUES 60-71	885-4006	N/A	1985	25.00
REMARK VOL 7 ISSUES 72-83	885-4007	N/A	1986	25.00
REMARK VOL 8 ISSUES 84-95	885-4008	N/A	1987	25.00
SCREEN DUMP	885-3043-37	MSDOS	UTILITY	30.00
UTILITIES II	885-3014-37	MSDOS	UTILITY	20.00
PC Compatibles				
ACCOUNTING SYSTEM	885-8049-37	MSDOS	BUSINESS	20.00
CARDCAT	885-6006-37	MSDOS	CATALOGING SYSTEM	20.00
CHEAPCALC	885-6004-37	MSDOS	SPREADSHEET	20.00
CP/EMULATOR II & ZEMULATOR	885-6002-37	MSDOS	CPM & Z100 EMULATORS	20.00
DUNGEONS & DRAGONS	885-6007-37	MSDOS	GAME	20.00
EZPLOT II	885-6013-37	MSDOS	PRINTER PLOTTING UTILITY	25.00
GRADE	885-8037-37	MSDOS	GRADE BOOK	20.00
HAM HELP	885-6010-37	MSDOS	AMATEUR RADIO	20.00
KEYMAP	885-6001-37	MSDOS	UTILITY	20.00
LASERWRITER CONNECTION	885-8050-37	MSDOS	PRINTER UTILITY	40.00
PS's PC UTILITIES	885-6011-37	MSDOS	UTILITIES	20.00
SCREEN SAVER PLUS	885-6009-37	MSDOS	UTILITIES	20.00
SKYVIEWS	885-6005-37	MSDOS	ASTRONOMY UTILITY	20.00
TCSPELL	885-8044-37	MSDOS	SPELLING CHECKER	20.00
ULTRA RTTY	885-6012-37	MSDOS	AMATEUR RADIO	20.00

You've got a screen full of important technical data that would be nearly impossible to memorize, and you already have writer's cramps from the last screen full. With ***SCREENDUMP*** from HUG, you can reproduce a complete video screen on a dot matrix printer, including both text and graphics without having to exit the current program. *SCREENDUMP* supports most of the more popular dot matrix printers, including the newer 24-pin and laser jet models. The latest version of *SCREENDUMP* is ***HUG P/N 885-3043-37.***

"Thank Heaven for *HADES*!" That's what a lot of MS-DOS users are saying when ***HADES*** rescues a file that just got accidentally erased. Erased file recovery is only a small part of the capabilities of this program. *HADES* is *HUG's Absolute Disk Editing System*. Within the realms of MS-DOS, *HADES* allows you to directly edit any part of any disk. Directories, files, file attributes. FATS: nothing can hide from you when you use *HADES*. *HADES* works on *ANY* computer that can run MS-DOS version 2 or greater. Order ***HUG P/N 885-3040-37*** today!

Want to keep your H/Z-100? Want to run a lot of that good PC compatible software out there? Don't want to buy a PC compatible though? Then get ***ZPC II, HUG P/N 885-3037-37***, and the ***ZPC II upgrade disk, HUG P/N 885-3042-37***.

ORDERING INFORMATION

For VISA and MasterCard phone orders, telephone the Heath Users' Group directly at (616) 982-3838. Have the part number(s), descriptions, and quantity ready for quick processing. By mail, send your order, plus 10% postage and handling ($1.00 minimum charge, up to a maximum of $5.00) to: Heath Users' Group, P.O. Box 217, Benton Harbor, MI 49022-0217. VISA and MasterCard require minimum $10.00 order. No C.O.D.s accepted.

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

Continued from Page 6

lightful Fringes of the CP/M World." It looks like this is what this turned out to be. So, I feel like commenting on two more pieces of software from Spectre Technologies, the REMBRANDT and the PRESTO!

REMBRANDT does just what the advertisement says, it lets you draw anything, flawlessly and easily. Out of curiosity, I unearthed my ED-A-SKETCH of Software Toolworks, HDOS version mind you. My trusty HUG HTOC utility converted an HDOS ED-A-SKETCH picture into CP/M, and REMBRANDT's ED-A-SKETCH conversion utility printed it fine. I still have an MPI-99 printer and mentioned it to Spectre. There was no driver then for it, so I was using my Diablo 630 compatible printer. Two-three weeks later, I got for free a new MPI printer driver for the REMBRANDT!

Those who like a desktop utility would like PRESTO! The latest version occupies less memory, about 10K, because the various utilities stay on disk till called for. The Notepad is quite versatile, a miniature word processor. It is through this Notepad that you can create a Rolodex, appointment calendar and other simple databases. I bought it mainly for three reasons. One was access to CP/M commands while in programs without this facility. The second was key mapping. I like my trusty HUG KEYMAP, but I did not want to have both in memory at the same time, just to have access to CP/M. The third was the Programmer's calculator, but sorry to say, the decimal integer calculator part does not work. Spectre has been aware of it. I hope they correct it soon because in the CP/M world, otherwise, good and trustworthy names lose their credibility fast, if errors are not corrected or at least acknowledged.

To conclude, not everything is perfect even in the world of 8-bits, but almost so. Keep on pounding merrily on your 8-bit-Dinos (surnamed Saur?) friends! If Jim is not allergic to dust, he may uncover some more "archaeological finds" for us!

Products Discussed

HiSpeed C Library:

a. Reference manual with object code on disks approximately $30.00*.

b. Source code archives and utility sources on disks approximately $20.00*.

Labeler, manual, object and source code on disks approximately $50.00*.

* Prices are approximate, +− a few dollars, depending on disk format.

Viking C Systems
P.O. Box 9244
2243 Belaire Drive
Salt Lake City, UT 84109
(801) 466-6820

REMBRANDT
Business Toolkit ..... $39.95 + S/H
PRESTO!
Desktop Utility ..... $39.95 + S/H

Spectre Technologies, Inc.
22458 Ventura Boulevard, Suite E
Woodland Hills, CA 91364
(800) 628-2828 (ext. 918)

AUTOFILE (Z80 ONLY)
No. 885-1110 (HDOS) ... $30.00 + S/H
KEYMAP CP/M-80
No. 885-1230[37] ..... $20.00 + S/H
HTOC in UTILITIES
No. 885-1212-[37] ..... $20.00 + S/H

Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217

Sincerely,

Alkis J. Sophianolpoulos
2994 McCully Drive, N.E.
Atlanta, GA 30345

Obsolescence

Dear HUG:

I would love to see an article in REMark dealing with how people handle obsolescence. The turmoil affecting the "accepted state-of-the-art" have now apparently decreed that floppy disk drives and Epson printers are no longer with it. It's getting so that people with 640k are tempted to feel ashamed with being so underpowered. Daisy wheels are out — no graphics, so we are told. Remember the Z-100? asks one recent letter writer. Help! I still do almost all my word processing on a Z-89 with a single hard-sector drive.

That leaves me about a half dozen or more "waves" behind in the march of technology. I went from 1981 with the salesman claiming "you've got to have this . . . it's got CP/M" to 1988 where I feel like a country farmer with a Model A Ford.

Seven years ago, I felt the excitement of being a pioneer. Now I am a stick-in-the-mud. Still using my first computer after 7 years . . . have I gone from hacker to hasbeen so quickly? What will be the fate of today's MS-DOS key twiddlers? Should we get ready for the idea of MS-DOS graybeards?

Hanging onto an old system is like owning an Edsel and then getting AIDS. As we fogeys sit hunched over our screens, we face the agonizing realization that time may have passed us by. Is our fate to become hacker relics, talking a dying language to a relentlessly shrinking audience? Will we be mocked and shunned by our neighbors because we still use — HORRORS! — floppy disks? Will we be targeted for perdition because we refuse to participate in the latest upgrade extortion from our friendly software supplier? Should we feel humbled and emasculated because we still abide in the primitive provinces of monochrome and daisy-wheel printers?

Does IBM or Zenith or anyone else have the right to abandon us and leave us as orphans? When I bought my Z-89, I never thought of the difficulties which were well under way for the H-8 users and the HDOS zealots. By 1984, the MS-DOS tide was so overwhelming that HDOS and CP/M users were left high and dry. Now the MS-DOS users feel threatened because of the OS/2 hoopla of IBM.

So CP/M was crushed, with the machines and software relegated to closets and boneyards. A few zealots crusaded on, stubborn and unyielding. Like someone who doesn't want to give up on the memories of an old rattletrap car, they hung onto the faded past. Their world became smaller, more isolated, but also more stable. They could still do word processing and small spreadsheets. There was a certain relief from the terrible pressures of always keeping up with "what's new" and paying the financial price to stay abreast of the latest wave. There was a release from the temptation to follow the latest seductive fad, with the consequent shame and humiliation when the fad evaporated into ethereal inconsequence.

My obsolescence is so puzzling because it seems so rational. I see waves upon waves of Yuppies scrambling around trying to become Power Users and thrashing excitably and incoherently amidst a tangle of LANs, dBase hierarchies and downloadable fonts that won't download. Like IBM corporate executives from the distant

past, one can imagine them getting up in the morning and singing, "All I want for Christmas is an 800 meg hard disk." I sit here quietly poking away with my one-macro 64k Selectric keyboard clunker, with no reason to change. I do have an MS-DOS machine with a mushy keyboard which I use for spreadsheets, but no one is going to take away my puddle jumper.

I have chosen to partake of a voluntary vow of rational obsolescence. We are like the citizens who refused to accept the New Coke and the Edsel. We become the technological Amish, shunning the latest technological fads. From the shield of togetherness, we achieve sustenance and protection from the ravages of hucksters and peer pressure.

Many CP/M users remained productive, and there are numerous stories of books being written and sent by modem after coming from the innards of CP/M machines. For many of those old critters, sentimental love shrouded a certain logic: a true Selectric keyboard, with large letters on the keys and no CAPS-LOCK horrors. With DOS of only 5k and 64k RAM, many users had no choice except simplicity. They never had to undergo the exquisite excitement Symphony owners encounter when they are faced with a choice of over 600 commands. They never share the apprehension of the WordStar user who finds the latest upgrade includes 300 new commands — and comes on 21 disks. The CP/M user lives in a sublime tranquility by comparison.

For word processing it's just fine, and I can get at any printer code I want on any printer. That Selectric keyboard smiles at me every day. The CAPS LOCK key is in the right place and has a detent lock, the function keys are all in a nice row across the top (just like the latest IBM keyboards). The keys are black with nice large letters. The feel is super, and I have been four years without a service call. But it has also been over 4 years since I talked to anyone else who used a similar machine. Maybe there needs to be a new lonely hearts column in REMark to deal with the interests of ex-pioneers who turned patriarchs. Maybe even an orphan survey to find out who is still using their oldies, or if not, did they just stash it away for a rainy day or did it long ago go to the old happy hunting ground in the computer sky. It might be a fascinating study in hacker fogeyism.

Sincerely,

Stephen H. Kaiser, PhD
Traffic & Transportation Engineer
191 Hamilton Street
Cambridge, MS 02139

ED: I'm sure Stephen expresses the feelings of many huggies still using 8-bit computers. Perhaps we could start a "Lonely Hearts" column, but the articles would have to be submitted on MS-DOS formatted disks!

Fine Work

Dear HUG:

Thanks for your continuing fine work on REMark. It's still the magazine I look forward to receiving the most each month, including Playboy (either I have my priorities mixed up or I've been married too long). Please convey my thanks to Pat Swayne also, for his never-ending string of fine programs. If there were an award for Longest-Running Wizard/Guru/General-purpose Resident Brilliant Person, he'd have won it.

Yours truly,

Mark Van Sickle
1115 N. Altadena
Royal Oak, MI 48067

HUGMCP V1.10 On An H-100 With DOS V3.10

Dear HUG:

Just started using it to talk to my EPROM programmer with no problems. Set up was smooth with no surprise.

Next was a borrowed UDS 212A modem. Plugged in a 7-wire cable (2-3, 5-8, 20) and it worked first time.

Then I swapped the UDS for an old Hayes Smartmodem 300 and the surprises began. I started with the same cable as above. I got no echo from the modem and the computer locked up. After some fooling around with a breakout box, here's what I found.

1. You should not tie pins 5 & 6 of the H-100's J2 together as diagrammed on page 5. The RS-232-C "standard" for a DTE port is that pins 5 & 6 are both inputs. If they are active, but unused, they should be tied to either a high or low pin to eliminate surprises, 5 should be high and 6 low. (In reality, they can usually be left floating.) In my case, pin 5 is floating at about +6V and pin 6 is at 0V. Tying them together pulls pin 6 high enough to lock the system. Something in either MCP or DOS V3.1 is looking at pin 6 and doesn't like it pulled high. So far, I've had no problem leaving them floating.

2. You should not use pin 8 in your cable. On page 6, the documentation says that MCP ignores pin 8, but on my system I get the same lock up if pin 8 is connected. Again, something is looking at pin 8 and responding to it. If the documentation is right, and MCP is truely ignoring pin 8, then DOS is the culprit. I find nothing in DOS V3.1 CONFIGUR that gives me any options concerning pin 8.

On top of all that, the Hayes book never states that you must talk to the modem in upper case only. Moral: When in doubt, CAPS LOCK!

Once I figured out these problems, (it's always harder when you have three things wrong simultaneously) everything started flying. HUGMCP is very easy to use and has all the features I need now. Thanks to HUG for another good, inexpensive program.

John Burch
136 Arlington
Elmhurst, IL 60126

Patch for MASM V4.0 Problem . . .

Dear HUG:

In the September issue of REMark, I mentioned that a patch was available for a MASM V4.0 problem dealing with multiple end-of-file characters that are on INCLUDE files. Since that note appeared, I have received a number of requests for the patch. I am including the patch here for the benefit of all HUG members who have MASM V4.0.

The patch should always be made first to a copy of MASM.EXE, then, and only then, should it be made to your production version. Also, please note that changes cannot be made to files loaded with the "EXE" extension, so your copy of MASM .EXE has to be renamed with another extension first. The Symbolic Debug Utility, SYMDEB, is used for the patch.

```
COPY MASM.EXE MASM.SAV
REN MASM.EXE MASM.FIX
SYMDEB MASM.FIX

-A 72B8          (assemble code at 72B8)
```

Please note that the segment and offset values are put out by the debugger and all you have to do is enter the assembly code to the right of the segment/offset addresses.

```
1476:72B8    MOV  BX,[09D6]
1476:72BC    MOV  BYTE PTR [01C0],00
1476:72C1    JMP  7539
1476:72C4    LES  DX,[BX+06]
1476:72C7    INC  BYTE PTR [01C0]
1476:72CB    JMP  7542
1476:72CE    CMP  BYTE PTR [SI-01],1A
1476:72D2    JZ   72D7
1476:72D4    JMP  7574
1476:72D7   DEC  SI
1476:72D8    JMP  72CE
1476:72DA       <RETURN> by itself

-A 7535

1476:7535    JMP  72B8
1476:7538       <RETURN> by itself

-A 753F

1476:753F    JMP  72C4
1476:7542       <RETURN> by itself

-A756D

1476:756d    JMP  72CE
1476:7570       <RETURN> by itself

-W             (writing 14E3E bytes)

-Q             (quit)

REN MASM.FIX MASM.EXE
```

Now try it. If you have a problem, start the whole procedure over; possibly a HEX number was entered incorrectly in one of the jump instructions. Good Luck . . .

Rich Mueller, Ph.D.
11890-65th Avenue N.
Maple Grove, MN 55369

HELP! Need CRASH Program for HDOS

Dear HUG:

I have a Heath H-89 with 2 disk drives and use a hard-sector controller and HDOS. I recently had some drive problems which caused several of my floppy disks to be "corrupted".

I found an ad in some of my old papers for a program called "CRASH" from Software Wizardry which is supposed to allow the recovery of information off unmountable HDOS hard-sector disks. I called Software Wizardry, but they don't have any more copies of the program on HDOS hard-sector disks and no longer have any equipment to make them. They could make me a soft-sector copy, but I would still have to find someone who could convert it to a hard-sector HDOS format.

I would like to buy your copy of CRASH (or a similar program) with all the documentation that goes with it. I lost some important files on some of those disks. Any help would be greatly appreciated. Please write if you can help in any way.

Thanks in advance,

Luther Miller
P.O. Box 118
Reeves, LA 70658

Installing an Irwin 420 Streaming Tape Drive into a ZW-148-42

Dear HUG:

The old addage, "It's not IF your computer is going to lose data . . . it's WHEN!" is a given. Few statements could be more correct, and ignoring the wisdom of this proverb can be incredibly costly. Once I had acknowledged the painful reality of this version of the infamous Murphy's Law, my next course was to seek out a way to back up my Winchester without having to use many floppy disks. (20MB divided by 360K is 55.56 — costly!!!)

I keep all of my system software on the Winchester and work on the floppy disks. Over the course of the year since I bought my computer, I had backed up my Winchester several times, each time used lots of floppy disks. The procedure is time consuming, and the time and floppy disks could be put to other uses.

One method used by industrial and professional computer installations to back-up Winchesters is the use of streaming tape drives. Seeing no mention of streaming tapes in the Heath/Zenith catalogs, I called the Heath/Zenith dealer in my area (3606 W. Dempster, Skokie, IL 60076) to see if they carried such a device. The salesman, Chris Bielawski, recommended an Irwin International 420 External 20MB streaming tape backup system (designed for use with IBM XTs).

My ZW-148-42 comes with a factory-installed 20MB Winchester hard disk drive and a 5-1/4", 360K floppy disk. It has been expanded to the full 640K RAM (two Z-205-4 256K memory upgrade kits) and includes the 8087 math coprocessor (Z-316-8). A Zenith ZVM-1230-A monochrome (green) monitor on a ZVM-1200-1 tilt/swivel base, a Zenith ZM-2401 2400 (adjustable) baud modem, and an Epson FX-86-E printer round out the peripherals.

Because the 20M byte Winchester is factory installed, a daughter board already existed in the computer, meaning I didn't have to purchase a daughter board for the installation.

The "fun" began when I found out that the Irwin tape drive needed two slots, one for the data plug and one for the power connection. Unfortunately, the ZW-148-42 was designed with only one slot for expansion. I had just encountered the first of several problems.

I solved this first problem by using a nibbler to make a slot big enough to pass the wires for the power connection in the back aluminum wall, just above the nameplate. Several pieces of electrical tape ensured that constant use would not let the new slot's edge surfaces cut into and, consequently, short out the power connections. A cable tie enabled me to strain relieve this jury-rig.

My problems certainly didn't end here. According to Irwin's instructions, the tape controller card has to be inserted in the wiring between the floppy disk drive and the floppy disk controller card. After installing the Irwin controller card in my only expansion slot, I discovered that the flat cable used by Zenith to connect the floppy disk controller card to the floppy disk drive wouldn't connect to the Irwin tape controller card. This was because the floppy disk drive had an edge connector while the Irwin tape controller card was designed to accept a 34-pin connector.

Discussions with the salesman, Chris Bielawski and Tim Novak, a Heath/Zenith technician, lead to Novak fabricating a ribbon cable with a 34-pin connector at one end and an edge connector at the other. Novak's adaptor cable connected the floppy disk's edge connector to the tape controller card's 34-pin port.

Irwin supplied a flat cable to connect the tape controller card to the floppy disk controller card. I removed the now useless flat cable that Zenith used to connect the floppy disk drive and the floppy disk

controller and discovered that the Zenith floppy disk controller card had all 34 pins. Since Irwin's flat cable had polarized (one pin position blocked off) 34-pin connectors at both ends, I had to replace one polarized connector with a new 34-pin connector. The modified cable went between the floppy disk controller card and the tape controller card.

The tape drive worked on the first try and my Winchester was backed up within minutes.

Rethinking the whole episode, an easier way becomes obvious. Replace the edge connector on the Zenith flat cable that went between the floppy disk drive and the floppy disk controller card with a 34-pin connector and plug it directly into the Irwin tape controller card. Then, replace one of the 34-pin connectors on the Irwin-supplied flat cable with an edge connector and use it to connect the tape controller card to the floppy disk drive. Assuming the proper polarities are observed, the installation will be quicker and use fewer parts.

Henry M. Morris
7231-A N. Campbell
Chicago, IL 60645-1423

No Cross Reference?!

Dear HUG:

I was very surprised and disturbed when I searched my January 1988 'REMark' for the Cross Reference and Index of 1987 Articles of 'REMark'. I just happened to be looking for an article on batch programming that I had seen in one of the issues. As it turns out, the issue containing the article was the June of 1986 issue. I was able to find it because of the cross reference of 1986 issues in the January 1987 issue of 'REMark'. I am glad that the cross reference of articles was still available in the earlier years.

As you can see, I would have had one heck of a time finding the issue that I sought. I don't know how others feel, but I depend very much on the Cross Reference that you've provided in the past. I have very limited time available to read issues from cover to cover and often times will not even look at the issue until months later, if at all. I like the convenience of the Cross References since it allows me to look for all articles that apply to subjects of interest, such as: PC compatibles, MS-DOS applications, hardware and so forth.

I would very much like to see you continue to provide this valuable resource and wonder how other members feel. If it comes to saving space for other items in the January issue, I say forget the other items because nothing is more important to me than the Cross Reference of Articles and issues of the prior year.

Sincerely,

Walter Jenkins
6640 N. Braeburn Lane
Glendale, WI 53209

WordStar's Shorthand

Dear HUG:

WordStar Release 4.0 Shorthand menu has a "today's date" and a "current time" macro. I wanted a STANDARD date like March 16, 1988, and a military date like 16 March 1988. I sacrificed the "current time" macro. Here's how:

1. Enter WSCHANGE WS.EXE.
2. Go through the menus D C D.
3. At the Shorthand menu, choose C to change "today's date" or leave it STANDARD.
4. At the Shorthand menu, choose D to change "current time" as if it was a date. Enter 1 3 4 for MILITARY date.
5. Exit WSCHANGE.

That's it!

If you want to modify WSMSGS.OVR SHORTHAND MENU to read "standard date" and "military date", use a copy of WSMSGS.OVR and try the patches that follow. If these fail, you have a different version of WSMSGS.OVR (2-23-87, 12:00p) than I have. Use DEBUG to search for your starting addresses and substitute them in the patches given. (The labels, "standrd date" and "militry date", must be exactly twelve characters long!)

Using PATCHER.COM (from ZPC 2 disk) and a copy of WSMSGS.OVR, add the following to PATCHER.DAT and run PATCHER:

```
WordStar rel 4.0 Shorthand Menu
Insert disk containing WSMSGS.OVR.
WSMSGS.OVR
3BE6,73,74,61,6E,64,72,64
3C30,6D,69,6C,69,74,72,79,20,64,61,74
z
```

Using DEBUG and a copy of WSMSGS .OVR . . .

1. Edit address cs:3CE6 to 73 74 61 6E 64 72 64 replaces "today's" with "standrd").
2. Edit address cs:3D30 to 6D 69 6C 69 74 72 79 20 64 61 74 (replaces "current tim" with military dat").
3. Write out the patched file.

Using DEBUG to search for the correct addresses . . .

1. Search for "today's date" and note the address. s cs:0 0 74 6f 64 61 79 27 73 20 64 61 74 65
2. Substitute this address for 3CE6 in the DEBUG patches or subtract 100h from the address and substitute the result for 3BE6 in PATCHER.DAT.
3. Search for "current time" and note the address. s cs:0 0 63 75 72 65 63 74 20 74 69 6D 65
4. Substitute this address for 3D30 in the DEBUG patches or subtract 100h from the address and substitute the result for 3C30 in PATCHER.DAT.
5. Try the patch with the new addresses.

Sincerely,

Stephen A. Jacob
103 B 4th Street
Honolulu, HI 96818

Current HUG Club Information

Dear HUG:

Please change the information on the San Antonio (SAHUG) Heath Users' Group from Group size of 65 to group size of 140, change the contact person to David Smith and the meeting date to the first Wednesday of the month at the Heath/Zenith Electronics Center at 7:30 pm.

Thanks,

David A. Smith
SAHUG Secretary/Treasurer

*

The other cats get to sing along!

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

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

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

HEPCAT gets along with everyone . . .

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

HEPCAT works harder . . .

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

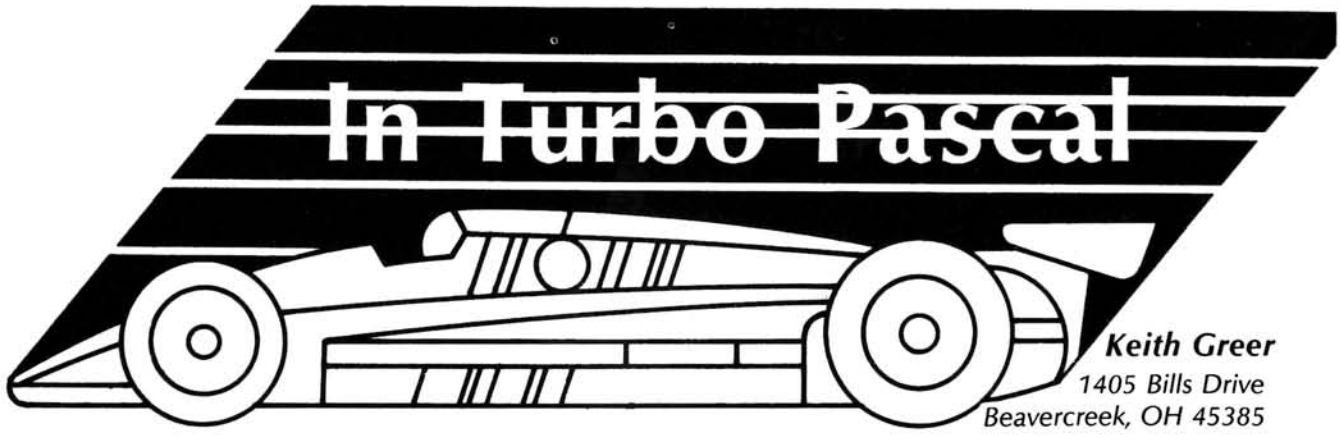
The HEPCAT floating point calculator supports 8 significant digits and can display numbers four ways: floating point, fixed point, scientific notation, and engineering notation. Numbers are handled internally in BCD format to eliminate binary round off errors in addition and subtraction.

HEPCAT eats less . . .

HEPCAT uses less than 16k of memory — less than any other pop-up calculator that we know of. It also uses less than 16k of disk space, so you don't have to worry about where to put it on a small system. The HEPCAT window uses less screen space, too. It shows you more real information than other pop-up calculator displays, but it doesn't waste space by showing you a keypad layout. You already know what your keypad looks like! HEPCAT is easier to learn, too, with commands that make sense.

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

Clearing (Less Than) The Screen In Turbo Pascal



Keith Greer
1405 Bills Drive
Beavercreek, OH 45385

Ever try to write a program that uses formatted screen data entry and wish you had an easy way to clear just a portion of the screen without disturbing the rest? This happened to me recently while I was working on a specialized database application. I had dedicated the left side of the screen to certain information which I wanted to clear and update independently of the right.

What do you do in this situation? If you're like me, you probably resorted to writing spaces over the area you wish to clear. This method is painfully slow for larger areas, and just doesn't look very professional. If you are a sophisticated programmer and you have a PC, you may have used the video interrupt to accomplish the task. But what if you have a late, great, Z-100? If you program in Turbo Pascal Version 3, help is here! I offer you the results of my many hours of struggling with a way to handle this problem: Procedure ClrBox.

This article assumes you are reasonably familiar with Turbo, but no detailed technical knowledge is required to use ClrBox.

Using The Z-100 Version

Listing 1 contains the commented inline code so that the more technically inclined can see what's going on. You don't need all of the comments of course, so if you type in Listings 2 and 3, you'll be ready to use ClrBox. This procedure gets its speed by writing directly into video memory, so you must enable writing to video memory first by calling Init_Video (Listing 3). Only one call is required, usually at the beginning of your main program. It is helpful to define the Z-100 colors in your declarations, thus:

```
const
   Black = 0;
   Blue = 1;
   Red = 2;
   Magenta = 3;
   Green = 4;
   Cyan = 5;
   Yellow = 6;
   White = 7;
```

Generally, you will want to set your text background color to the same color passed to ClrBox before writing in the area. This can be accomplished using the Esc'm'fore,back sequence described in your Z-100 manual. For example:

```
Write(#27'm',Green,Black);
```

would set the foreground to Green and the background to Black.

Using The PC Version

This is one area where the PC has it over the Z-100. The PC provides a built-in method for clearing windows on the screen. This version of ClrBox takes advantage of that capability. Listing 4 is again for the curious, while Listing 5 contains just what you need to get going. Since the colors are already defined in the PC version of Turbo Pascal, you don't need to have a const declaration for them in your program. See your Turbo manual for the available color names. The PC sets an "attribute byte" for the cleared area which defines both foreground and background colors. I wanted to use the same calling syntax for both the PC and the Z-100, and the only color passed to ClrBox is the background color. Therefore, I chose to have the routine extract the foreground color for the cleared area from the color of the character at the cursor position when ClrBox is called. When you write something in the area, Turbo will write it in the foreground color you last selected using the TextColor procedure anyway. Generally, you will want to call Turbo's TextBackground procedure to set the background color to the same color passed to ClrBox before writing in the area.

General Usage

A sample call to ClrBox might look like:

```
ClrBox(1,10,35,22,Blue);
```

This sets the rectangle with (1,10) at the upper left and (35,22) at the lower right to the color Blue. The upper left corner of the screen is (1,1). If the rectangle is already blue, any text within it is erased and the background is unchanged. The passed values need not be constants. They

Listing 1

```
Procedure ClrBox(X1,Y1,X2,Y2,Color : integer);
{
      ************** Z-100 Version *************
Fills a rectangular area of the screen with Color.  Parameters
are CHARACTER coordinates, X1,Y1 is upper left corner and X2,Y2
is lower right corner of the area to be cleared.  1,1 is the
upper left corner of the screen.  This procedure is extremely
fast, and is intended to serve primarily with the color black (or
current background color) as a means to clear only a portion of
the screen.  Color is written directly into video memory, so
video write must be enabled via procedure Init_Video (or
equivalent) prior to calling this procedure.
}

begin {ClrBox}
inline(

$8B/$96/X2/    {       MOV     DX,[BP+0FFSET X2]     ; Get X2                  }
$2B/$96/X1/    {       SUB     DX,[BP+0FFSET X1]     ; -X1                     }
$42/           {       INC     DX                    ; DL has # columns        }
$8A/$B6/Color/ {       MOV     DH,[BP+0FFSET COLOR]  ; DH has color            }
$8B/$9E/Y1/    {       MOV     BX,[BP+0FFSET Y1]     ; Get Y1                  }
$4B/           {       DEC     BX                    ; make 0-based            }
$B1/$0B/       {       MOV     CL,11                                           }
$D3/$E3/       {       SHL     BX,CL                 ; * 2048                  }
$03/$9E/X1/    {       ADD     BX,[BP+0FFSET X1]     ; + X1                    }
$4B/           {       DEC     BX                    ; make 0-based--> VRAM offset }
$8B/$8E/Y2/    {       MOV     CX,[BP+0FFSET Y2]     ; Get Y2                  }
$2B/$8E/Y1/    {       SUB     CX,[BP+0FFSET Y1]     ; - Y1                    }
$41/           {       INC     CX                    ; CX has # character rows to do }
$51/           { CLB1: PUSH    CX                    ; Save row counter        }
$B9/>$9/       {       MOV     CX,9                  ; Put in scan lines/row   }
$8B/$FB/       {       MOV     DI,BX                 ; DI has scan line offset }
$B8/$00/$C0/   { CLB2: MOV     AX,0C000H             ; Blue plane              }
$8E/$C0/       {       MOV     ES,AX                                           }
$51/           {       PUSH    CX                    ; Save scan line counter  }
$B9/>$3/       {       MOV     CX,3                  ; Put in # of color planes }
$8A/$C6/       {       MOV     AL,DH     ; AL has color (only right 3 bits used) }
$D0/$C8/       { CLB3: ROR     AL,1      ; move color bit to high bit position   }
$98/           {       CBW                           ; replicate it thru AH    }
$86/$C4/       {       XCHG    AL,AH                 ; swap AL<-->AH           }
$51/           { CLB4: PUSH    CX                    ; Save the color plane counter }
$57/           {       PUSH    DI                    ; Save the scan line offset }
$8A/$CA/       {       MOV     CL,DL                 ; Put in # of columns     }
$F3/$AA/       {       REP     STOSB                 ; and fill the scan line  }
$86/$C4/       {       XCHG    AL,AH                 ; swap AL<-->AH           }
$5F/           {       POP     DI                                              }
$59/           {       POP     CX                                              }
$8C/$C6/       {       MOV     SI,ES      ; Increment the color plane segment  }
$81/$C6/$00/$10/ {     ADD     SI,1000H                                        }
$8E/$C6/       {       MOV     ES,SI                                           }
$E2/$E7/       {       LOOP    CLB3                  ; Loop over all 3 color planes }
$59/           {       POP     CX                    ; Get scan line counter back }
$81/$C7/>$80/  {       ADD     DI,128                ; DI-->next scan line     }
$E2/$D5/       {       LOOP    CLB2                  ; Loop over this character row }
$59/           {       POP     CX                    ; Get character row counter back }
$81/$C3/$00/$08/ {     ADD     BX,2048               ; BX-->next character row }
$E2/$C8);      {       LOOP    CLB1                  ; Loop over all character rows }

end; {ClrBox}
```

Listing 2

```
Procedure ClrBox(X1,Y1,X2,Y2,Color : integer);

{ ************** Z-100 Version ************* }

begin {ClrBox}
inline(

$8B/$96/X2/$2B/$96/X1/$42/$8A/$B6/Color/$8B/$9E/Y1/$4B/
$B1/$0B/$D3/$E3/$03/$9E/X1/$4B/$8B/$8E/Y2/$2B/$8E/Y1/$41/
$51/$B9/>$9/$8B/$FB/$B8/$00/$C0/$8E/$C0/$51/$B9/>$3/
$8A/$C6/$D0/$C8/$98/$86/$C4/$51/$57/$8A/$CA/$F3/$AA/
$86/$C4/$5F/$59/$8C/$C6/$81/$C6/$00/$10/$8E/$C6/$E2/$E7/
$59/$81/$C7/>$80/$E2/$D5/$59/$81/$C3/$00/$08/$E2/$C8);

end; {ClrBox}

******** Before calling ClrBox on the Z-100, call: *************
```

Listing 3

```
Procedure Init_Video;
{Enables writing to video memory on the Z-100}
const
  vcr=$D8;
var
  old_vcr :integer;

begin
  old_vcr:=port[vcr];
  port[vcr]:=old_vcr and $7F;
end;
```

For you PC programmers, the equivalent is:

Listing 4

```
Procedure ClrBox(X1,Y1,X2,Y2,Color:integer);
{
  PC procedure to clear a window on the screen.  Parameters are CHARACTER
  coordinates,  1,1 is the upper left corner of the screen, 80,25 is
  the bottom right.  Sets Color as the background color of the cleared
  window.  Uses the foreground color of the current cursor position in
  the cleared window.
```

```
}
begin {ClrBox}

inline(

$B4/$0F/        {MOV    AH,15           ;Get video state                }
$CD/$10/        {INT    10H             ;BH has display page            }
$B4/$08/        {MOV    AH,8                                            }
$CD/$10/        {INT    10H       ;AH has attribute at current position }
$80/$E4/$0F/    {AND    AH,0FH          ;Strip off the background       }
$8A/$BE/Color/  {MOV    BH,[BP+ OFFSET COLOR];Get the background color  }
$B1/$04/        {MOV    CL,4            ;And shift it into position     }
$D2/$E7/        {SHL    BH,CL }                                         }
$0A/$FC/        {OR     BH,AH           ;Set the foreground bits        }
$B8/$00/$06/    {MOV    AX,600H         ;Clear window function          }
$8A/$8E/X1/     {MOV    CL,[BP+ OFFSET X1] ;Load the window boundaries  }
$8A/$AE/Y1/     {MOV    CH,[BP+ OFFSET Y1]                              }
$8A/$96/X2/     {MOV    DL,[BP+ OFFSET X2]                              }
$8A/$B6/Y2/     {MOV    DH,[BP+ OFFSET Y2]                              }
$FE/$C9/        {DEC    CL              ;Zero based positions           }
$FE/$CD/        {DEC    CH }                                            }
$FE/$CA/        {DEC    DL }                                            }
$FE/$CE/        {DEC    DH }                                            }
$CD/$10);       {INT    10H             ;Do it                          }

end; {ClrBox}
```

Listing 5

```
Procedure ClrBox(X1,Y1,X2,Y2,Color:integer);

{ ******************** PC Version ****************** }

begin {ClrBox}

inline(

$B4/$0F/$CD/$10/$B4/$08/$CD/$10/$80/$E4/$0F/$8A/$BE/Color/
$B1/$04/$D2/$E7/$0A/$FC/$B8/$00/$06/$8A/$8E/X1/$8A/$AE/Y1/
$8A/$96/X2/$8A/$B6/Y2/$FE/$C9/$FE/$CD/$FE/$CA/$FE/$CE/
$CD/$10);

end; {ClrBox}
```

may also be integer (or byte) variables. It's as simple as that! I hope some of you will find this useful. I know I have.

Other Routines Available

ClrBox is also available as an EXTERNAL routine, along with a complete set of external hand-optimized Assembly Language graphics routines for the Z-100 using Turbo Pascal. I call the package ZGRAPH. Some of you may have seen it mentioned in Paul Herman's newsletter. ZGRAPH provides all of the graphics features of the PC version of Turbo (except Turtle Graphics) and then some! The Z-100 operates in native mode, no special raster modes are used, no patching of Turbo is required. ZGRAPH adds only about 3K to your program. All or part of the screen may be saved and edited later with Paul Herman's fine Doodler V program, or pictures may be developed with Doodler and used in your own program. Because the code is external, the compiler has little work to do and compilation is very fast, which is one reason you bought Turbo in the first place, right?

Send me $25.00 and I'll send you:

1. ZGRAPH complete with fully commented Assembly source code and ready-to-print formatted on-disk documentation.
2. A demo illustrating how to use the routines. (With source, of course.)
3. XYPLOT, a fully functional X-Y data plotter I developed using ZGRAPH. (Again, with source and on-disk documentation.)

*

On The Leading Edge

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

Computer Problems, Programming in BASIC and C, Detecting Viruses, Preventing Viruses

One of my friends recently asked me: "Don't you ever get tired of all the computer problems?" The context of the question was based on our discussion of my having a power supply failure and a hard disk failure within just a few weeks. Since he really has no particular preference for a specific brand of computer, he also wondered if I wasn't having more problems with my Heath (Zenith) systems than were normal. Actually, I never considered that question in that context before, so I suppose it is reasonable to take a more detailed look at it.

Computer Usage and Failures

In order to place the question into some kind of perspective, I think it is important to know how a computer is used. At this point, my '248 is my primary production writing system, and it gets a LOT of use. As a minimum, I usually work six 10-hour days, and my computer is powered-on for at least 60 hours a week. If I take two weeks' vacation, that means a total of 50 weeks or a minimum of 3,000 hours of power-on time a year. In reality, I suppose the actual number is higher because there are many days when I work more than 10 hours, and I sometimes do a considerable amount of writing on Sunday, too. Let's say that, on the average, my system is powered-on for 3,500 hours a year.

I've had my '248 for about two years or about 7,000 hours of use. And, as reported last time, the power supply failed because it was furnishing nearly 22 volts on the 12 volt line. I have since been told that that particular type of failure is impossible with this kind of power supply design, but the fact remains that it did happen on my system. You will probably recognize that this is a direct application of Murphy's Law — If it can go wrong, it will.

Some people might say that a power supply should last longer than that, but I think that is reasonable (but not outstanding) performance from that particular power supply. Heat is a particular enemy that must be considered in a computer, and I don't think this power supply failure was related to poor design or anything like that. During most of that time, that power supply was used to drive two hard disks, two floppy drives, and a variety of add-in boards; and I suspect that configuration is rather uncommon to the extent that I placed more of a load on the power supply than the "average" configuration.

In short, I don't believe that there is any particular problem with the power supply, and I consider that as a maintenance item, much like the transmission on a car. If you don't believe that, perhaps you should look at some of the other popular computer magazines to see advertisements for replacement power supplies for IBM computers. Unfortunately, replacement power supplies for Zenith computers are significantly more expensive, and the IBM version will not fit in a Zenith computer. As I've mentioned before, some of the Zenith prices for add-ons and replacement parts are outrageous. For example, the new power supply for my '248 made a considerable dent in $200 (exchange).

Since the only mechanical part of the power supply is the fan, the failure I had was clearly somewhere in the electronics. Perhaps I didn't keep the house air conditioning low enough during some of the 110+ degree Texas summer days because of energy conservation — I have a ceiling fan in my study to help circulate the air so

I can leave the thermostat at 78. And there may have been enough heat from all components in my system to make the power supply fail prematurely. It is precisely for this reason that most mainframe computer rooms are actually cold, and many of them set the air conditioning for 68 degrees to minimize heat-related computer failures.

Mechanical Failures

For the most part, I would expect mechanical items on my computer to fail before the electronics, assuming that heat is not a problem. For example, I had to have a 5.25" 360K floppy drive "fixed" because it suddenly went bad with no warning at all. I found this out by accident when I sent an article to Kit Builders' Journal a couple of years ago, and Rick Simpson (Editor of KBJ) called to tell me that he received a blank disk. Floppy drives need to be aligned occasionally, and that was a part of the problem with the drive.

When you have a hard disk, the question is not IF the hard disk will fail; the question is WHEN. As a Boy Scout, I learned to "Be Prepared", and I have had enough experience with hard disks on mainframes and micros to follow that rule. That's why I continue to stress the importance of backups. For one reason or another, I have never had too many backups. Aside from disk failures of one kind or another, you never know when computer gremlins will cause you to make a stupid mistake like typing "DEL *.*" for the wrong disk drive or subdirectory. This is the primary reason that it is also important to make backups for floppy disk systems, as well as a hard disk.

The good news is that my 40 MB hard disk is repaired, and it was significantly less than the $200 I expected. The actual cost was just over $80, but I don't know exactly what was repaired or replaced. If you need to have a hard disk repaired, the best plan is to expect $200 or so, and then be pleasantly surprised if it is less. Sounds like taking your car in for some repair, doesn't it?

I have seen technical information for hard disks, and many of them seem to be rated at least 20,000 hours MTBF (Mean Time Between Failure). Most of you probably know that the "mean" is different from the "average", so I will not go into that discussion. And most of you will probably not have problems with your hard disk failing, simply because your system does not get as much use as mine does.

In the International Mail

I recently received a letter from Carlos Rashid (Lima, Peru) who asked about using the ANSI driver (e.g., ANSI.SYS) with BASIC programs. As most of you know, the ANSI driver defines standard ANSI Escape sequences that can be used to clear the screen, position the cursor, and so on. Carlos notes that he has written some programs for his H-89 using the H-19 Escape sequences with BASIC-80. More importantly, Carlos uses a good programming technique and defines these control sequences at the beginning of the program. The question is: "How can you use the ANSI driver Escape sequences with GW-BASIC or QuickBASIC on an AT compatible?"

The answer is that the ANSI device driver was not designed to work directly with BASIC. When I was doing the technical editing on the GW-BASIC FlipFast book, I tried some of the more obvious conversion techniques for this because it seemed like an easy way to convert programs from the Z-100 to my '248. The most obvious is to use the PRINT CHR$(27)... to "print" the various ANSI Escape sequences, such as highlighting to the screen. That does not work.

When you want to perform functions that are usually available in the ANSI driver, you must normally use the standard BASIC statements. For example, highlighting can be turned on by using the COLOR 0,7 statement and turned off with the COLOR 7,0 statement. Positioning the cursor is performed with the LOCATE statement, and you can erase to the end of the line by using PRINT with the appropriate number of spaces. And of course, CLS clears the screen. All of these BASIC statements are specifically provided for screen control and display. In short, BASIC provides the general screen control functions, but they are not implemented so that the ANSI driver can be used in the normal way. That leads to a more interesting question.

Why can't BASIC normally use the ANSI sequences in a PC or AT compatible system? The answer is that BASIC is "self-contained", and it provides all of the screen control as part of the language. In other words, BASIC bypasses many of the "general" DOS programming techniques (i.e., functions) since the BASICA interpreter handles most of that based on what statement is used. In particular, the interpreter handles virtually all of the screen input/output (I/O) which is part of the reason that some of the graphics-related commands are different in BASIC-80 compared to GW-BASIC (or IBM BASIC). The real reason for the differences in the commands (and the interpreter) is due to hardware differences, such as between the H-89 and the IBM PC or AT. That also explains why the BASIC-80 interpreter won't run on a PC compatible system and vice-versa.

Although many other languages use the DOS function calls that are part of the operating system, BASIC essentially bypasses DOS and performs its own I/O for screen and file handling. Because BASIC effectively bypasses DOS functions, it ignores the ANSI driver because it does not even know that it is installed.

The only way around that problem seems to be to write a screen handling subroutine in another language, such as assembler, and call that subroutine for any screen handling function to avoid the PRINT command in BASIC. At best, this approach is very clumsy, and it requires that you know another programming language in order to do it. From a design perspective, that is really not an unreasonable requirement for BASIC because it was developed to be "self-contained". Unfortunately, little thought was given to compatibility considerations, such as using the ANSI driver.

I have not found any BASIC compiler directive or interpreter command that will allow you to use DOS extensions, such as the ANSI driver, in a BASIC program. Since a lot of the older hardware (e.g., the H-89) and the BASIC interpreters for them have been discontinued, I consider it highly doubtful that anyone will "fix" this problem.

Converting programs from the H-89 or Z-100 BASIC is not particularly difficult, but it is time consuming. Program conversion requires considerable familiarity with both BASIC versions, as well as the hardware differences between the systems. For example, the Z-100 can only use eight colors while a PC compatible can use 16. This kind of thing can cause serious problems for beginning programmers who are not aware of these significant hardware differences between computer systems.

I am reminded of a user complaint about the Ecosoft C Compiler that one of the example programs (COLORTST.C, I think) would not run on a Z-100. That particular program does not run on a Z-100 because it relies on the special screen functions available on a PC compatible. The mascr and sascr library screen functions were clearly written for a PC-type computer, and a quick look at the required header (COLOR.H) used for those functions will verify this. Unfortunately, you must already know some details about the Z-100 to the extent that colors are handled differently even though the function summary on page 8-7 of the documentation clearly indicates these functions in a list which mentions that they may not be portable. When in doubt, read the manual.

Although the example program will compile just fine on a Z-100, my recollection is that execution of the program causes the famous "Wild Interrupt" error message on that computer. When I checked that at one point, I seem to recall that the Ecosoft function uses the INT 10H (Video I/O) interrupt that is not available on a Z-100 and results in the Wild Interrupt error. Once you recognize what happened, then you can proceed to write a special Z-100 C function which would probably use the Ecosoft sysint21 library function. By the way, that approach is strictly a first guess, and I have not actually tried it.

This situation illustrates a very important point about programming languages. In most cases, learning the language itself is not enough — you must also know something about the hardware in your computer and to some extent how it works. BASIC is one exception to that, simply because it is self-contained. Other programming languages, such as C and assembler, are not as simple. Assembler in particular is especially unforgiving in this respect.

Choosing a Programming Language

Most of you probably know by now that C is my favorite programming language, and there is a good reason for that. I frequently need to write short programs that get down to the hardware or disk level in my system. By its nature and design, BASIC is particularly limited, and it cannot easily do the things that I want to do.

For example, I was having a printer problem that seemed to be something related to communications with the parallel port on my '248. I suspected that there was a problem with Zenith MS-DOS or perhaps the ROM, since I was occasionally getting an error message that the system was unable to write to the PRN device. So I wrote the C program shown in Figure 1 to initialize and check the status byte for the parallel port.

```
/* pinit.c - initialize and display parallel status byte */
/* by: William M. Adney */
/* Copyright (C) 1988 by William M. Adney */

#include <stdio.h>
#include <dos.h>

int main()
{
    struct HREG ireg, oreg;

    ireg.ah = 1;        /* to initialize */
    ireg.al = 0;
    ireg.dh = 0;        /* dx is zero for printer 0 */
    ireg.dl = 0;

    sysint(0x17, &ireg, &oreg); /* for parallel port */

    printf("\n\nStatus byte in hex is: %X\n", (oreg.ah & 0xff));

/*  END OF PROGRAM */
}
```

Figure 1
PINIT.C to Initialize and Check Parallel Port

For those of you unfamiliar with C programming, I have used interrupt 17H to initialize the parallel port by setting the AH register to 1 for the first parallel port. This program was written using the Ecosoft C Compiler sysint function, and most C compilers have a similar library function.

For one reason or another, I decided to write a similar program for the serial port, and that is shown in Figure 2. Note that the program is almost identical, except that I used interrupt 14H for the serial port.

These two examples illustrate why I like the C programming language, in general, and the Ecosoft C Compiler, in particular. As you can see, both programs are quite trivial in C, and they did exactly what I wanted.

Although there is probably some way to do this in BASIC, I did not have the time to fool around to figure out how to do so. If you are considering learning a programming language, you should consider if that language will do what you intend. For

```
/* sinit.c - initialize and display serial status byte */
/* by: William M. Adney */
/* Copyright (C) 1988 by William M. Adney */

#include <stdio.h>
#include <dos.h>

int main()
{
    struct HREG ireg, oreg;

    ireg.ah = 0;        /* to initialize */
    ireg.al = 0;
    ireg.dh = 0;        /* dx is zero for printer 0 */
    ireg.dl = 0;

    sysint(0x14, &ireg, &oreg); /* for serial port */

    printf("\n\nStatus byte in hex is: %X\n", (oreg.ah & 0xff));

/*  END OF PROGRAM */
}
```

Figure 2
SINIT.C to Initialize and Check Serial Port

the most part, BASIC cannot do the things that I want to do. In some cases, I admit that I don't know how to write a BASIC program to do certain things, and these programs are a couple of examples. There are few things more frustrating than spending the time learning a language only to find that it really cannot do what you want to do.

How can you avoid learning a language that does not do what you want? First of all, you must spend a few minutes thinking about what kind of programs you want to write. Do you want to do lots of bit-fiddling (like Figures 1 and 2) or do you want to write other kinds of programs? Will your programs need to have direct access to system hardware, interrupts, and other related functions at the bit-fiddling level? Or do you simply want to learn something about programming in general?

A Closer Look at BASIC

First of all, BASIC is an acronym that stands for Beginner's All-Purpose Symbolic Instruction Code. It was developed in the early 1960's at Dartmouth College for the specific purpose of teaching programming, and it was not intended for anything beyond that. Sure, you can write useful programs in BASIC, but compared to other languages, BASIC is quite primitive and limited by design.

Perhaps the most singular difference between BASIC and virtually all other high-level programming languages is its lack of language-enforced structure. For example, most professional programming languages require that all variables and their types be defined BEFORE they are used in a program. In my C programs shown in the Figures, a variable name is "ireg.ah" that was defined in "struct" for the dos.h header. PASCAL, COBOL, and other languages require similar kinds of variable definitions, but BASIC does not. In BASIC, variables can be defined and redefined "on the fly". While this may initially appear to be an advantage, I think it causes problems that far outweigh any advantage.

I have found the worst part of this problem to be that it can lead to incredibly sloppy programming that can be nearly impossible to troubleshoot. For example, a variable name of "$I" or "$N" is customarily used as a counter, but if that same variable is used and reused for other purposes, the program logic can get very blurred and extremely difficult to follow. In order to help reduce the nature of this problem when I taught BASIC, I required that REM statements defining variable names and their meanings be included at the beginning of each program. You can imagine what happened when I deducted points for the assignment if every variable name was not indicated at the beginning of the program. While my approach led to some inevitable complaints that this was not a BASIC programming requirement, it was more than made up for by the students who later appreciated it because they had gone on to more advanced languages like PASCAL, COBOL and C Programs in those languages will simply not compile if all variable names are not defined before they are used in the program. While that may seem to be a disadvantage, it really is an advantage because it forces the programmer to think logically which is, after all, the "prime directive" of programming.

Another problem with most BASIC versions is that they do not permit the use of labels in the program. For example, the GOSUB and GOTO statements require the use of line numbers instead of labels. Unless you are quite familiar with the program logic and the line numbers of a specific program, it can be quite difficult to do any troubleshooting.

One answer to this problem is to assign "standard" line numbers, but that can be very difficult. For example, you might want all subroutines to begin at line 10000, and while that is not too difficult in the initial program development, it can be quite tricky when you add enhancements or fix bugs. More importantly, labels make programs easier to develop, read, and understand. If labels are chosen carefully, a statement like "GOTO CHECKKEY" is much more meaningful than "GOTO 4732". While some BASIC versions now allow the use of labels instead of line numbers, the standard IBM and GW-BASIC versions do not.

When a system is developed for mainframe applications, many companies use a System Development Life Cycle process for the development of new application systems. Part of that process usually involves the choice of the best programming language for that application. Although it is true that many programming organizations have adopted COBOL as the standard language for all programs, some have learned that it is better to choose a language that is best suited to the application. Unfortunately, that approach is not obvious when one begins to program a microcomputer.

When BASIC was originally developed, it was designed to be a language to introduce students to the fundamentals of computer programming. It was especially designed to be easy to learn which accounts for much of its popularity. Indeed, I believe that BASIC was one of the primary factors that accounts for much of the popularity of microcomputers today, and that is certainly a tribute to its designers. But the easy-to-learn capability has caused some problems.

I think the worst problem here is that BASIC is being used for applications where it is not appropriate. I have especially noticed that many BASIC programs are quite long, and they may contain 500 or more lines of code. Long, complicated programs are difficult to write and debug anyway, and since BASIC was originally intended for short, "training" programs; it has been used inappropriately for these long programs. There is quite a bit of evidence to support this if one does not know about the design intentions, but the most obvious is the BASIC editing features or lack of them. For example, if BASIC had been intended for writing long programs, its designers would certainly have included a simple search and replace feature in the editor, not to mention some easy way to "page" through the program with something other than the rather awkward LIST command.

Since BASIC was designed for training, it was also designed as an interpreted language rather than a compiled language. Compiled languages result in the generation of a COM or EXE file that can be directly executed from a command line. Interpreted languages require that the INTERPRETER be available, as well as the program code. This leads to two kinds of problems.

By the very nature of interpreted languages, they are slow because every single line of code must be "translated" into binary before the computer can use it. And if you write some kind of a loop that is executed 25 times, the interpreter must translate each line within the loop 25 times because there is no way to "predict" what statement comes next. Interpreted languages are quite helpful for the immediate feedback necessary for rapid learning of a language, but the penalty for this is that it is slow.

The second problem is that you must always have the BASIC interpreter available to the system whenever you want to execute a BASIC program. In the latest version of Zenith's GW-BASIC (version 3.2) that I have, BASICA.COM is 8,659 bytes and BASICA.EXE is 78,528 for a total of 87,187 bytes. That space requirement can be quite critical in a floppy disk system although it is not as serious in a hard disk system. Even if you write a small program, you still need over 87,000 bytes of disk space for the interpreter alone. The bottom line is that the space requirement for the interpreter and the program is significantly more than a compiled program needs. And aside from being smaller, you can copy a compiled program to any disk or subdirectory without having to worry about whether the interpreter is available or not.

By now you may be wondering if I ever use BASIC, and the answer is: Of course I do. It has its uses, and I generally try to limit my BASIC programs to a single screen for ease of editing. As a personal matter, BASIC cannot generally do the kinds of things that I frequently need, and I have found that the C programming language is a much better choice for my needs.

Why I Like C

There are lots of reasons to choose C, and I guess my main one is primarily related to lack of time — I simply don't have the time to spend in learning a new assembler for every different CPU and feature that has been and will be developed. I have probably forgotten more machine-unique assembler languages than most people know about beginning with the IBM 7090/7094 and CDC 6500 mainframes. And of course, there is no way to forget the IBM 360 and IBM 370 series not to mention the current IBM 30xx (3033, 3081, and 3090) series computers. And there is no way to forget the IBM 4300 series of minicomputers. For PC microcomputers, there are the instructions for the Z80, 8080, and 8085 series CPUs, as well as the 8088, 80286, 80386 series. And you can throw in the new instructions for the numeric co-processors (e.g., 8087, 80287, etc.) for good measure. Perhaps a well-rounded computer programmer should also know something about assembler for the Apple computers (6805?), as well as the 32-bit 68000 CPU.

The point of this is that all assembler languages are CPU-dependent within a series, and while most are quite similar in structure, they frequently have different mnemonic instructions and process differently. When the 8088 series computers used a substantially different instruction set compared with the 8080 series, I gave up because I did not have the time to learn ANOTHER new assembler.

So I began looking for a language that could be efficiently used on a mainframe, as well as a microcomputer. In short, I was trying to find a language that was "portable" so that I could learn it once and compile programs by simply using the appropriate compiler for that specific type of computer. I rejected COBOL because it is known as a resource hog (for memory and disk space), and FORTRAN was not powerful enough to do the things I wanted to do. Ideally, the programming language would be available across all of the various computer sizes I work with, and I wanted to be able to easily access computer hardware at the bit-fiddling level I need.

Since C compilers are available for a broad range of CPUs, it seemed like the ideal choice for me. In addition, I found that functional C programs could be quite short, and there was a pretty good standard established for the C library functions. And many compilers have additional functions that make it quite easy to access hardware features on PC compatibles even though there is no real standard for that. In addition, you can also buy supplemental libraries with functions that make program writing even easier because you don't have to spend time re-inventing the wheel. That was a definite plus in my book.

All of these considerations conspired to make C my programming language of choice. Some people believe that C is a "write-only" language — that is, you can write it but never read and decipher it when you look at a program after a few months. Perhaps these people have seen poorly written programs with few comments as to how the code works, but I have found that sloppy programming can convert nearly any language into a write-only one — even BASIC. Although it is true that C is a "terse" language, I think that is an advantage because fewer keystrokes are required to write a program. On the downside, you must KNOW what you are doing when you write a C program because the language is so powerful.

Everything considered, C is much more powerful than BASIC. In particular, C's I/O capabilities are far greater than the primitive functions available in BASIC. A similar examination of the input and output capabilities of COBOL will also illustrate why it is still used in over 90% of mainframe applications. Standard COBOL and C programs can use a lot of the built-in features and libraries. Then, all you have to do is get a new compiler for a new computer, and you don't have to re-learn the language.

Computer Viruses

Some skeptics may claim that all the current concern about computer viruses is more hype than fact. Not true. Viruses present a very real problem for computer users, and I have seen the so-called "Scores" virus that infected about a dozen Macintosh computers at one company. Despite protests to the contrary, these viruses can cause serious problems.

As Pat Swayne pointed out in the June 1988 REMark, there seem to be a few "bad guys" who take some kind of perverted pleasure in causing problems for other people. While there are few who would argue that these bad guys are frequently quite intelligent, it is clear that they have absolutely no judgment whatever. It is these same people who have changed the commonly understood meaning of "hacker" from a complimentary term to one which essentially means an idiot who delights in penetrating computer systems and wiping out data.

Many of the viruses attach themselves to a file used by the operating system as Pat pointed out in his article. One of the most insidious viruses that I know about is attached to COMMAND.COM and passed on to other disks by the DIR command. Pat's method of detecting a virus in the system files is good, and I suggest you read his article. But there are some other ways to find out if you have a virus in your system.

The first way is somewhat obvious because your system will not perform "normally". You may have sudden and unexplainable system freezes using the same software that you have always used. Or you may have problems running the software — it doesn't save or print files properly, it doesn't respond to commands normally or it just generally does strange things that it didn't used to do.

The second way is to note the drive light on a floppy or hard disk drive. Does it light

up every few minutes indicating some kind of disk activity when you are not using your system? That is not a positive indicator because some software has an "autosave" feature that records data stored in memory after so many minutes or keystrokes. Check your software manual to see if you have that feature — if you don't, you may have a virus in your system.

Another clue for virus-infected programs is to keep an eye on the date of COM and EXE files displayed by the DIR command. In particular, there should generally be no date change in COMMAND.COM unless you have personally made a change. Since that requires technical knowledge, you can check the date of COMMAND .COM on your DOS distribution disk, and it should be the same throughout your system on your working disks. Dates for application software will usually be the date of installation or change since install programs frequently update the code, such as happens during printer installation and similar changes. If you have a directory listing program that displays the date of system or hidden files, be aware that the BIOS file (e.g., IBMBIO.COM or equivalent) will be updated by both the CONFIGUR and DSKSETUP programs, and the date (but not the size) will be changed accordingly.

The last way is to have an idea what files should be on a disk and what files shouldn't. Unfortunately, it is quite easy to "hide" files from the DIR command, so you will need some kind of utility program that displays ALL files on your disk. If you have a PC compatible computer, I recommend the HUG PS's PC Utilities disk because it contains the D.COM program that will list ALL files on a disk regardless of attributes. By use of clever highlighting on the directory display, you can specifically tell which file has what attributes. For example, the PC BIOS (IBMBIO .COM) and System Kernel (IBMDOS .COM) usually have both the System and Hidden attributes set. In some cases, they are also Read-Only. Pat's D program will tell you what the current attributes are in any case. Since this program displays the size of the files, you can also watch for any unexpected changes in the file size. Even though this program does not display the dates of the files, it is usually more important to keep track of critical files and their sizes. The easy way to do this is to run the program and print the screen so you will always have a ready reference showing the "authorized" size for each file.

Some Thoughts About Viruses

I, for one, am getting very tired of all of the "product tampering" that is going on. Whether it's adding cyanide to a product or creating something that causes a computer to lose data, enough is enough. And to add insult to injury, many of the so-called "child proof" packaging is also "adult proof". During the last years of her life, my mother had gout, and she was unable to open any of these containers. It is unfortunate that we have to deal with the small number of people who make life miserable for the majority.

Because computers are relatively new, there are few laws that can be used when computer property (i.e., data) is vandalized or destroyed. I believe that a federal law should be enacted that makes it a felony to intentionally destroy data in a computer system. This kind of thing would need to be carefully written to allow research to be conducted, but I think that could be done with a little thought.

Preventing a Virus

In all documented cases, there is only one way to get a virus in your system, and that is to copy an infected program. Since viruses are rarely benign, an "ounce of prevention" is still worth a couple tons of cure. Unfortunately, the prevention approach presents considerable operational problems. The rules for preventing a virus in your system are shown in Figure 3.

1. Copy program files from original disks only.
2. Do not copy files from computer bulletin boards.

Figure 3
Virus Prevention Rules

As you can see, the prevention of viruses is not easy and results in considerable difficulty for everyone. Even these rules are not absolute because some viruses can attach themselves to data files too. Interestingly enough, the first rule will also aid in the fight against software piracy, but the point is to use a known original source that you trust.

The second rule is somewhat ridiculous because it is frequently not a viable option. Unfortunately, some viruses are spread through bulletin boards, and conscientious operators are inventing many ways to cope with the problem. For security reasons, I will not discuss any of those here. Many of us use bulletin boards on a regular basis, and that is one reason I think harsh penalties should be invoked against the perpetrators of these problems.

These are a few things you can do to help prevent a virus from infecting your system. I hope that you find some of these suggestions helpful and informative.

Powering Down

There are other kinds of problems that have been found in computer programs, and the virus is only one of them. I will tell you about a couple of others I know about next time, and I have also been looking at one of the "vaccine" programs that can help protect your system.

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

Products Discussed

HUG Software
PS's PC Utilities (885-6011-37) $20.00
GW-BASIC FlipFast Guide
(885-4601) $20.00
MS-DOS FlipFast Guide
(885-4602) $23.00
FlipFast Guide Combo
(885-4603) $41.00
Heath/Zenith Computer Centers
Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3838 (HUG Software only)

Eco-C88 C Compiler (MS-DOS) ... $59.95
Ecosoft, Inc.
6413 N. College Avenue
Indianapolis, IN 46220
Orders only: (800) 952-0472
Technical Support: (317) 255-6476

GW-BASIC Version 3
PC only (MS-4164-11) $99.00
Heath/Zenith Computer Centers
Heath Company Parts Department
Hilltop Road
St. Joseph, MI 49085
(800) 253-7057
(Heath Catalog orders only)

*

How to Add Drives to a Zenith Z-386

by Joseph Katz

Part 1. One of Micro Solutions' CompatiCards lets you add two internal and two external floppies. Part 2 of this series will explain ways to help you multiply them and your present drives.

When you look at today's stock Zenith Z-386 computer head on, you'll see its one 5.25″ high capacity floppy disk drive. You can add a second floppy disk drive, perhaps a 3.5″ drive for compatibility with IBM's PS/2 lineup or computers such as Zenith's Z-183 and other laptops. A panel on the lefthand side of the first drive covers an opening for the second. Although there are two more such panels on a Z-386, the disk controller board in Slot 1 of the computer is a combination controller that supports only two floppy disk drives and two hard disk drives. You can't mix them. So you'll be told that the Z-386 itself is limited to no more than two floppy disk drives.

That's wrong. To begin with there's space enough in Zenith's 80386 desktop computer for a total of six half-height drives. If you use half-height hard disk drives, therefore, you can install four half-height internal floppy disk drives, 5.25″ or smaller. And the Z-386 power supply is hefty enough for all those drives. What you need are the floppy disk drives themselves, mounting hardware and cabling, and a supplemental controller board. But the catch is that it can't be just any floppy disk controller board. You need one that can be configured so it doesn't conflict with Zenith's combination controller already in Slot 1.

Micro Solutions' CompatiCard is such a board. It's a supplemental floppy disk controller that can be used in Zenith's Z-386 computer to install two additional floppy disk drives internally, filling the computer with a full complement of four internal floppy disk drives. Of course those four floppy disk drives leave room for two half-height hard disk drives too.

CompatiCard can do more. Put up to two more floppy disk drives in an external cabinet with its own power supply, run the proper cabling through the computer's back to the board, and that one CompatiCard will let you have a total of six floppy disk drives operational on one Z-386.

Want still more? You can put up to a total of three CompatiCards in a Z-386, so it's possible to operate as many as fourteen floppy disk drives on that one computer. Why anyone would want so many is another matter entirely. But if you find comfort in such large potential quantity, you may be even more heartened to know that CompatiCard can handle all major and most minor kinds of microcomputer floppy disk drives. (See Table 1.)

What follows here are instructions that apply specifically to a CompatiCard used to add just two internal floppy disk drives to a Z-386. With these instructions as a complement to the *CompatiCard User's Guide*, the manual, someone experienced in drive installations ought to have little trouble using CompatiCard to install still more drives in and around Zenith's 80386 machine. If you've had no such experience, or relatively little, you should turn the job over to a Zenith dealer, service facility, or consultant. Most of the following assumes conventional wisdom, but there are parts that bend it a little. An experienced installer can extrapolate from these instructions and that *User's Guide* to install external floppy disk drives up to the capacity of CompatiCard and your budget.

Contents

The CompatiCard package includes the board itself, software such as the driver

PRODUCT INFORMATION
CompatiCard. $175.
Micro Solutions Inc.
132 West Lincoln Highway
DeKalb, IL 60115
815/756-3411

needed for the board, and the *User's Guide*. The latter has good instructions that assume—as they must, because there are many different models of so many different drive types—you know your drives, their characteristics, and what to do with them. Not included is cabling, either to control the drives or to power them, or any other hardware that might be needed for your installation. Those limits also are necessary and reasonable given the multiplicity of drives CompatiCard supports. You must obtain all such hardware. (Micro Solutions sells an optional cable for 8″ drives.) Nor does the manual include instructions for installing drives in the computer, either the mechanical or the electrical installation. You must know how to do those things. Requirements for all that will differ according to the drives and the computer, so you assume their burden and the ultimate responsibility for your installation. Micro Solutions' Technical Support is extraordinarily helpful over the telephone, but not even the best technician can see what you actually have or have done with it.

Table 1

Floppy disk form factors supported by CompatiCard.

The listed version of DOS is the earliest required to support that form factor. When no DOS version is listed, that form factor is unique to CompatiCard's driver. The diskette will not be usable without it.

Description	Capacity	DOS Version
5.25″ 48 TPI	360 KB	2.00
	320 KB	1.10
	180 KB	2.00
	160 KB	1.00
5.26″ 96 TPI	800 KB	
	360 KB	2.00
	320 KB	1.10
	180 KB	2.00
	160 KB	1.00
5.25″ 96 TPI	1.2 MB	3.00
high capacity (AT type)	800 KB	
	360 KB	2.00
	320 KB	1.10
	180 KB	2.00
	160 KB	1.00
5.25″ 96 TPI	1.2 MB	3.00
300/360 RPM (dual speed)	800 KB	
	360 KB	2.00
	320 KB	1.10
	180 KB	2.00
	160 KB	1.00
8″	1.2 MB	
soft sector	250 KB	1.00
3.5″	720 KB	3.20
3.5″ 600 RPM	720 KB	3.20
3.5″	1.4 MB	3.30
high capacity	720 KB	3.20

Overview

Here's an overview of the process for installing two additional internal floppy disk drives in a Z-386. Make sure you do it while the computer is switched off and its power cord is unplugged.

First, configure each additional drive. Second, configure the CompatiCard. Third, mount each additional drive in one of the two Z-386 drive cages. Fourth, daisy chain the first additional drive to the second and the second additional drive to the CompatiCard. Fifth, extend the power cables for the normal two internal drives so they also power the additional two internal drives, then plug the appropriate cable to each drive. Sixth, install the CompatiCard in the Z-386. Seventh, install the CompatiCard driver.

When those seven steps are completed, and you have checked each of them carefully, you may plug in the computer and test the installation. Make sure you test each drive with scratch disks you can afford to lose before you entrust it with any valuable disks. That precaution is normal when installing disk drives and will not deter an experienced installer. Someone with such experience should find this installation relatively simple. If the unexpected happens, call Micro Solutions' Technical Support: it is really good.

Installation notes

Because individual floppy disk drives usually are not supplied with instructions, you'll have to draw on experience and the useful section in Zenith's *Technical Manual* for the Z-386. Two general guidelines concern the position of each pair of drives on a controller cable. The first guideline concerns the terminating resistor: the second of the two drives usually requires a terminating resistor, the first usually does not, and if there is only one drive on the cable it usually does. The second guideline concerns the Drive Select jumper or switch: set both drives to DS 1 if the controller cable has connectors 10-16 twisted at the far end, but set the first drive to DS 1 and the second Drive to DS 2 if the controller cable is straight through. Those are only general guidelines and might not apply to your installation. If anything in those two guidelines sounds like gibberish, you probably should have someone else do the entire installation.

It's simple to configure CompatiCard for two more internal drives in a Z-386. Figure 2 shows a skeletal diagram of the board with two pin connectors (P2 and P3) and a

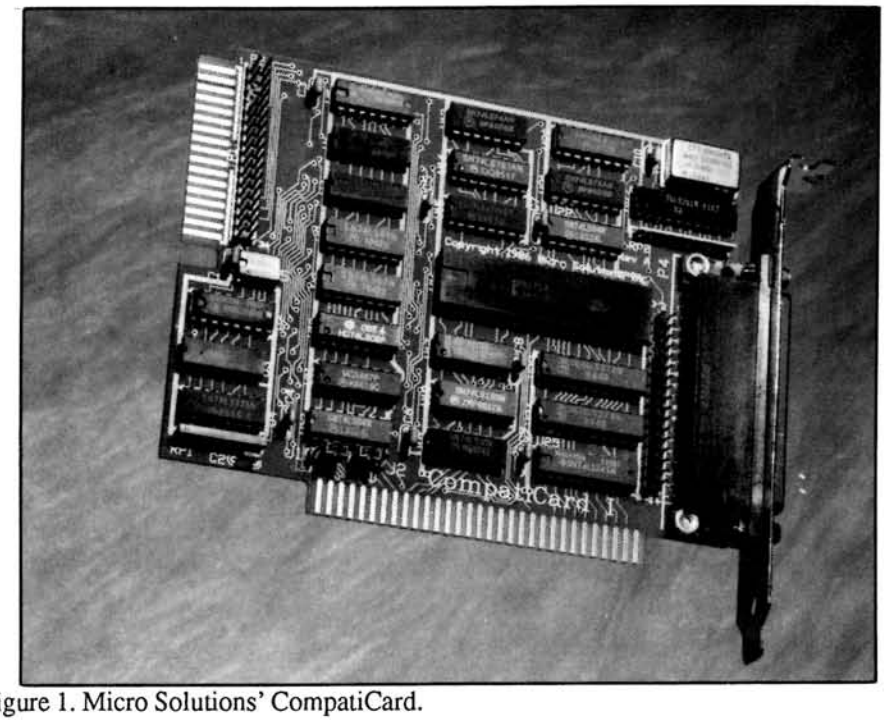


Figure 1. Micro Solutions' CompatiCard.

pair of jumper blocks (J1 and J2). There are two pin connectors because each board will control up to two pair of drives, each pair on its own run of cable. But there are additional connectors on the board for use with drives that require an edge connector and drives that require a plug. You therefore must configure the CompatiCard to select the connectors you'll actually use: the pin connectors, P2 or P3. Because you can install up to three CompatiCards in a Z-386, you must configure the CompatiCard as the second, third, or fourth controller: logic suggests it should be configured as the second in this installation. Although the CompatiCard can serve as the primary controller in other computers, you must not use it that way in a Z-386. Jumpers J1 and J2 select both the connectors and the controller designation at one time: set them so both J1 and J2 are in position "B" for this installation. Your CompatiCard will be configured as the second controller, selecting a cable at either P2 or P3. (This jumper setting actually directs the controller to I/O ports 0370H-0377H. You don't have to know that to use CompatiCard, but you might need to know it if you add some other board that happens to use those ports too. Then you might find the CompatiCard easier to configure than the conflicting board.)

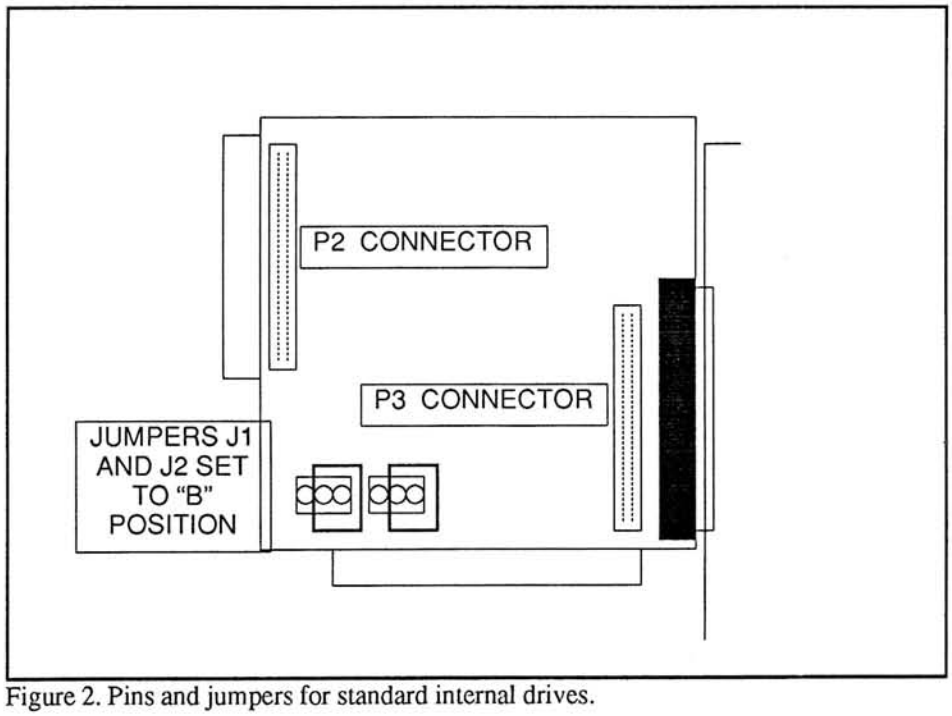


Figure 2. Pins and jumpers for standard internal drives.

You likely will need to buy, modify, or devise hardware for mounting the additional drives. With a combination of 5.25″ and 3.5″ drives, it might be easier to mount the 5.25″ drives in the first row and the 3.5″. drives in the second, flipping the mounting platforms for the 3.5″ drives.

Be careful when you daisy chain the drives to the CompatiCard: make absolutely certain about polarity so you're unshakably sure—and right—that the cable is oriented to Pin 1 on each device. If polarity is reversed at any point, you will at least destroy the contents of any disk you insert into the drive. You may plug the cable into either P2 or P3 on the CompatiCard, but P2 should be more convenient when the board will be installed in a Z-386 slot to the left of a slot that is, or will be, occupied. For each pair of additional floppy disk drives, you'll need to be prepared with one 34-conductor ribbon cable made up with the proper connectors: a standard 34-pin jack at one end to join the card, and either edge connectors or jacks as appropriate to join the drives you'll use. You may use either straight-through cables or those with conductors 10-16 flipped at the connector for the first drive in the pair. Make sure the second drive in the chain is configured properly for the cable you've used.

You'll need one Y power extension cable for each additional drive. These special cables have a four-pin male Molex connector wired to two branches that terminate in female Molex connectors. The extension cables tap power from the two cables intended for the internal floppy disk drives normally supported by the Z-386. The existing cables in the computer are marked "P4" and "P2". Attach the P5 power cable to the male Molex connector of one extension power cable and attach the P2 power cable to the male Molex of the other. The connectors are molded to mate in a way that

preserves polarity: be careful not to force them together the wrong way. Insert one of the two female connectors from the extension of power cable P5 onto the power plug of the first drive already in the computer, the one you call "Drive A." Insert the other female connector from that same extension of power cable P5 onto the power plug of the additional drive installed right below that "Drive A." Follow the same procedures with the power cable P2 extension.

Although CompatiCard has an XT-style buss connector, it may be installed in any free Z-386 slot—even one of the 32 bit slots. Of course common sense will lead you to choose the 8 bit Slot 10 if it's available, the 16 bit slot 9 otherwise, and one of the 32 bit slots only if the two smaller slots are occupied already.

CompatiCard requires a special device driver, CCDRIVER.SYS. Copy it from the distribution disk to the root directory of Drive C. You will need to create or modify the CONFIG.SYS file there so it loads CCDriver properly for each floppy disk drive you've installed with CompatiCard. (Don't load this driver for drives controlled by Zenith's controller.) Make sure to use an editor that produces straight ASCII text. Here's an example command line from an actual installation of two additional internal drives:

```
device=ccdriver.sys /4,7 /4,0
```

Words on both sides of the equals sign are boilerplate instructions so MS-DOS loads the device driver. What follows is two sets of parameters to that driver, one for each drive. Beginning with a space followed immediately by a slash, the parameters give the drive address, a comma, and the drive type. The drive address is a code summarizing all this: your designation of the CompatiCard as second, third, or fourth controller, your choice of a flipped or straight-through cable, and your use of the P2/P3 plugs on the board. If you have followed my suggestions here, the drive address always will be "4" if your cable is flipped or "5" if your cable is straight through. If you've flown solo, there's a table of drive addresses in the manual. The drive type will depend on which drives you've installed: the manual includes a clear table of the eight supported drive types and formats. My example command line loads the driver to support one 3.5″ high capacity drive of the kind supplied with a stock IBM PS/2 computer and one 5.25″ double density drive of the kind used in most IBM PC and XT compatible computers.

Testing and use

Right after CCDriver loads, it reports the "logical" drivenames of the additional internal drives. Those are names you'd use to access the drives, and you'd use them in all commands the same way you'd use the logical drivenames of the two drives on Zenith's controller.

What might surprise you, though, is that these drivenames likely will be further up the alphabet than you may expect. Those in the example installation were Drives G and H: the system's hard disks included one partitioned as C and D, and another partitioned as E. Keep in mind that it's MS-DOS, not CCDriver, that assigns drivenames. Because MS-DOS deals with hard disk partitions in pairs, it reserved F for a second partition on the second hard disk.

Don't assume that the additional drives are in proper working working order just because CCDriver successfully reported their drivenames. Conversely, don't assume that the additional drives aren't working just because their access lights didn't flick on momentarily at boot time, as do the first two internal drives. The Z-386 BIOS and Zenith's controller are never aware that these drives exist, and MS-DOS is ignorant of their existence until CCDriver is loaded. The additional drives therefore will not be exercised at boot time as the first two drives are. You must test the drives.

The first best test is to look at the drive access lights. If they glow steadily, shut the computer immediately and check to see if you've reversed polarity on the controller cable associated with those drives. Under no circumstances should you insert a disk into those drives until you are absolutely sure the polarity is correct.

The next best test is to try the drives. If the drive lights are off, as they should be, try accessing each drive by calling for a directory of a disk in it—but don't put a disk in it. The drive access light should glow until you get a "Drive Not Ready" error. Abort and go on to test the next drive the same way. When that test is passed you know that the drives have power and are being selected. Then try formatting a blank disk in each drive. If you get a "Drive Not Ready" error after a few minutes, you probably have a defective controller cable. Replace it. If the disk seems to format properly, check it with the MS-DOS Chkdsk utility program to make sure.

The CompatiCard software includes a CCFormat program. It's more versatile than the MS-DOS Format program because it accepts parameters for formatting disks on all the drives supported by CompatiCard. The MS-DOS Format program doesn't. You'll therefore want CCFormat if you're using such a drive. (Note that some drives—such as 8″ drives—supported by CompatiCard can't be installed internally in a Z-386. But they can be used externally: check with Micro Solutions about any special cabling requirements. Available from them, for example, is an adaptor and cabling for 8″ drives.) Otherwise use Format as you usually would.

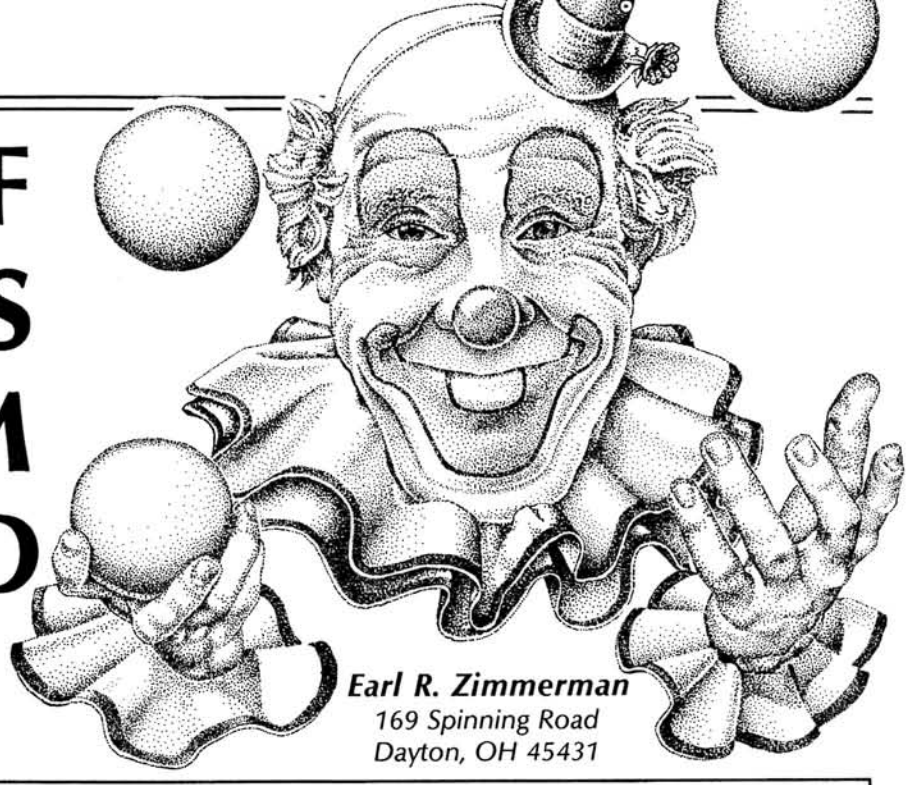
You might want to change the order of the drivenames MS-DOS assigned to your additional floppy disk drives. Say you want the 5.25″ drive in my example to be Drive G and the 3.5″ to be Drive H. All you need do is change the order of the parameters to CCDriver in the CONFIG.SYS file and reboot the computer. MS-DOS assigns logical drivenames anew each time the computer boots.

It's good practice to load CCDriver early as possible in any chain of device drivers that affect storage devices. That way your disk drives will have the lowest possible drivenames. Remember that RAM disks have drivenames too. If you sometimes boot the computer with a CONFIG.SYS file that loads a RAM disk driver and sometimes boot it with a CONFIG.SYS file that doesn't, you may be confused by changing floppy disk drivenames. That's all the more reason to invoke CCDriver right at the beginning of each CONFIG.SYS file.

Make sure to note, when CCDriver reports them, the logical drivenames assigned to your additional floppy disk drives. If you forget, the CCDrives program supplied with CompatiCard will remind you.

You should find CompatiCard an excellent product that increases the versatility of Zenith's already-versatile Z-386 computer.

SHOWOFF GRAPHICS PROGRAM REVIEWED



Earl R. Zimmerman
169 Spinning Road
Dayton, OH 45431

Introduction

One of the newer third party software products on the market for the Z-100 is ShowOff, by Hogware. This program gives a Z-100 high resolution graphics (640 X 480 pixels) and allows a user to produce professional looking charts or drawings. The emphasis of this article will be on the minimum system requirements, keypad versus mouse drawing, major program features, slide presentation utilities, import capabilities, supported printers, customer support and upgrades, and suggested program improvements.

Minimum System Requirements

ShowOff requires a H/Z-100 (hard disk or dual floppy) with 384 KB of RAM, MS-DOS 2.XX operating system, and three banks (color) of 64KB of video memory. Although not required, a Logitech C7, Microsoft, or PC mouse is also recommended for ease of drawing. A ShowOff supported printer is also optional.

If you are not sure your Z-100 meets the RAM and video requirements, simply turn your computer on. Press the Control, Reset, and Delete keys simultaneously. A finger prompt will appear, then type S. Your computer will then respond with how much RAM you have and the type video you have. You can also order a demo disk for $3.

Keypad Versus Mouse Drawing

Two different versions of ShowOff are supplied on the disk, SHOWOFFM, for use with a mouse, and SHOWOFFK, for

Table 1 Keypad Versus Mouse Drawing

FUNCTION	SHOWOFFK	SHOWOFFM
Cursor Movement	Arrows & Numeric Keypad	Mouse is moved
Command Execution	Enter Key	Left mouse button
Freehand Drawing	Keypad "5" Key	Middle mouse button
Connected Lines	Enter Key	Left mouse button
Spray Painting	"V" Key	Right mouse button

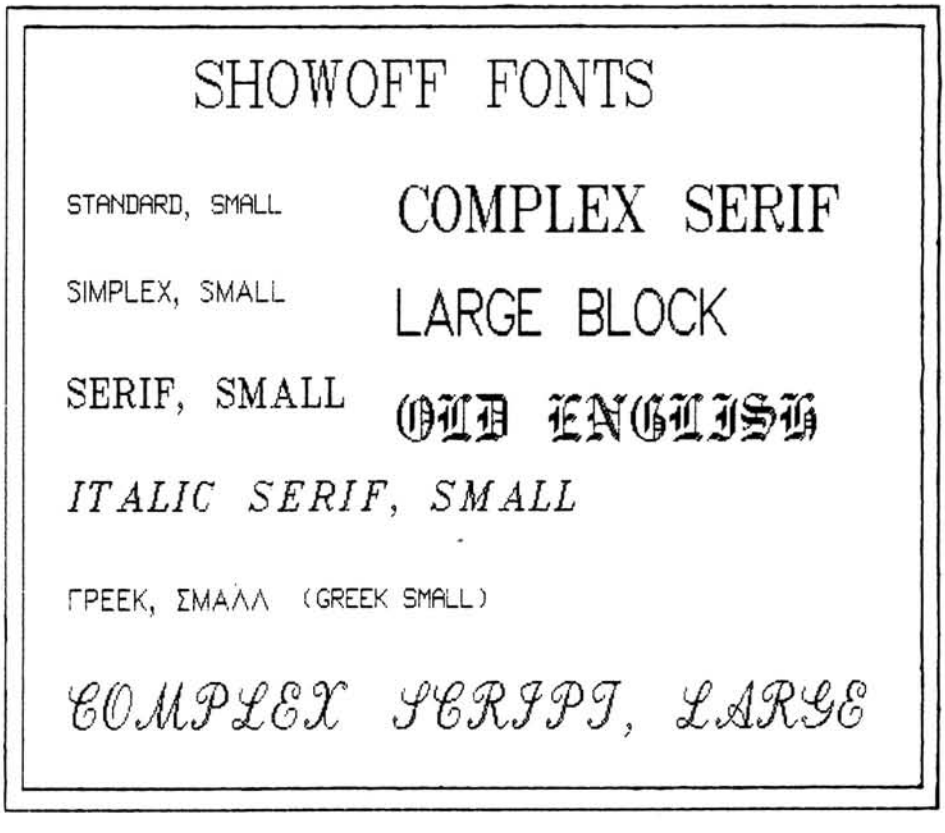


FIGURE 1 - SAMPLE SHOWOFF TEXT FONTS

use with the numeric keypad. Both programs have the same capability. The primary differences in the two programs are illustrated in Table 1.

Which method is the most efficient and recommended? I, along with the authors, recommend the use of a mouse because it is much easier to create graphs. It takes a long time to move a cursor pixel by pixel around a 640 X 480 screen. SHOWOFFK allows you to set the pixel movement by depressing 0-9 on the keyboard (not the numeric pad). Depressing "0" will set the movement to 10 pixels at a time. Even at the maximum setting, it still takes a great deal of time to get around the screen.

Major Program Features

Text/Symbol Fonts. There are 25 different fonts to choose from when designing your pictures. Some of the text fonts are displayed in Figure 1, while some of the symbol fonts are displayed in Figure 2. There are more than enough for everyday use. Not only can you display these 25 fonts, you can also select the angle (0-360 degrees), and the size (1-9) of the text or symbol.

Lines and Shapes. With ShowOff, you can select one of ten different lines to draw with. The default line is a solid line. Line patterns are changed by depressing the "-" key and selecting a number from 0-9.

One of the better features is the ability to draw connected lines. Pressing "Z" puts ShowOff in the connected line mode. Pressing the Enter key in SHOWOFFK or the left mouse button in SHOWOFFM sets the starting point of the various lines. For instance, let's assume you want to draw a triangle and fill it with a color. You would use the connected line mode because it would eliminate the possibility of drawing an imperfect character where the fill color would "leak" out of the triangle and cover your drawing.

In addition to drawing lines, you can draw a circle, ellipse, or a rectangle.

Freehand Drawing. You may have a requirement to draw some unusual shapes or just want to doodle. ShowOff allows you to do this with its freehand drawing function. Freehand drawing is activated in SHOWOFFK by depressing the numeric keypad "5" and one of the cursor movement keys. Cursor movement keys consist of the arrows keys and the numbers on the numeric keypad, with the exception of the "5"and "0". Pressing "5" again ends freehand drawing. If you are using SHOWOFFM, all that's necessary is to depress the middle button on the mouse and then move the mouse.

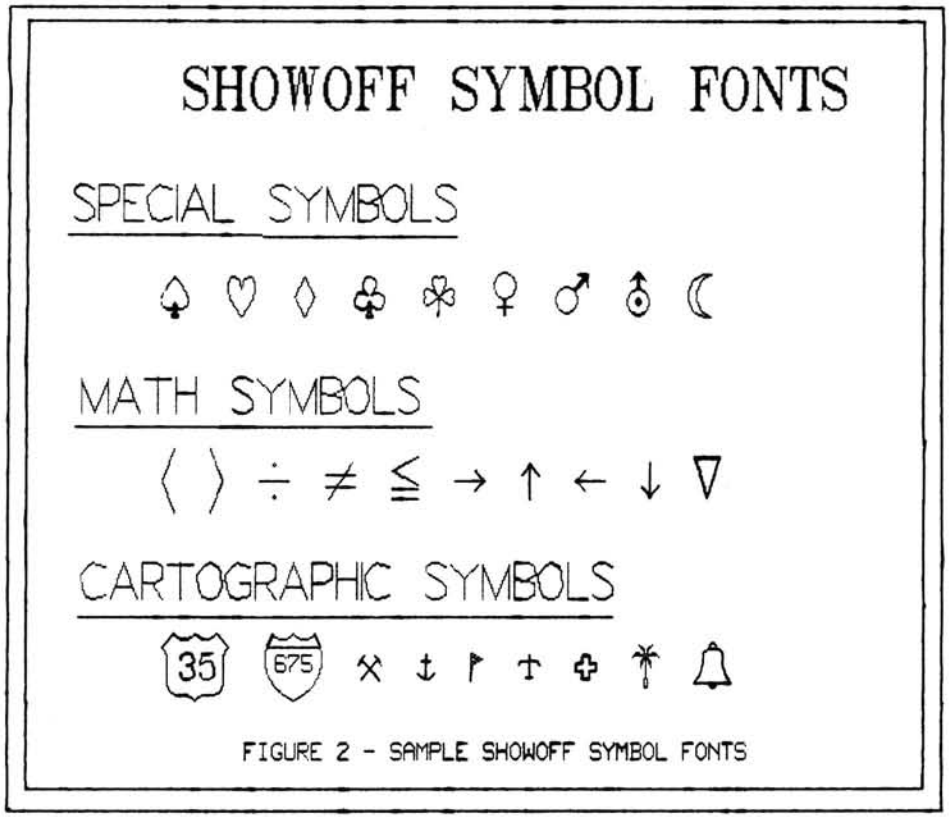


FIGURE 2 - SAMPLE SHOWOFF SYMBOL FONTS

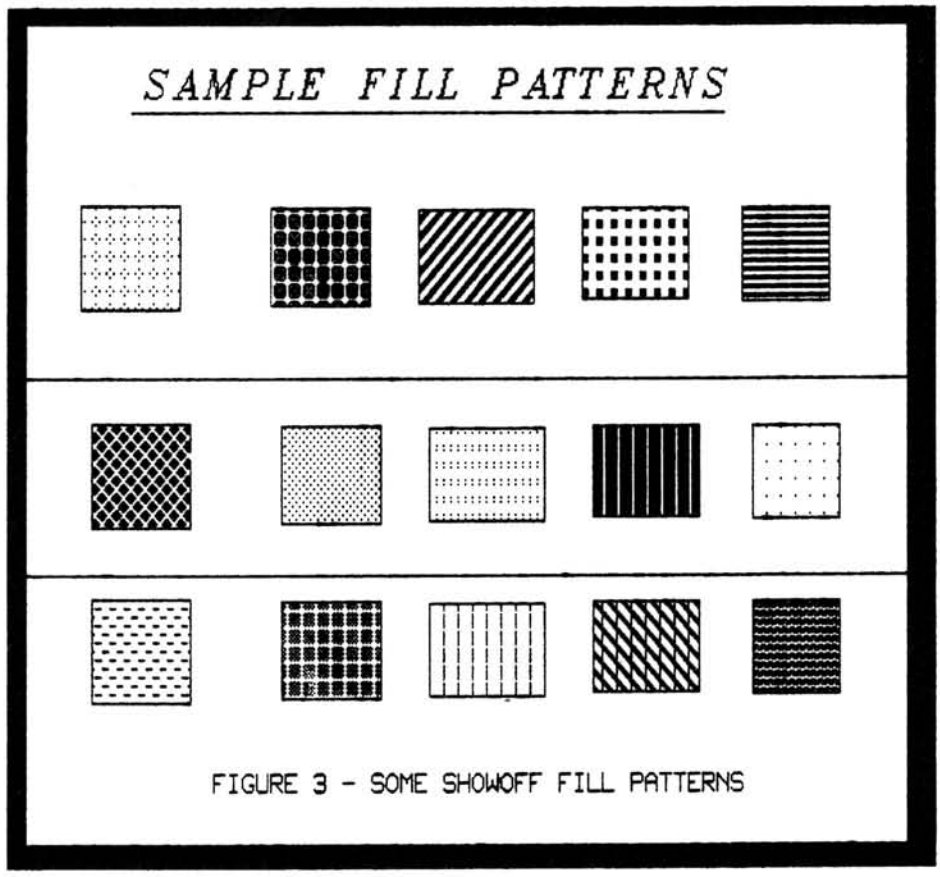


FIGURE 3 - SOME SHOWOFF FILL PATTERNS

Various Drawing Colors. ShowOff has eight different drawing colors. These eight colors are the same colors that are used in

ZBASIC. The drawing color is selected by depressing the "C" key and then specifying the color number by depressing a number from 1 to 7. One minor drawback is that you have to remember what number corresponds to each color, as the prompt does not list the choices. The current drawing color is displayed in the upper left hand corner of the screen.

Area Filling with Colors and Patterns. Another nice feature of ShowOff is the capability to fill areas with 92 fill colors or 92 predefined patterns. See Figure 3 for some of these fill patterns. The fill color is selected by depressing the "F2" key and selecting the number of the desired color. If you wish to use a fill pattern instead of a fill color, just press the Return key after the fill color menu is drawn and the fill pattern menu will appear.

In addition to the 92 fill patterns, you can define your own fill pattern by depressing the "F3" key and entering numeric values for the four prompts which represent various contributions to the pattern. A table in the documentation explains the meaning of each value, while another table lists the numeric values for each of the existing fill patterns. For instance, pattern 100's (the first fill pattern) value is 4848.

Spray Painting. Instead of filling an area with a color or pattern, you can spray paint the area. Before spray painting, the intensity of the spray can be user defined by depressing the "I" key and selecting the intensity value. The higher the value, the larger the area that's painted. The default value is 24 pixels.

To spray paint using SHOWOFFK, the cursor is positioned and the "V" key is depressed. Using SHOWOFFM, all that's necessary is to move the cursor with the mouse and depress the right mouse button.

Image Rotation. An image can be rotated up to 360 degrees. See Figure 4 for an example of rotated images. Image rotation is accomplished by pressing the SHIFT/F0 keys simultaneously and enclosing the selected image in a circle. You then specify the angle you want the image rotated to. The rotation of an image is rather slow because of the computations required to move the pixels from one position to another. You should therefore only rotate small images if you are pressed for time.

In Figure 4, I rotated text as an example. Normally when creating text, the desired angle would be specified at the time the text was initially entered. However, this is a handy feature if you decided to change your picture at a later date.

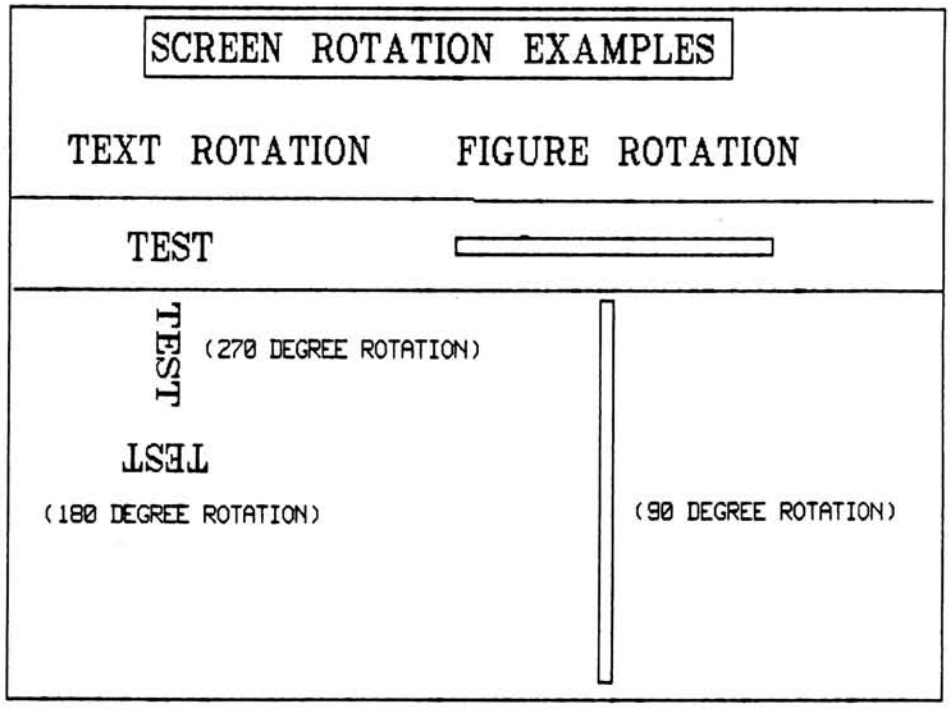


FIGURE 4 - DEMO OF SHOWOFF ROTATION FEATURE

Slide Presentation Utilities

ShowOff contains two utilities for creating slide presentations. The first, **SLIDE**, will display a single picture, and the other **SLIDESHO** will display a series of pictures. These two programs are separate from the actual ShowOff program.

To use this utility, all that's necessary is to type:

```
SLIDE <FILE NAME> (The .PII extension isn;t necessary.)
```

A suggested use is including **SLIDE** in a batch file. Below is an example using the four figures with this article. It assumes **SLIDE** is in drive A and the pictures are in drive B.

ASCII text files could also be displayed inside a batch file by using the command: **TYPE <filename.doc>**. This is possible because ShowOff has the capability of displaying high resolution pictures and normal resolution pictures.

A much easier method of accomplishing the same thing as the previous batch file is to use **SLIDESHO**. A word processor that creates ASCII files is used to create a file named **PICNAMES.DAT**. Using the same four figures and the same assumptions, your file would look as follows:

```
B:FONTS
B:SYMBOLS
B:FILL
B:ROTATE
```

To run the program type: **SLIDESHO GO.**

Instead of creating a permanent presentation, you can also create a temporary presentation by just typing **SLIDESHO.**

BRIEF.BAT	**EXPLANATION**
ECHO OFF	
SLIDE B:FONTS	(Display Figure 1)
PAUSE	(Wait until key depressed)
SLIDE B:SYMBOLS	(Display Figure 2)
PAUSE	
SLIDE B:FILL	(Display Figure 3)
PAUSE	
SLIDE B:ROTATE	(Display Figure 4)
PAUSE	
A:	(Return to A drive)

ShowOff will prompt you to enter "F" or "K". By selecting "K" you are then prompted for the slide names and the information is entered exactly the same as the previous example. You are then prompted to enter the time (1-9) for each slide to be on the screen. The documentation does not state how long the time increment is for each number. I've timed each interval at approximately five seconds, but it could vary according to whether or or not you have a speedup kit or a V-20 chip.

Import Capabilities

ShowOff can also be used to "capture" and enhance normal resolution graphs created by other programs. A graph can be created using Lotus 1-2-3 and enhanced with Showoff. While Lotus is specifically mentioned in the documentation, it will most likely work with programs that create graphs in a similar fashion (i.e,CAD programs). For instance, I discovered that by following the procedures in the tutorial, I was able to capture and enhance three dimensional bar graphs I created with the Z-100 version of Enable 2.0. Instead of being limited to the nine texts fonts in Enable, I now have a wider variety to choose from. In addition, I am not limited to eight fill patterns. Also, the SHOWMAC.COM utility will allow you to capture and enhance files stored in the Macintosh format.

You can also purchase the Art Utility Disk from Hogware for an additional $15.00. This disk has utilities that will allow you to capture CGA screen images and will allow you to convert ShowOff pictures to MacPaint format for use with Logitech Publisher. Hogware expects this product will run on the Z100 under ZPC. Logitech Publisher is a page layout program that includes some text fonts as well as ShowOff compatible clip art. The disk also contains graphics that are used in the company newsletter, print drivers for the Epson Laser printer and the Hewlett-Packard Laser Printer, and a patch for MS Windows, Z-100 Version, for use with a Logitech Mouse.

Supported Printers

In order to get the maximum use out of ShowOff, you should have a printer. Pictures are printed through the use of a screen dump command that you load before entering ShowOff. Like the MS-DOS PSC command, the screen dump is accomplished by depressing the SHIFT/F12 keys simultaneously.

The **README.DOC** file on the main program disk contains a listing of the screen dump files and the printers each file supports. There is also another disk, the printer support disk, which also contains numerous printer drivers. While I can't list all the supported printers, I can say that there are quite a few Epson printers, as well as color ink jet, and laser printers supported. Before ordering, I would highly recommend you write or call Hogware to determine if your printer is supported. It is Hogware's company policy that "Every ShowOff customer deserves a printer driver". Consequently, they will develop print drivers for anyone purchasing the program, providing they are provided with the appropriate programming codes from the printers user's manual.

Customer Support and Upgrade Policy

Customer Support. Hogware provides purchasers with good service after the sale. Quarterly they send registered users a 4-6 page newsletter entitled *ShowOff News*. This newsletter expains how to use the capture procedures and provides helpful hints on how to use the program. It also advises users of upcoming utilities, program improvements, and new print drivers.

Upgrade Policy. Hogware also has a good upgrade policy. There has only been one change to ShowOff since its first release. Owners receive a free update if they send in their program disk, or they can pay $3 for an update disk.

Suggested Program Improvements

The one feature ShowOff is missing is an obvious and documented method to edit a file once it has been placed in the temporary buffer or saved to disk. ShowOff will allow you to erase the entire screen by depressing "E" and filling the entire screen with a background color. You can also erase selected parts of the picture by defining an area, then filling it with the background color, and finally recreating the image. However, there is no section on editing pictures in the documentation. Currently, this information is spread throughout the manual. One suggestion is a feature where the cursor can be placed on a circle, ellipse, rectangle, or text string and the user has the option of editing or deleting it.

I would also like to see a larger status line located at the bottom of the screen, as opposed to the top of the screen where it's currently located. It would be worth giving up some drawing space to be able to read a larger status line with less difficulty. One annoyance is the screen flicker in the high resolution mode. It's a small price to pay for improved graphs.

Conclusion

Showoff is a good drawing program and certainly deserves user support. It is relatively easy to learn, however, documentation can be improved. User support will allow Hogware to continue to improve the program and develop print drivers for printers that aren't currently supported.

Products Discussed

Hogware Company
470 Belleview
St. Louis, MO 63119
(314) 962-7833

ShowOff - $79.00
ShowOff & Logitech C7 Mouse - $174.00
ShowOff also available for the Microsoft mouse, PC Mouse, HIPAD, and Summagraphics digitizers.

Demonstration Disk - $3.00
Art Utility Disk - $5.00
Summagraphics and HIPAD version of ShowOff - $89.00

Logitech Publisher - $74.00 *

How to Prepare Menu-Driven Include Files For MAGIC WAND and PEACHTEXT

Kirk L Thompson
#6 West Branch Mobile Home Village
West Branch, IA 52358

Despite the proliferation of "what-you-see-is-what-you-get" (WYSIWYG) word processors in the market place ("WordStar", in its many incarnations, is the archetypical example), many of us feel more comfortable with the older-generation technology represented by "Magic Wand" (under CP/M-80) and "PeachText" (under CP/M and MSDOS). One reason, certainly, is familiarity. It apparently doesn't even breed contempt!

A second may be that we enjoy the challenge which these two packages require of our imaginations. Keeping tabs on tabbing, over-hanging indents, margins, and such are grist for our creative mental mills. Having a properly-formatted file appear at the very first printout demonstrates our subtle sagacity at systematization, to say nothing of command recall.

But the best reason, I think, may be that we (and I certainly include myself here) haven't come close to utilizing all of the capabilities which the programmers built into these two. And one of the most neglected of these features is the include file.

To be sure, both you and I load existing ASCII files into ongoing projects such as this very article. Even before I started entering the text you see before you, I pulled in a format file to set up the options I prefer in my manuscripts. I'm also sure that some of you are including files for boilerplated letters.

But there is one area where the otherwise excellent tutorial and documentation for these programs falls down. That is the discussion of more sophisticated uses of the include capability. And, specifically, I have in mind menu-driven include files and their screen layout. So I will take you through creating a file of this nature. My example will specifically be the beginnings for a menu-driven, online, help file for your word processor. (More on that, below!) But the same principles I describe can be used for even more sophisticated boilerplate and online applications.

But there is one further thing I should note before I begin my discussion. My frame of reference is the 8-bit version of "PeachText." This upgrade for "Magic Wand" is available from a number of vendors, but in limited quantities. Among other things, it fixed several annoying bugs in PRINT.COM and added a few enhancements to EDIT.COM. These changes appear to be completely compatible, at the ASCII text file level, with the 16-bit version of the program. So if you are running the latter, I don't believe you will have any difficulties with my examples. However, if you are running "Magic Wand," some trivial changes may be required, but these shouldn't be too difficult. And finally, although I will call the word processor "PeachText" in the following discussion, I am, of course, also referring to "Magic Wand."

Include Screen Layout

Laying out the screen, whether as a menu or retrievable information, isn't covered at all in "PeachText's" documentation. This is regrettable since this is, in fact, the most difficult part of include file preparation. And it revolves around how the word processor handles the text screen.

"PeachText" ignores the terminal's 25th line completely. Instead of using it as the "message line" (where the text moves when you press the "full insert" key), the editor uses the **24th** line on the screen. This means that you only have 23 usable lines of text on the cathode-ray tube (CRT) at any one time. The 24th is permanently reserved for system use.

When pulling in an include file of any kind, this shrinks by either one or two **more** lines. "PeachText" uses the 24th line for its include prompts, but also automatically reserves the line directly above the prompt for blank space between the include file's text and its own prompt. You will see how this works when you try displaying a long ASCII file with the include function. The final wrinkle is that using the "page feed" feature for formatting include files removes a **second** line from effective use.

You can check this out for yourself by typing in the material in Figure 1. Just bear in mind, of course, that the tilde character (~) is the mark for a carriage return and the up-arrow or caret (^) denotes the "page feed" key. Be sure that there are only 22 lines between the up-arrow and the top of the file. Now, end the edit session, recall the editor with any work file, and include the file you just created. Slowly press carriage returns to display the second page of the file. Notice, when you do this, that there is one entirely blank page displayed before the second page appears. This blank page is actually the ASCII form feed character (decimal 12, hex 0C), which is what "page feed" is! When it encounters it, the editor fills out the screen with carriage returns. Now re-

```
This is a simple text file~
~
to demonstrate what happens~
~
when more than 21 lines~
~
are included in a "PeachText"~
~
include the file above the~
~
"page feed" control~
~
character. As you will~
~
see, the editor inserts~
~
an extra carriage return~
~
between your typed~
~
text and the "page feed."~
22nd line of text~
^Second page of text which~
~
will actually be displayed~
~
as the third!~
```

Figure 1: This short file demonstrates how include files formatted with the "page feed" character limit page length to 21 lines.

```
                    HELP SCREENS FOR PEACHTEXT                    \
                    ==========================                    |
                       MENU AND INSTRUCTIONS                      |
                                                                  |
                            Page 1 of 2                           |
                                                                  |
      k1      Function keys : moving around your document.        |
      k2      Function keys : insert/delete/miscellaneous keys.   |
                                                                  |
      e1      Edit commands  :  set tabs and line width.          |   First
      e2           ''        :  handling external documents.      |  > Menu
      e3           ''        :  block manipulations.              |   Screen
      e4           ''        :  set edit modes.                   |
      e5           ''        :  escape text screen and print.     |
      e6           ''        :  read/write large documents.       |
      e7           ''        :  spool print, save, and exit.      |
                                                                  |
      --> Press <RETURN>; then <ESC ESC> for EDIT STATUS SCREEN,  |
          or selection from menu and <RETURN> for that screen,    |
          or merely press <RETURN> again for page 2 of menu...    /
^                   HELP SCREENS FOR PEACHTEXT                    \
                    ==========================                    |
                       MENU AND INSTRUCTIONS                      |
                                                                  |
                            Page 2 of 2                           |
                                                                  |
      p1      Print commands:  formatting (first of 2 pages).     |
      p2           ''        :  formatting (second of 2 pages).   |
      p3           ''        :  document header.                  |
      p4           ''        :  print and disk-save.              |   Second
      p5           ''        :  multipage commands.               |  > Menu
      p6           ''        :  screen display.                   |   Screen
      p7           ''        :  variables.                        |
      p8           ''        :  recognition characters.           |
                                                                  |
      (Set HELP.DOC to top with I@)                               |
                                                                  |
      --> Press <RETURN>; then <ESC ESC> for EDIT STATUS SCREEN,  |
          or selection from menu and <RETURN> for that screen,    |
          or merely press <RETURN> again for next screen...       /
^k1                 KEYBOARD:  MOVING AROUND DOCUMENT               \
                         FOR H/Z-19 TERMINAL                        |
      Cursor Movement:                                              |
              o One line up           )                             |
              o One line down         )  ARROWS on                  |
              o One character left    )  keypad.                    |
              o One character right   )                             |
              o Tab right from left margin          TAB.            |
              o Backspace over previous character   BACKSPACE.     |   First
              o Fast left, top, and bottom          HOME on keypad. |  > Display
      Scrolling:                                                    |   Screen
              o Forward one line      f2.                           |
              o Forward one page      f4.                           |
              o Backward one line     f1.                           |
              o Backward one page     f3.                           |
      Full-Text Movement:                                           |
              o Top                   ctrl-T.                       |
              o Bottom                ctrl-B.                       |
      --> Press <RETURN>; then <ESC ESC> for EDIT STATUS SCREEN,    |
          selection from menu and <RETURN> for that screen,         |
          or merely press <RETURN> again for next screen...         /
^k2               KEYBOARD:  INSERT/DELETE/MISCELLANEOUS            \
                         FOR H/Z-19 TERMINAL                        |   Beginning
      Insert:                                                       |  > of Second
          o Character under cursor                IC on keypad.     |   Screen
                                                                    /
```

Figure 2. The first three-plus screens of a menu-driven include file for "Magic Wand" or "PeachText".

edit Figure 1, remove the twenty-second line, and repeat the include operation. You will notice that the blank page is no longer there.

What this means is that we are limited to **21** lines of text, maximum, per display when preparing cleanly-formatted pages of information. However, as I will demonstrate below, this number could actually be less, but for reasons not connected with how the editor handles include files!

Setting up the menu

Knowing that we are restricted to 21 lines is critical information. It places a limitation on the size of the menus we create and the information we can put on any given screen. And preparing the former will depend heavily on how much descriptive material we need for each menu selection.

For purposes of demonstration, I'm going to presume one line of text for each selection. ("PeachText's" manual illustrates a two-column arrangement for boilerplating letters.) For example, Figure 2 shows a little over three screens at the beginning of my online "help" file. The first two screens contain the menu for the file. (Of course, the curly braces and labels on the right are not parts of the screens. They are there simply to show the screen-breaks in the file.)

For this online file, I've broken the menu into two sections, as you can see. The first is devoted to information we need when editing a document and the second to embedded commands required by the formatting program. And while I could have scrunched both menus onto one display, I chose not to for aesthetic reasons.

One of those is legibility! The less information packed onto a screen, the more readable that screen is. Blank space, whether on the CRT or the printed page, improves readability. I'm sure you've noticed how much easier on the eyes a page with wide margins is. Compare that with some of those manuals you certainly have which are solid text from top-to-bottom and side-to-side on each page. Or imagine a newspaper which has no columns across the page; rather, it has solid text! But legibility is worth a little extra paper or, in our case, kilobytes in the file.

A second reason is to make the file as "friendly" as possible. For that reason, I include three lines of instructions at the bottom of each and every screen telling us what to do next. These inform us how to return to the "edit status screen," move to a particular screen of information (the reason for the file in the first place), or proceed to the next display. These instructions clarify "PeachText's" prompts. The include function certainly wasn't designed with online help files in mind!

And third, screen titles and page numbering (as illustrated for the menu) tells us where we are. Like the exit instructions, they improve the "friendliness" of the file.

Before moving on, I should make two brief observations about the menu. The first is that its second page is not part of the menu-selection process. You reach it from the first by simply pressing the **RETURN** key twice. Notice in the figure that the up-arrow which separates the two isn't followed by a screen label.

The second concerns the characters we select for menu options. These could be any alphanumeric characters we might like. But I suggest lower-case if we use letters. The reason is because the include function is case-sensitive. Upper-case will not find lower-case labels in the file, nor will the reverse. Besides, lower-case is easier to type at the keyboard; shifting is unnecessary.

Addressing Screens

Once we've settled on the scheme for identifying information screens and laid out our menu, we have to label each display screen in the file. We do this by pressing the "page feed" key on what will become the top line of the particular screen and follow it immediately with the characters from the menu. And since "PeachText" ignores everything beyond the first space in the label, we can put things such as screen titles on this same line. This is illustrated in Figure 2.

There, you will observe that the first two information screens are labeled "k1" and "k2". The up-arrow (or caret) immediately preceding the label is, of course, the character representing "page feed." Actually, this special character and the label can appear almost anywhere on the line in front of the screen title. By convention only, I keep them left-flush.

Preparing the Information

The same aesthetic reasons for laying out the menu apply, as well, to the information screens. We should make the information as digestible as possible. That means we include a bit of blank space to reduce eye strain and also consider each screen's user-friendliness.

For the latter, we should left-justify text. You can see that I've done this in the figure. I've also made extensive use of indents to set off sub-headings. If I had include numbers, they would have been right-justified, with the decimal points lined up.

Moreover, I suggest that we mix upper- and lower-case. Something written entirely in upper-case is **very** hard to read! Upper-case titles are all right, as you can see in the figure, but most of each screen should be lower-case.

Finally, include files aren't automatically word-wrapped at the end of each line, either. So if we need a short paragraph of text, we need to include carriage returns at the end of each and every line.

HELP.DOC

As a complete example of the style of online documentation which we have discussed, above, I would be happy to supply you my HELP.DOC for "Magic Wand" and "PeachText" under either CP/M-80 or MSDOS. With this help file, you no longer need to refer to the manual or even a quick reference card. All the information is online, just like it is with many of those big and expensive packages out there for the Macintosh or IBM-compatible.

But I must also observe that this file has three distinct disadvantages! One is speed. If you are running exclusively on floppy drives, you will find this help file quite slow. The reason is the size of the file; it occupies 14 kilobytes and it could take up to 20 seconds for the word processor to wend its way down to the last page of information. Performance improves dramatically on RAMdrive or hard disk. For example, on C.D.R.'s "SuperRAM 89" RAMdisk system on my 4 MHz '89, it only takes 4 seconds to find the last screen (of 19) in the file. Retrieval on hard disk will be comparable.

A second disadvantage of the file is that it

isn't context-sensitive. Unlike many on-line systems, which bring up a relevant help screen directly related to where you find yourself at a loss for what to do next, HELP.DOC must be specifically called from the word processor's "edit status" screen. This is done as you would with any other include file: the "I" command followed by the file's name. A second "I" command presents you with the first of the two menus.

The third disadvantage is inherent in the word processor, itself. It keeps track of where it is in an include file. So when you re-enter a file you've accessed before, you are taken immediately to the screen **following** the last one you viewed. If you wish to return to the menu, you must first enter \bf\"I@"\bf@\ to reset the file to the top, then recall it with "I".

***Bold-face recognition character is turned off, then on, to print "I@" on the last line.

But if you are interested in acquiring HELP.DOC, there are two ways you can get it. The first is directly from me. I would be happy, as I mentioned, to send it to you for a very small charge. I can access most CP/M-80 **40-track** formats, including standard hard-sector and double-sided soft-sector. I can also format, read, and write MSDOS 2.0's 360K format from CP/M for those of you running '100's and '150's. But I **can't** format (only read and write) a ZDOS-compatible disk, so if you require that, you will have to send a formatted disk. If interested, write me at the address below for further details.

The second method is to download it from HUG's PBBS at (616) 982-3956. You'll find it, and a short documentation file (HELP.PRN), in the CP/M catalog.

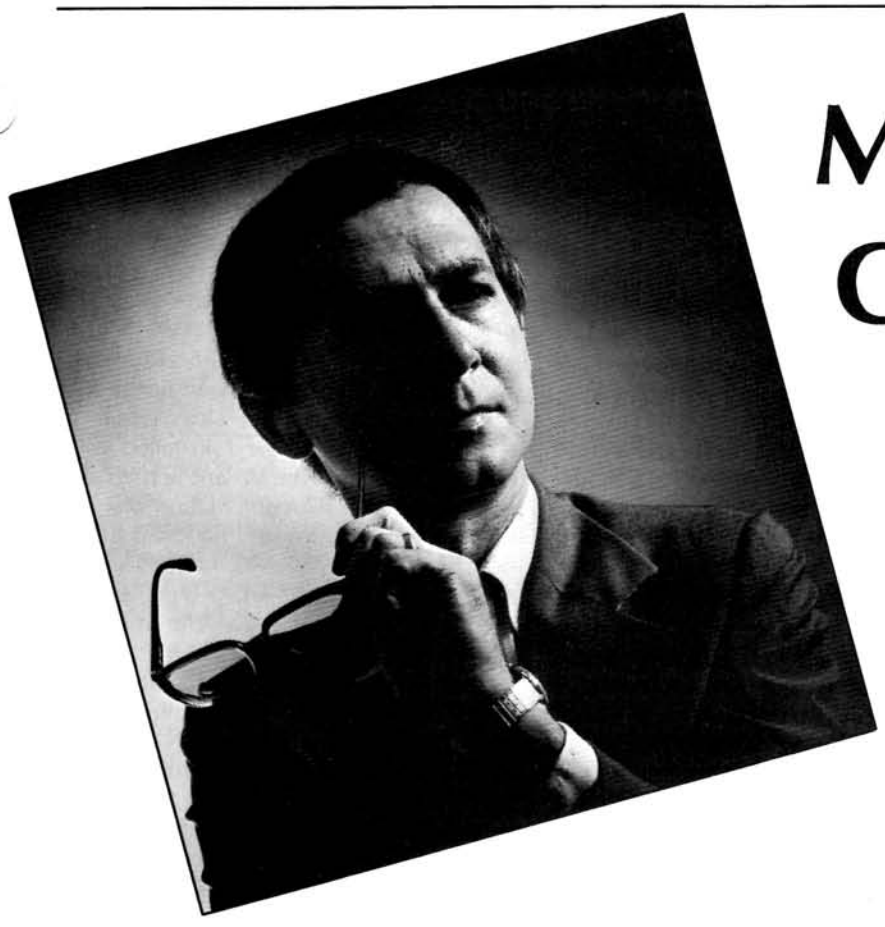
So, to wrap everything up, knowing how to set up menu-driven include files under "Magic Wand" and "PeachText" considerably enhances the utility of these two older-generation, but still very capable, word processors. In fact, a menu-driven structure opens up a veritable Pandora's box of potential applications which I, for one, plan to expedite posthaste!

Additional Information

HELP.DOC.
Kirk L. Thompson
#6 West Branch Mob Hom Vil
West Branch, IA 52358

"SuperRAM 89" RAMdrive system for CP/M and HDOS 2.0 on '89/90.
C.D.R. Systems, Inc.
7210 Clairmont Mesa Blvd.
San Diego, CA 92111
619-560-1272

*



Mainstream Computing

Joseph Katz
103 South Edisto Avenue
Columbia, SC 29205

I hadn't intended to get an 80386 computer just yet, but First Capitol Computer announced a unique Z-248-to-Z-386 offer it calls an "upgrade." The more I thought about it, the better the offer seemed. Besides which, we had a spare Z-248 I bought last year for a project recently completed. No major bridges would be burned behind us if it itself became a bridge to the current new generation of microcomputers. So I took First Capitol up on its offer.

I'm glad I did. What Tom Jorgenson of First Capitol Computer calls an "upgrade" turned out to be essentially a brand new computer at a very good price. When the Z-386 arrived and I began exploring it, I realized that my reservations had been influenced by the torrent of words about 80386 computers in general, the river about IBM's PS/2 computers in particular, and only the merest trickle that has appeared about Zenith's Z-386. Most of the ink and street talk has represented the Zenith 80386 as just one of many without IBM's proprietary Micro Channel Architecture ("MCA"). I've seen no applause for the 32 bit buss the Z-386 uses instead. At least to my mind right now, that particular 32 bit buss in the Z-386 is a strength, not a weakness, and a reason why the Z-386 is a good choice for people like me who want to ease across instead of jump in blindly.

What had bothered me was the idea, which came from every direction, that I'd have to make major investments just to get a Z-386 up and runnning. For instance I'd read and been told that the additional memory boards from my Z-248 would be useless in a 386 machine because the buss was wrong for them. I'd also heard that Zenith's Z-386 had one XT-style and one AT-style slot available for others of those boards, but two slots were not enough for the host of boards my computers need so they'll do what I want. I most certainly was not prepared right now to invest the vast resources it seemed would be required to give a Z-386 just the capabilities of the Z-248 it would be replacing.

Indeed, I know that eventually I'll need products especially for the 80386 to get its full benefits. But I'd have been willing to invest a reasonable amount to get just its speed now. With equal willingness would I postpone benefits from the rest — especially because there's not much of them now anyway — if I could postpone their costs too. But I was unwilling to take several expensive steps backwards before I could even get to the new starting line.

I wish I'd known that the Z-386 is the right machine right now for someone like me. There's nothing wrong with the computer, only perhaps a failure in communications about it. I hadn't heard it and I don't have the technical manual yet, but Zenith evidently has designed the 32 bit buss in the Z-386 to give it a remarkable degree of

Address all correspondence to me at 103 South Edisto Avenue, Columbia, SC 29205. I'll try to answer letters accompanied by a self-addressed stamped envelope, but my volume of mail is too heavy for me to promise. Unless it specifically says otherwise, I'll assume the right to publish your letter (edited, if I think that appropriate).

"downward compatibility" with the IBM XT- and AT-compatible computers of previous generations. Had I known so sooner, I'd have had a Z-386 sooner. Most of my reservations crumbled during a few days of the poking, prodding, and exploring I do when tailoring a new computer for my own needs. Now I'm delighted to have the Z-386 and I'm grateful that First Capitol's offer tantalized me into having one.

The Z-248-to-Z-386 Upgrade

Tom Jorgenson of First Capitol Computer calls its offer an "upgrade," but I think it's too modest a description. Upgrades improve your present computer, so no matter how substantial the improvement, you still end up with basically the same computer. There's nothing wrong with that, of course, but that's not the result here. With First Capitol's deal you end up with a new computer: not an upgraded Z-248, but a new Z-386 with some parts from your Z-248. Those compromises might not matter to you, if your aims parallel mine. So the offer seems to me different from an "upgrade" (the reverse of it, I think), although that's what Tom calls it. You can decide the semantics for yourself.

According to the terms of First Capitol's offer, I phoned for a time when I could ship my Z-248 to them in Missouri. What they did when they got it was install its drives and video display adaptor, plus a couple more things I ordered, in a brand new Z-386. Then they tested that new computer extensively to make sure it worked before shipping it back to me in South Carolina. So instead of ending up with new Z-386 boards in my old Z-248 I got a real Z-386 — power supply, case, keyboard, serial number, warranty, and all.

That's not merely an "upgrade," is it? Of course, it's not a "trade-in" either. Maybe it's a "transformation." Whatever the real right term for this deal, I think it's a really good deal. I'm happy with it. The price right now is $1,995, plus the cost of any extras you might want, plus 2% of the total for shipping. The Z-248 you ship to Missouri must be in perfect working condition.

16 Bit Memory for the Z-386

Of course, it will take Zenith's fast 32 bit memory boards to get the most out of a Z-386. Right now, though, there's no available software I need that takes good advantage of such fast Extended or Expanded memory. There's little likelihood of any such software until both OS/2 and the Presentation Manager have been out for a while. That combination operating system is not out yet, although it's talked about a lot. The talk is lovely and doesn't cost a penny to hear. What I'm looking at, however, is a very fast MS-DOS computer with the potential to be much more. That potential is very nice and doesn't cost much to contemplate. Reality, however, is here right now and it's what should not be overlooked: the Z-386 is a 386 machine that runs at 16 MHz with no wait states. That's a relative speed of 19.7 MHz when compared, according to the formula I told you about last time, to IBM's PS/2 Model 80, a 386 machine that runs at 16 MHz with 1 wait state.

MS-DOS still addresses 640 KB of base RAM. Anything above that Conventional Memory must be used as either Extended or Expanded Memory. With the costs of RAM sky high today and sure to go still higher before the price begins receding, if it does, I have no choice but to settle for the real all-around benefits of a fast 386 MS-DOS computer with half the top potential speed on Extended and Expanded Memory that's of limited use to me now anyway.

I'd heard you can't use the 16 bit Extended or Expanded memory boards from a Z-248 or Z-241 in a Z-386. Of course you can: that's a real benefit right now of Zenith's 32 bit buss. The *High Performance Workstation Owner's Manual* (Zenith's zippy moniker for the Z-386 is "High Performance Workstation") says so, although you have to read slowly and carefully in this booklet to find where it does, then do some tough interpretation to recognize that it does. At one time or another since the Z-386 arrived, I've installed a basic 1 MB BocaRAM (with and without a 1 MB BocaRAM daughter board), a 2 MB BocaRAM/AT, a 1.5 MB Cheetah Combo/80, and all of them together.

Here's how I did it. I had to reset both the BocaRAM/AT and Cheetah Combo/80 so they wouldn't use 128 KB of their RAM for backfilling Conventional Memory as was necessary on the Z-241 and Z-248, which like most AT-compatible computers were shipped with only 512 of RAM. The Z-386 has 1 MB of RAM, of which 640 KB is used for the full complement of that essential Conventional Memory. A 128 KB overlap would have created a memory conflict. At the same time, I set the switches and jumpers on those boards for whatever combination of Extended and Expanded Memory I wanted. Then I set the appropriate DIP switch on the Z-386 CPU board for the total amount of Expanded Memory on those boards. (Extended Memory is assigned through the Setup routine in the ROM Monitor.) And then I inserted the boards in the computer's available slots.

It made no difference whether I used an XT, AT, or 32 bit slot: so long as the board physically fit into a slot (the BocaRAM/AT and Cheetah Combo/80 won't fit the XT slot), it worked in that slot. When I rebooted the computer I installed the necessary EMM driver. What I settled on as software, after experimentation with all the boards and software I had, was Boca Research's BRATEMM.SYS memory manager, BRATDISK.SYS RAM disk driver, and BRATDISK.COM RAM disk program, all of which is supplied with their BocaRAM/AT. When I called Sylvia Fagiani of Boca Research to tell her what nice products she had, she scolded me for forgetting to register the BocaRAM/AT, said I had missed a couple of important software upgrades all registered owners had been receiving free, and told me she'd send the latest version of the software overnight if I promised to try it right away. I did. She did. I did. The Version 1.4 software she sent supports the LIM ("Lotus/Intel/Microsoft") 4.0 specifications, which makes me sorry I hadn't registered the board right away. I had meant to. I've settled on the plain vanilla BocaRAM board and its daughter board, giving me a total of 4 MB plus what remains of the 1 MB on the Z-386 memory board, and it's in Slot 10, the XT slot, at the moment.

More Ports for the Z-386

The Z-386 comes with one serial port and one parallel port on the I/O board in Slot 3. I always need more ports, especially more serial ports. I could use eleventy-seven if I could get them. Because the Z-386 accepts AT-compatible I/O boards, I simply took those from my Z-248 and put them in the new machine. Slot 8, one of the 32 bit slots, now has one of Boca Research's Boca/IO AT boards configured for COM2 and LPT2. Slot 9, the AT slot, has a Zoom/Modem 2400 configured as COM3. Slot 7, another of the 32 bit slots, has the board for a LogiTech buss mouse. (My Apple LaserWriter is on COM1, as always.) All I did in each instance was take the board from a Z-248 and put it in the Z-386. No reconfiguration was required.

And still I need more serial ports. Bruce Denton of D-G Electronic Developments has just sent me an early copy of his new DoubleCOM board, which should help satisfy my serial hunger in an unusual way. DoubleCOM is a pair of switch-selectable serial jacks on a single board. You configure the board itself as either COM1 or COM2. Then either jack can be used as that port. Say you need a total of three serial devices available, but no more than two of them need be active simultaneously. If they're a serial mouse, serial printer, and modem, for example, you'd probably use the mouse and the printer together, maybe the mouse and the modem together, but never the printer and the modem together. With the DoubleCOM configured as COM2, you can plug the printer into one jack and the modem into the other. Your application software for both of those devices would be configured for output to COM2. Then you simply switch between the two jacks to use the device whose turn has come. There's a hardware switch, a toggle, between the two jacks, and some good programming by Bill Parrott for when software switching is handier: you can decide between dip or chip. DoubleCOM is an ingenious solution to a serious problem. Anyone who does desktop publishing on an MS-DOS computer ought to know about it.

Understand that what I have now is what I wanted in the first place: a fast, versatile MS-DOS computer with a great deal of potential. It's about twice as fast as the Z-248 it replaced. For a rough-and-ready indication of the increased speed, the Z-386 scores around 17.2 with the SI program from Version 4 of the Norton Utilities. The Z-248 scored around 9.2 after all the tricks at my disposal.

That speed increase really does matter to me with the kinds of applications that most often concern me: mainly desktop publishing, graphics, programming, and original writing. It's not that the Z-248 was too slow or that I'm glad to be rid of it. As I began by saying, the Z-248 as I had configured it did what I needed at a speed with which I was content. That's one reason why I felt I could wait out the next year or so without anxiety. But when XyWrite III Plus does a complete spell check on one of these columns in five seconds, when Aldus' PageMaker and Xerox's Ventura Publisher snap, crackle, and pop, and when Microsoft's assembler and C compiler both finish their work in half the time, the machine becomes more transparent.

See you later.

Products Discussed

BocaRAM
BocaRAM Add-On
BocaRAM/AT
Boca/IO AT
Boca Research Inc.
6401 Congress Avenue
Boca Raton, FL 33487-9983
(305) 997-6227
(Prices depend on current RAM prices.)

DoubleCOM $149
D-G Electronic Developments Co.
700 S. Armstrong Avenue
Denison, TX 75020
(214) 465-7805

Z248-to-Z386 Upgrade $1,995
(plus shipping)
First Capitol Computer
16 Algana Drive
St. Peters, MO 63376
(800) TO-BUY-IT

*

CHECK THIS AD FOR MANY NEW ITEMS

For the past several months the world market price of 256K RAM chips has been climbing. Supplies are so short that major manufactures are facing tough decisions. We at PAYLOAD have established the following policy which will be in effect until RAM prices stabilize. We have removed all 256K RAM chips from kits normally containing them. In order to obtain current RAM prices please call us by phone. We are very sorry for this inconvience, but it is the only way we can keep our other prices stable.

*** ZENITH SOFTWARE FOR THE *** Z-100 SERIES COMPUTERS

Zenith packages with software, manuals and registration cards for the original Z-100 computer series (not for the IBM compatibles). These prices are good until our current inventory is sold. Only about 100 of each remain.

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
MS-463-1	Z-Basic (16 bit)	$175.00	$12.00
MS-463-7	Multiplan	$195.00	$12.00
MS-253-1	Basic-80 (8-bit)	$175.00	$12.00
CD-463-2	Condor File Manager	$299.00	$12.00
LT-Z100	All 4 Listed Above	$819.00	$40.00

*** SOFTWARE FOR IBM *** COMPATIBLE COMPUTERS

The Zenith version of Microsoft Windows is from a special purchase. After 300 packages are sold our price must revert to $89.50. The LEXISOFT packages are being closed out below cost and only a few are available for this sale.

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
MS-5063-30	Microsoft Windows	$ 99.00	$ 38.00
NU-411	Norton Utilities	$99.00	$ 69.00
NU-413	Norton Utilities Adv.	$150.00	$ 99.00
WP-528	WORD PERFECT	$495.00	$259.00
SR-3950	dBase III plus	$695.00	$479.00
BO-290	QUATTRO	$195.00	$159.00
LO-311	LOTUS 123	499.00	$359.00
MS-450	MS DOS 3.3 with GW BASIC		$124.00

ZFL-181-93 Laptop Portable with the amazing Supertwist LCD backlit screen. 640K RAM, two 720K 3.5" disk drives, clock, P & S Ports $1649.00

*** VIDEO MONITORS ***

ZCM-1490	ZENITH Color Flat Screen VGA	$718.00
XC-1409C	Mitsubishi Color CGA 640x200	$296.00
XC-1410C	Mitsubishi Color EGA 640x350 0.40mm.	$419.00
XC-1430C	Mitsubishi Color EGA 640x350 0.31mm.	$489.00
AUM-1371	Mitsubishi Color EGA Multi-Scan 920x580	$529.00

In addition to the above, we can supply Magnavox and Samsung monitors in monochrome and color models.

*** UPGRADE ACCESSORIES FOR *** ZENITH PC COMPUTERS

Z-171 CMS Lite Drive 10 Meg Hard Disk Internally mounted and with automatic head parking. This hard disk replaces one floppy drive and runs on either AC or battery. List price $995.00 but now available while supplies last at . $495.00

AUTOFAX IMAGER Video Tape Backup System Use your VHS or Beta Video Recorder to backup your hard disks. With this interface card and cable set connected to your video tape recorder, you can backup up to 110 Megabyte on a standard video tape. The required software is included. $212.00

SmartWatch from FBE Research Installs in ROM Socket on CPU Board in Zenith computer series Z-100/138/148/150/160. This clock/calander contains a ten year battery and keeps your computer informed of both time and date at each boot-up. Instructions and software included. $38.00

Z-150 Memory ZP640 Plus This PAL chip replaces the one on your Z-150/160 memory board and allows the use of 256K RAM chips to replace two banks of the existing 64K RAM chips. To complete the installation you will need 18 pieces 256K 150 ns RAM chips. This increases 128K memory to 640K or 256K memory to 704K. All chips plug into your existing Zenith Memory Board. ZP640 PLUS $18.00

Z-150 Series Hard Disk Drive Kit Includes new generation High Speed (28 MS) Seagate Drive with Auto Park heads. Each kit is complete with controller card, cables, hardware and instructions to mount the Hard Disk under your two floppy drives in the Z-150 series computers. Order 21 MEG ST-125/150 Kit $329.00 or 32 MEG ST-138/150 Kit $383.00

Z-148 Hard Disk Drive Kit Includes the Hard Disk Drive and controller in the kit above plus the PTZ-148 Expansion card described below. Each kit includes all cables, hardware and instructions required to replace one floppy drive with a high speed low power Hard Disk Drive. Order 21 MEG ST-125/-PTZ-148 Kit $438.00 or 32 MEG ST-138/PTZ-148 Kit $492.00 using an Expansion Card sold below.

PTZ-148 Expansion Card for Z-148 Includes 2 expansion slots plus a clock/calander. $118.00

VMM 150 Video Eliminator for Z-150 computers. Allows use of EGA color card $45.00

*** UPGRADE ACCESSORIES FOR *** Z-100 SERIES COMPUTERS

High Density 1.2 Meg Drives. External floppy drive set-up complete with drive, power supply, case and cable. Ready to connect to your 8" floppy controller $224.00 . . . . . .Dual Drive Unit $369.00

SmartWatch by FBE Research. If you don't have a clock for your Z-100, get this one. More details under Z-150 upgrade listings $38.00

Gemini Emulator Board. Makes the Z-100 compatible with the IBM PC library of programs. $432.00

UCI EASY PC. IBM PC Emulator. Makes your Z-100 IBM Software Compatible. Full 8 MEG operation, color graphics and audio compatible. Retail $699.00 . . . . . Payload $477.00

UCI Easy87. Add an 8087 Numeric Coprocessor. $69.00 for the board without an 8087 Chip. With 5 MEG 8087 $188.00 or with 8 MEG 8087 installed $234.00

ZMF100A by FBE Research. A modification package which allows 256K chips to be used on the old-style motherboard to reach 768K. Simple assembly with no soldering or trace cutting. Compatible with Easy PC and Gemini Emulator. . . .$60.00 Requires 27 of the 256K RAM chips to complete the kit.

UCI Memory Upgrade Pal Chip Set For the Z-100's with the newer motherboard part number 181-4918 or greater. This chip set allows the installation of 256K RAM chips on the motherboard. With the addition of 27 256K RAM chips, a total memory or 768K is obtained. PAL Chip Set $64.00

UCI Memory Upgrade Card We recommend this one highly. The board has sockets for up to 2 MEG of RAM. With no RAM installed $288.00. Add $35.00 for EasyDrive RAM Drive Software if desired. Either 64K or 256K RAM chips may be used to complete this kit.

UCI EASY-I/O S-100 board that provides IBM PC communications port compatibility with your EasyPC. Easyl/O-1, One Serial Port $91.00. Easyl/O-2, Two Serial Ports, One Game Port, Clock- Calendar $127.00

UCI EasyWin Winchester Drive Systems at reasonable prices. Complete Hard Disk Systems for mounting inside your Z-100. Systems complete with Seagate Drives, 21 MEG $598.00, 31 MEG $634. System without Drive $317.00

CDR Z-100 Speed Module Run your Z-100 Computer at 7.5 MHz. Installs easily with no soldering. Externally switchable between Speed and Normal mode. Payload $44.00

*** FLOPPY DISK DRIVES ***

MITSUBISHI MF501	5.25" 48 TPI DS/DD 320K/360K	$ 95.00
MITSUBISHI MF504	5.25" High Density 360K/1.2 MEG	$124.00
MITSUBISHI M-353	3.5" in 5.25" frame 720K	$103.00
MITSUBISHI M-355	3.5" in 5.25" frame 1.44 MEG	$139.00
	M-355 Software Driver	$ 19.00

M-355 runs on AT compatable or special controller only.

** SEAGATE HARD DISK DRIVES ***

ST-125	21 MEG, 28 MS, Auto Park Heads	$ 275.00
	With Western Digital Controller & Cables	$ 329.00
ST-225	21 MEG, 65 MS,	$ 259.00
	With Western Digital Controller & Cables	$ 313.00
ST-138	31 MEG, 28 MS, Auto Park Heads	$ 329.00
	With Western Digital Controller & Cables	$ 383.00
ST-238	31 MEG, 65 MS, RLL	$ 278.00
	With Western Digital RLL Controller & Cables	$ 323.00
ST-251	42 MEG, 28 MS, Auto Park, Software	$ 409.00
	With Western Digital Controller & Cables	$ 468.00
ST-4096	82 MEG, 28 MS, Auto Park, Software.	$ 797.00

V-20 Chips High Speed NEC V-20-8 8088 replacement. These run at up to 8 MEG and are said to increase CPU speed 10- 30%.....Payload $14.75

PAYLOAD COMPUTER SERVICES

15718 SYLVAN LAKE
HOUSTON, TEXAS 77062
PHONE (713) 486-0687

MasterCard VISA

Your satisfaction is guaranteed. All hardware carries a 90 day Payload warranty. VISA ar MASTERCARD orders welcome with no surcharges. Add $5.00 to all prepaid orders f handling and shipping in Continental USA, we pay the balance. Actual shipping costs for foreign, overseas and net billing orders to approved accounts. We accept purchase orders from schools, government and approved accounts. Mail or Phone your order for prompt service. Texas residents please add 8.0% state sales tax.

POWERING UP

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

Using File Names And Subdirectories

As you learned last time, file names are a critical part of using a computer system. You'll remember that a file name consists of a filename and a file type or extension in the general form of: *filename.typ*. One of the most important ideas presented in the last article was the use of some kind of standard for creating your own file names in your system. There is an important reason for this.

More on File Names and File Types

All computers rely on a file name to locate specific data on a disk. If you have a standard way of naming a file, it will make it much easier for you to locate and identify a specific file. And yes, DOS also has a set of naming standards that specifically apply to the file type. In particular, DOS reserves the use of certain file types — COM, EXE, and BAT — for a special purpose, and you should never use those file types for your data files. The next article will discuss those file types in more detail, but for now, just accept the fact that you should not use them for data files.

DOS has an interesting feature associated with file names that can help you take advantage of your file naming convention. Take a DOS system disk that you use to boot your computer and insert it into drive A. Then, enter the following command:

```
A>DIR ????????.COM
```

Notice that eight question marks are typed, then the period for the end of the filename followed by the COM file type. The actual command you enter is shown in italics, and you must press RETURN to execute the command as usual. When you do, you will see a display something like Figure 1.

The most important thing to notice about Figure 1 is that all files displayed by the command have the COM file type. Your system disk may have more files, and you will probably have a different number of bytes free shown by the DIR command. But what did all the question marks do in the "DIR ????????.COM" command line? Perhaps you will want to fire up your com-

```
 Volume in drive A is SYSTEM-3-21
 Directory of A:\

APPEND   COM     1725   5-01-87  12:00a
APPLY    COM     2165   5-29-87  10:21a
ASGNPART COM    16484   5-11-87   3:57p
ASSIGN   COM     1523   5-01-87  12:00a
BACKUP   COM    24715   8-24-87  10:31a
RTCLOCK  COM     2432   7-23-87   2:30p
SEARCH   COM     4343   5-29-87  10:22a
SHIP     COM      914   4-11-86   8:51a
SYS      COM     5233   7-27-87   1:58p
TREE     COM     1827   1-28-86   9:52a
ZSPOOL   COM     3004   5-07-87  10:29a
       11 File(s)   1158912 bytes free
```

Figure 1
File Display for
DIR ????????.COM Command

puter and try the following DIR commands to see how they work.

When used in a DOS command line this way, the question mark (?) is called a WILDCARD, and it means to *match any character at this position and only this position*. In other words, the question mark wildcard is positionally dependent, and the actual number of question marks in the command line will determine what files are displayed by the DIR command.

Now let's be a little selective and display only those files beginning with the letter A followed by seven question marks, a period, and the COM file type. Remember that the filename piece of the file name is limited to eight characters. If you enter the following command:

```
A>DIR A???????.COM
```

Then you will only see the file names as shown in Figure 2.

```
 Volume in drive A is SYSTEM-3-21
 Directory of A:\

APPEND   COM     1725   5-01-87  12:00a
APPLY    COM     2165   5-29-87   0:21a
ASGNPART COM    16484   5-11-87   3:57p
ASSIGN   COM     1523   5-01-87  12:00a
       4 File(s)   1158912 bytes free
```

Figure 2
File Display for
DIR A???????.COM Command

As shown in Figure 2, only file names beginning with an A are displayed. Now try the DIR command with the letter A followed by SIX question marks as follows:

```
A>DIR'A??????.COM
```

With only six question marks, you will see the file names as shown in Figure 3.

```
 Volume in drive A is SYSTEM-3-21
 Directory of A:\

APPEND   COM     1725   5-01-87  12:00a
APPLY    COM     2165   5-29-87  10:21a
ASSIGN   COM     1523   5-01-87  12:00a
       4 File(s)   1158912 bytes free
```

Figure 3
File Display for
DIR A??????.COM Command

The important thing to see is that the ASGNPART.COM file was not listed by the command because you specified an A followed by six question marks for the filename for a total of seven characters. The ASGNPART filename has eight characters, and since the question mark wildcard is positionally dependent, there was no match.

You will see the same kind of display if you enter a "DIR ?????.COM" (five question marks) command since all of the remaining filenames have six or less characters. Now use the same format with only four question marks as follows:

```
A>DIR'A????.COM
```

In this case, the only matching file name is APPLY.COM because the other file names are longer than five characters — the letter A followed by four question marks.

By careful use of the question mark wildcard, you can selectively display just about any combination of file names on your disk. And although the examples have shown the question mark in only the filename, the question mark can also be used in one or more places in the file type.

In that case, all file names on a disk could be represented by the "DIR ????????.???" command. Each question mark represents every possible position in the general file name form of *filename.typ* — eight question marks, a period, and three question marks.

More Wild Stuff

While the question mark wildcard is useful in many cases, it was introduced to show you how a wild card is used. Again, the important point about the question mark is that it is positionally dependent. Each question mark in a "????????.???" file specification will match any and all characters in every position for a file name. Fortunately, we don't have to enter that every time we want to work with all files on a disk — the asterisk (*) wildcard may be used instead.

The asterisk wildcard means to *match all characters beginning at the current position, plus any that follow up to the end of the filename or file type*. Instead of using the "????????.???" file specification, you could simply enter an "*.*" file specification instead. Or you could display all of the COM files on your system disk by using

```
A>DIR*.COM
```

This command will produce exactly the same results as the "DIR ????????.COM" command did in Figure 1, but you didn't have to type the question mark eight times. Can you see how to use the asterisk wildcard to produce the same results displayed by the "DIR A???????.COM" command used for Figure 2?

The answer is to use the "DIR A*.COM" command to display all of the COM files beginning with the letter A. Again, notice that the asterisk "replaced" the seven question marks thus reducing the required keystrokes to display the requested files. In other words, the "DIR A???????.COM" command is exactly the same as the "DIR A*.COM" command. As a matter of fact, that statement has more truth to it than you might expect.

DOS actually "expands" the asterisk wild card into question marks. Even though you may actually type a "A*.COM" file specification, DOS changes that to be "A???????.COM" before it begins looking for the files. Or, if you type "*.*", DOS expands it to the most general form of the file name, which is "????????.???", that has been mentioned.

Talking About File Names

There are two ways to talk about and use file names. The first way is to use a completely specified file name, such as HUG88-14.LTR. If you want to be more precise as to where the file is, you might also want to precede the file name with a drive letter so that it becomes something like B:HUG88-14.LTR. Virtually all programs that accept file names as part of the command syntax will accept a completely specified file name, and some programs require a completely specified file name. For example, a word processor requires a completely specified file name because it does not make sense to use a file name that contains any wildcards. How would

the program know which file you wanted to edit?

In these articles, I have and will consistently refer to the general form of the file name (two words) as containing the filename (one word) and an optional file type so that a file name is understood to be in the form of *filename.typ*. You may find some documentation that refers to only the required filename part of the file name because the file type is optional.

When discussing the two types of file names, I like to refer to a completely specified file name, such as HUG88-14.LTR, as an unambiguous file name or ufn. AN UNAMBIGUOUS FILE NAME or UFN does not contain any wildcards, and it is completely specified. A single letter or number can represent a ufn, and "A", "B" or "1" are valid file names for DOS, although they are generally cryptic and meaningless to most people.

The second way to talk about file names is to use wildcards. If I wanted to see a directory list of all letters I wrote to HUG in 1988, I could use the command "DIR HUG88*.*" or "DIR HUG88???.???". Because these file names contain the asterisk and question mark wildcards, they can refer to a wide variety of file names. As a result, it is easy to think of this kind of file specification as an ambiguous file name or afn. An AMBIGUOUS FILE NAME or AFN contains one or more of the asterisk and/or question mark wildcards. Moreover, it is important to recognize that, by definition, any afn file specification usually includes one or more unambiguous file names.

For example, consider the afn specification in the "DIR HUG88*.*" command. Assuming that we had the right disk, that command would display directory entries for letters written to HUG beginning with HUG88-01.LTR up to and including the HUG88-14.LTR. The important point is that the afn specification includes all of these unambiguous file names.

I have introduced these terms because it is important to know which commands accept an afn specification and which commands accept only a ufn specification. Most of the common DOS commands will accept an ambiguous file name because it is useful for manipulating files. Most other programs, like word processors or spreadsheets, will only accept an unambiguous file name without any wildcards.

Using File Names

The capabilities and uses of the wildcards have perhaps given you the reason for using a standard file naming convention in your system. It makes it easier to name and find files when you are looking for something specific. If you have a general standard for file names, you will be able to guess what the file name is even if you don't remember it. While that is more important if you have a hard disk system with lots of files on the hard disk, it is also important in a floppy disk system. In order to see how this standard might be used with file names, let's review the naming convention introduced in the last article.

A filename is in the form of

```
IIIyy-nn
```

where:
III = initials of organization or individual
yy = year created or scheduled (e.g. 89)
nn = sequence number beginning with 01

Figure 4
File Naming Convention

When I began writing articles for REMark nearly five years ago, I chose REM as the initials for those articles, and the very first article I wrote has a file name of REM83-11. When I began writing this series, I had to choose another "organization" name so that I could follow the standard shown in Figure 4. As a result, this particular article has a file name of POW88-08.

Because I have a standard file naming convention, I can use the wildcard concept to quickly list all files for a given year. If, for example, I want to see what articles I wrote in 1986, I can use the "DIR REM86", since I generally do not use the file type for my word processing files. I can also quickly locate personal letters to my sister Susan using the SLA* file name or my brother David using the DNA* file name.

When I started using personal computers a number of years ago, I was using a spreadsheet extensively to track my department budget. The planned budget for the year was BUD84.CAL with the CAL file type being unique to SuperCalc. Since all managers had to provide monthly budget reports on variances from the original budget, each monthly update would have a file name like BUD84-01.CAL, BUD84-02.CAL, and so on.

These file naming conventions were helpful even though I had each category of files on a separate floppy disk because I did not have a hard disk at that time. I had a floppy identified as "1984 Budget" for my company, "Personal Correspondence", and so on. When I was finally able to afford a hard disk for my computer, I simply transferred the contents of those floppies into a subdirectory for my system. And now let's take a look at subdirectories and how to use them.

More on Disk Directories

With the increasing use of hard disks, an understanding of subdirectories is extremely important to help you manage all of the files that you have. An understanding of subdirectories is important in any DOS-based computer because of a simple limitation for file storage on any type of hard or floppy disk.

As mentioned in the last article, every disk contains a disk directory which contains a list of table of contents for files on that disk. This "master" or ROOT DIRECTORY can contain a limited number of file names that is specific to the type and format of disks you are using in your system. For example, a standard 5.25" or 3.50" DS/DD (double-sided, double-density) floppy disk has a maximum of 112 root directory entries, while most hard disks have 512 root directory entries. The maximum number of root directory entries is important because it also defines the maximum number of files that may be stored on that disk. Let's see why that is important to you.

Consider the example of a 3.50" DS/DD disk that has a nominal capacity of 720 kilobytes. Let's say that you are using that disk to store personal correspondence created by your word processor and each file (i.e., letter) requires an average of 2 kilobytes. Now you can quickly calculate that you can store 360 letters (720/2) based on the amount of space available on the disk, but you will quickly find that is not true. You can only store 112 letters because that is the maximum number of file names which can be stored in the root directory. The point is, you will reach the maximum number of files that can be stored in the root directory long before you run out of disk space. If you have 112 2-kilobyte letters, you will use 224 kilobytes of disk space, and the remaining

space on the 720 kilobyte disk (about 500 kilobytes) will be totally useless. Or will it?

Tree Directory Structures

Fortunately, you can use a DOS feature that creates and uses a hierarchical directory structure that allows the use of multi-level subdirectories and overcomes the limited number of files allowed in the root directory. This is usually called a TREE DIRECTORY because it resembles an inverted tree with the root at the top, as shown in Figure 5.

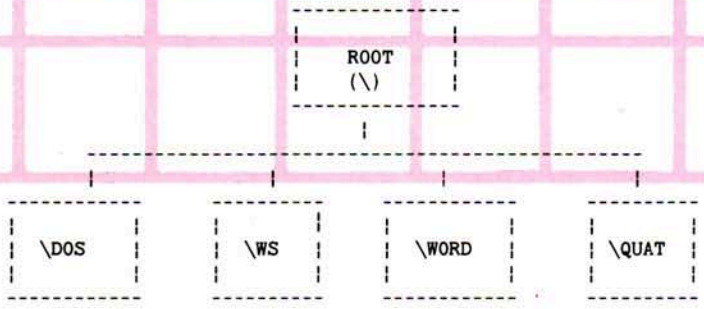


Figure 5
A Tree Directory Structure

By definition, the root directory on the highest level is always indicated by the reserved backslash (\) character used by itself. In this example, the \DOS subdirectory contains all of the program files (i.e., COM and EXE file types) copied from the distribution DOS disks, the \WS directory contains all of the WordStar program files, the \WORD subdirectory contains all of the Microsoft Word program files, and the \QUAT subdirectory contains all of the Quattro spreadsheet program files. Like file names, it is important to keep subdirectory names easy to remember, and shorter, but descriptive, subdirectory names are best.

There are some important terms that you should know in dealing with subdirectories. The root directory is frequently referred to as *the* directory. And a SUBDIRECTORY is any directory that is not the root directory. You will also find that most documentation and many discussions omit the "sub" prefix so that the word *directory* can apply to any subdirectory, as well as the root directory.

Regardless of where you are in the directory structure, you always have a CURRENT or WORKING directory. The terminology is the same used for a disk drive letter so that you are always "logged on to" a current drive and directory. If you have a floppy disk system, you always begin work in the root directory (\) on the boot drive A, even if you did not realize it. And when DOS searches for external commands, it always begins the search in the current directory on the current drive.

Subdirectory and Path Names

Naming a subdirectory is easy because it follows the standard DOS file name convention of *filename.typ*, but I recommend that you restrict your subdirectory names to eight characters or less. In other words, keep subdirectory names short, simple, and meaningful to you. I have found that some program documentation, including some DOS manuals, illustrates programs stored in the \BIN (binary) subdirectory. Unless you have worked with the Unix or Xenix operating system, that convention is rather silly. Most people prefer a subdirectory name like \DOS, \SYSTEM or \SYS, for short. Pick whatever name is most useful for you.

When you look at a directory structure, it is important to know that a DIRECTORY PATH is just a sequential list of subdirectory names that DOS uses to find a file. In Figure 5, \WS is a path, and WS is the directory (or subdirectory) name. Let's carry this one step further and add another level below the WS subdirectory as shown in Figure 6.

In Figure 6, various letters and documents created by WordStar are organized in the PERS (personal), BUS (business), and MISC (miscellaneous) subdirectories. To find a letter in the PERS subdirectory, DOS must use the path of \WS\PERS. Similarly, the BUS subdirectory has a path of \WS\BUS and the MISC subdirectory has a path of \WS\MISC.

Many DOS commands and application programs will accept path names. But there is one limitation imposed by DOS, and that is: a single path name to a subdirectory cannot exceed 63 characters including the backslashes — another good reason for keeping subdirectory names short.

Using Path Names

Subdirectory names are used in a DOS command line just like drive letters. For example, if your current drive was B and the CHKDSK program was located in the \DOS subdirectory on drive A, you could execute the program by entering:

B>*A:\DOS\CHKDSK*

Or you could get a listing of all the COM file types in the \DOS subdirectory on drive A by entering:

B>*DIR A:\DOS*.COM*

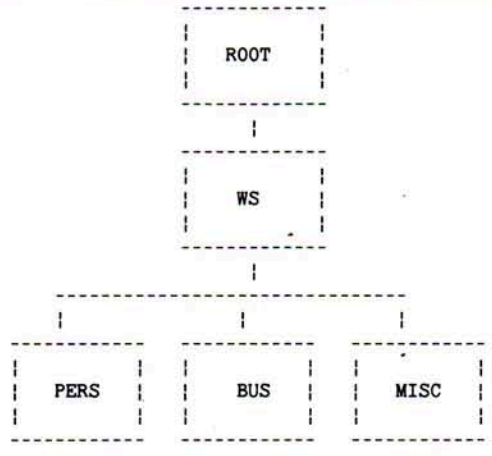


Figure 6
A Multi-Level Directory Structure

Notice that the backslash is required to separate the subdirectory name from the file specification, "*.COM", in this case.

Creating a Subdirectory

You can create a subdirectory on a disk by using the MKDIR or MD command. Most people prefer the use of the shorter MD (Make Directory) command, but you can also use MKDIR in exactly the same way. Let's create the WS directory and all its subdirectories shown in Figure 6 using the following commands:

A>*MD ×WS*
A>*MD ×WS×PERS*
A>*MD ×WS×BUS*
A>*MD ×WS×MISC*

By the way, you can duplicate this on your system, and it won't hurt anything if you are careful to enter exactly the commands shown. Now that these directories are created, it is important to know which is your current directory, as well as the current disk drive.

Where Am I?

Subdirectory names are useful, but they are not normally displayed as "a", like the current drive letter is. That makes it more than a little difficult to keep track of your current directory, but there is an easy solution — use the PROMPT command.

My personal preference is to use the following PROMPT command to display the current drive and directory:

A>*PROMPT $P QQ$G*

This command is useful because it changes the DOS command prompt to look like:

A:× ==>

The backslash indicates that the current directory is the root directory, and the drive letter A indicates the current drive, as usual. Now let's move around in the subdirectory structure.

Changing the Current Subdirectory

Changing the current directory is done with the Change Directory command: CD or CHDIR. Since you should still be in the root directory at this point, you can change your current working directory to WS by entering the following command:

A:× ==>*CD ×WS*

You will be able to see that your current working directory is WS, as shown by the command prompt of:

A:×WS ==>

Or you could have gone directly to the PERS subdirectory from the root directory using the following command:

A:× ==>*CD ×WS×PERS*

Then your command prompt would display as:

A:×WS×PERS ==>

By using a little "shorthand", you can move back up the tree to the WS subdirectory by entering:

A:×WS×PERS ==>*CD ..*

The dot-dot (..) notation simply means to go to the PARENT (the next level up) of the current (or CHILD indicated by a single dot) directory. When you use the DIR command in any subdirectory, you will find the current child directory indicated as a single dot, and its parent directory indicated as a double dot. If you enter the CD command shown above, you will change to the WS directory, and the command prompt will be displayed as:

A:×WS ==>

By entering the "CD .." command again, you can get back to the root directory, and the command prompt will be displayed as:

A:× ==>

If you remember that the root directory is always indicated by a backslash, a shorter way to get back to the root is to enter:

A:×WS×PERS ==>*CD ×*

When you use the "CD ×" command, you will always return to the root directory no matter where you are in the directory structure.

Deleting a Subdirectory

You can delete a subdirectory by using the Remove Directory command: RD or RMDIR. There are two special restrictions in using the RD command that you must know. First, you cannot delete a subdirectory name unless it is completely empty. That is, it cannot contain any files or other subdirectory names. For example, you could not delete the WS subdirectory until all of its subdirectories — PERS, BUS, and MISC — were deleted first. Since we have not added any files to these subdirectories, they could be deleted by using the following commands:

A:× ==>*RD ×WS×PERS*
A:× ==>*RD ×WS×BUS*
A:× ==>*RD ×WS×MISC*
A:× ==>*RD ×WS*

The first three command lines shown can be entered in any order, but the "RD ×WS" command *must* be entered last or you will see an error message displayed.

The second restriction is that you cannot delete the current directory, and I suggest that you always go to the root directory before using the RD command. This is not a requirement, but it is a standard way of doing business with your computer that can help make your life easier.

Sailing the Seven Subdirectories

In this example, you have learned to create, navigate, and delete seven subdirectories. The first four subdirectories — DOS, WS, WORD, and QUAT — were child directories of the parent root directory. The last three subdirectories — PERS, BUS, and MISC — were child directories of the parent WS directory.

Subdirectories are created with the Make Directory (MD) command and deleted with the Remove Directory (RD) command. You can change your current or working directory with the Change Directory (CD) command.

Path names are a list of the appropriate directory names in the hierarchical order created in the directory structure. In general, you can precede any valid command by a drive and/or path specification depending on the location of the file. The example shown in this article was: "A:×DOS×CHKDSK" where the CHKDSK program was located in the DOS subdirectory on drive A.

Since the use of wildcards was mentioned, it is also important to note that you *cannot* use the asterisk (*) and question mark (?) wildcards in a path name. Wildcards are only valid for file names in a subdirectory, and they cannot be used to mask the subdirectory name.

Using Wildcards and Paths

You can combine the use of a path name and an ambiguous file name specification in many DOS commands, and since we have already used the IR and DEL commands, let's continue with that example.

Since the DEL command allows us to erase files, it is important to check which files will be deleted before the command is executed. For example, let's check all of the KEY*.* files from the ×DOS subdirectory on our working system disk by using the following command:

A:× ==>*DIR ×DOS×KEY*.**

Since these programs are used to set up special keyboard configurations for non-US keyboards, you will probably not ever need these files, and they can be deleted with the following command:

A:× ==>*DEL ×DOS×KEY*.**

If you wanted to be exact, you could also precede each command with a drive letter, although that is not necessary in this case, because the ×DOS subdirectory is located on the default drive A. However, there is nothing wrong with adding the drive specification to these commands so that they would be:

A:× ==>*DIR A:×DOS×KEY*.**
A:× ==>*DEL A:×DOS×KEY*.**

In all cases, notice that the drive letter precedes the path which precedes the file name. Also notice that the path and file name are separated by the backslash which is required between them.

Next Time

At this point, you have seen how to use wildcards in file names and how to set up subdirectories in your system. We will use some of these features next time when we take a more detailed look at the DOS command line. You will see how to generally construct a command line using the information presented in this article.

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

Classified Ads

REACH THOUSANDS OF DEDICATED HEATH/ZENITH COMPUTER OWNERS!!

Classified Ad Rates

There is a minimum of $5.00 for 10 words or less. Each additional word is $.40. Payment for all classified advertising must be in advance.

Display Classified is available at $35.00 per column-inch. Minimum one inch. Display Classified is commissionable to recognized agencies.

How To Count Words

Count one word each for initials, standard abbreviations, whole numbers, name, address, city, state, zip, area code and telephone number. Dimensions (such as 6 × 9) are counted as one word. Box or department numbers are counted as one word each. All classified ads (not display) are set in the same size type. The first several words, depending on the ad, are set in all caps.

Continuity Discounts

Run your ad at least 3 times during the year; and receive a discount off the regular price.

Run 3 insertions — 3% off, 4 insertions — 4% off, 5 insertions — 5% off, . . . 12 insertions — 12% off.

Issue & Closing Date

Issue Date	Closing Date	Issue Date	Closing Date
January	November 15	July	May 15
February	December 15	August	June 15
March	January 15	September	July 15
April	February 15	October	August 15
May	March 15	November	September 15
June	April 15	December	October 15

REMark Magazine has the right to refuse any ad for any reason.

Classified Order Blank

First	10	words	at	a
cost	of	$5.00	each	insertion.
11-$5.40	12-$5.80	13-$6.20	14-$6.60	15-$7.00
16-$7.40	17-$7.80	18-$8.20	19-$8.60	20-$9.00
21-$9.40	22-$9.80	23-$10.20	24-$10.60	25-$11.00
26-$11.40	27-$11.80	28-$12.20	29-$12.60	30-$13.00
31-$13.40	32-$13.80	33-$14.20	34-$14.60	35-$15.00
36-$15.40	37-$15.80	38-$16.20	39-$16.60	40-$17.00
41-$17.40	42-$17.80	43-$18.20	44-$18.60	45-$19.00
46-$19.40	47-$19.80	48-$20.20	49-$20.60	50-$21.00

Enclosed is a check or money order of $_____ for _____ words.
(Minimum order: 10 words for $5.00. Each additional word $.40.)
Please insert this advertisement in the_____issue.

Signature_____

(Please type or print)

Name_____Phone_____
Company_____
Address_____
City_____State_____Zip_____

MAILING ADDRESS: Make check or money order payable to: Rupley's Advertising Service, Dept. — REM, 240 Ward Avenue, P.O. Box 348, St. Joseph, MI 49085, (616) 983-4550.

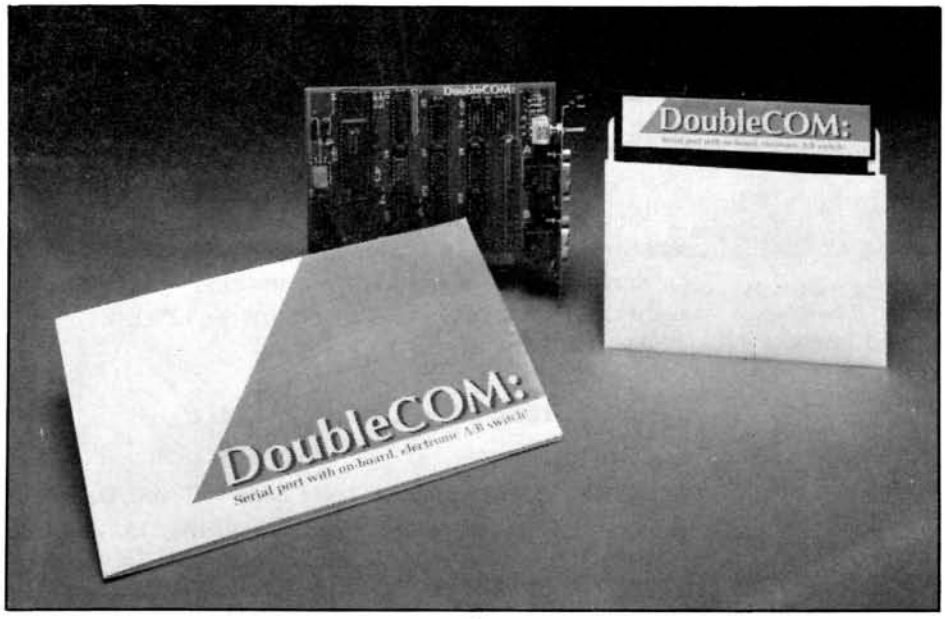
REMARK COMPLETE from No. 1 thru Vol. 8. (Nos. 1-13 reprint) Fantastic information in 95 issues. $60.00 plus shipping. Jerome Janick, 108 Duncan Dr., Troy, MI 48098.

C SCIENTIFIC LIBRARIES Personal CSL $85, Personal Statistics $99, and Personal Matrix $99. Personal libraries are subsets from CSL (440 functions) available for $275, source codes optional $75. Each routine is designed and documented for use by technical specialist. Call (408) 867-6575 or write Eigenware, 13090 La Vista Dr., Saratoga, CA 95070.

FOR SALE: Seagate 4051 40 meg. hard drive $275, Mitsubishi 4851 floppies for parts 3/$12, Z-151 power supplies $25 ea., 80287-6 Co-processor $130, Z-171 & Z-181 modems from $20, Z-100 Video boards $20 ea. Call (616) 926-6391, Surplus Trading Co., Benton Harbor, MI 49022.

THIS COULD BE YOUR AD! For as low as $5.00, you could advertise personal items in this space. And for as low as $35.00, you could advertise commercial items. People read these ads, you did!

Continued from Page 41



DG DoubleCOM:

- PC compatible serial I/O port
- Built-in, software controlled electronic A/B switch
- Simple keyboard-command operation

Double-COM: is a serial I/O port and software controlled electronic A/B switch on a single, easy to install, PC compatible card. DoubleCOM: allows two serial peripherals to be operated from one of the standard PC serial ports, COM1 or COM2, while maintaining full compatibility with all commonly available application software. The card is compatible with the IBM PC/XT/AT, PS/2 models 25 and 30 and fully compatible computers.

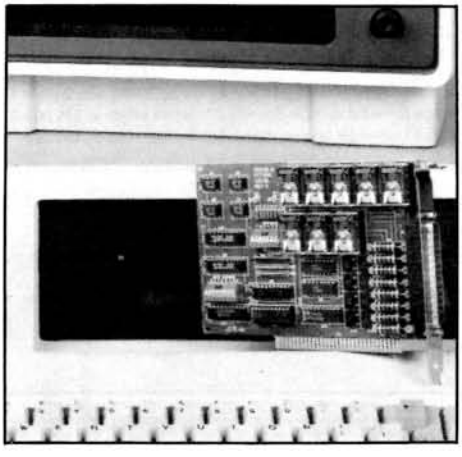
Software included with DoubleCOM: provides convenient peripheral selection in systems using either DOS or Microsoft Windows. This selection may be made even from within application programs. The DOS selection software uses a simple keystroke sequence to select the active device, while the Windows utility uses a familiar icon switch to facilitate device switching. For added simplicity, DoubleCOM: software for both DOS and Windows automatically performs the communication parameter setting usually performed manually with the DOS Mode command.

DoubleCOM: provides greater convenience and reliability than old style switching methods, such as cable-swapping and mechanical "black-box" switching devices. The cable and workspace "clutter" common with these switching methods is eliminated and DoubleCOM: electronic switching technology removes the "glitches" and other headaches associated with mechanical switches. DoubleCOM: also complies with FCC Class B specifications.

DoubleCOM: greatly improves the utility of peripheral intensive systems, such as Computer-aided Design/Computer Aided Engineering (CAD/CAE) or Desktop Publishing work stations. DoubleCOM: may be used with all common serial peripherals such as printers, plotters, modems, pointing devices such as a mouse or trackball, bar-code readers, digitizing tablets, computer-controlled instrumentation, and many RS-232 network systems.

Warranty: 1Year. Price: $149.00. For more information, contact:

D-G Electronic Developments Company
700 S. Armstrong
Denison, TX 75020
(214) 465-7805



MetraBytes' PDISO-8 is a low cost, industrial quality relay, input/output interface board designed for control and sensing applications that require IBM PC/XT/AT compatibility, intput/output isolation, and multi-channel capability.

The PDISO-8 provides 8 electromechanical relay outputs and 8 optically isolated inputs. Five of the output relays are SPDT Form C changeover types, while the remaining three are normally open SPST Form A types. Relay contacts are silver with gold overlay and are guaranteed for 10,000,000 operations. Each relay contact is rated at 3A @ 120 VAC or 28 VDC. All 8 relay outputs are addressed as a single I/O byte.

The 8 optically isolated input channels may also be read as a single I/O port. Inputs are non-polarized and have a range of 5 to 24 volts DC or AC (DC and AC inputs may be mixed). The input range may be extended by adding input resistance externally. Standard 12/24V AC control transformers may be used to drive the inputs.

Response time for DC inputs is typically less than 20 uSec. Channel to Channel, input to ground and output to ground isolation is 500 V. Programming is accomplished by utilizing direct I/O instructions in whatever application language is being used (BASIC, C, Assembly, Pascal, etc.).

Its blend of high performance features, full IBM PC/XT/AT compatibility, ease of application, and low cost, make the PDISO-8 the choice for applications such as programmed load switching, external switch input sensing and external voltage sensing.

For more information, contact MetraByte Corp., 440 Myles Standish Boulevard, Taunton, MA 02780. (508) 880-3000. Price: PDISO-8 — $225.00. *

Announcement!

HUG MEMBERS ONLY!!

The HUG-386 and HUG-386-C upgrade kits will be available shortly. Wheelin' Dealin' Jim has managed a super-fantastic deal on these two products for Heath Users' Group members who originally purchased an H-241 or H-248; ***one-thousand two-hundred dollars*** off the regular purchase price! That's right! If you originally purchased an H-241 or H-248, and you're a HUG member, you can get $1200.00 off the regular retail price of either of these two upgrade kits!

The HUG-386 and HUG-386-C are upgrade kits that let you upgrade your H-241 or H-248 series computers up to a full H-386. Now, how do you determine which upgrade kit to buy? The H-386-C includes a dual winchester/floppy controller, while the H-386 does not include any disk controller. Since the old H-241 controller is not '386 compatible, you'll probably want the "C" model if you're upgrading a '241. If you're upgrading a '248, your decision will depend on whether you need a new dual controller or not.

Here are the three ways you can order your upgrade:

Write-In Orders

- Non-HUG members ***can*** order by including payment (with the upgrade kit order) for one year's membership in the Heath Users' Group.
- All orders should be submitted to the Heath Users' Group.
- Each order must indicate the model number of the upgrade kit desired, and which computer kit it was purchased for.
- Each order must have the persons HUG ID number written on it.

Phone-In Orders

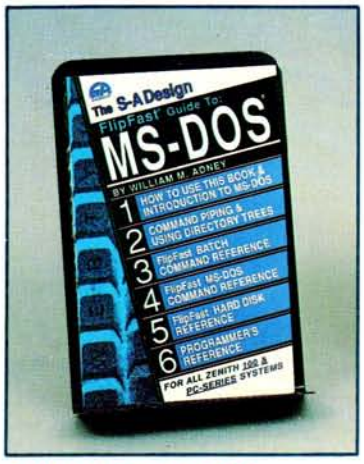
- Non-HUG members ***can*** order by first ordering a one year's membership in the Heath Users' Group.
- All orders must be phoned in to (616) 982-3838 from 8 AM to 4:30 PM EST.
- Each order must indicate the model number of the upgrade kit desired, and which computer kit it was purchased for.
- The person ordering must supply his/her current HUG ID number.

Heath/Zenith Computer Store Sales

- Non-HUG members ***can*** purchase an upgrade kit by first purchasing a HUG membership from the store.
- Orders for the upgrade kit can be taken in the normal fashion.
- Each order must have the buyer's HUG ID number on it.
- Each order should indicate which computer kit the upgrade was purchased for.

FlipFast Guides To . . .

Designed to be quick reference manuals, these books are for beginners and 'seasoned' users alike. Here are some of the features you'll find between the covers.



MS-DOS

- **Names** of all commands or sub-commands
- **Syntax** with available options
- **Applications** concisely explaining typical command usage
- **Examples** with "real-world" explanations
- **Requirements** for general usage that minimize system errors and speed computer operations
- **Error messages** are explained for each command, where applicable

Plus:

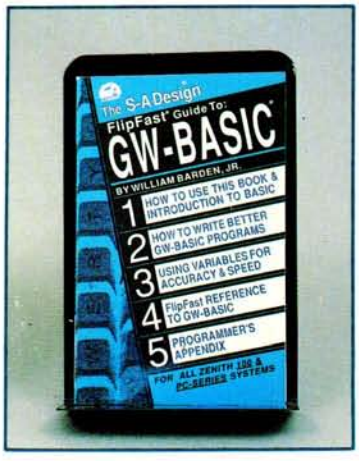
- How to use MS-DOS
- Complete batch file & hard-disk reference
- Programmer's command reference
- DOS Technical reference
- Config file reference, glossary and more!

GW-BASIC

- **Names** of all commands, statements, and functions
- **Syntax** with available options
- **Examples** with "real-world" explanations
- **Requirements** for general usage that minimize programming errors and simplify debugging
- **Application notes** that give specific information about command functions.
- **Hints, tips, and bugs** that can save you hours of programming time
- **Error messages** listed and explained for each command, where applicable

Plus:

- How to use GW-BASIC
- How to write better and faster programs
- Tips on using variables
- Complete reference appendix



FlipFast Guide to GW-BASIC, HUG P/N 885-4601 . . . . $20.00
FlipFast Guide to MS-DOS, HUG P/N 885-4602 . . . . . $23.00
FlipFast Guide Combo, HUG P/N 885-4603 . . . . . $41.00

Heath/Zenith Users' Group • P.O. Box 217 • Benton Harbor, MI 49022 • (616) 982-3463

Heath/Zenith Users' Group

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

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

POSTMASTER: If undeliverable, please do not return.

P/N 885-2103