

## CHANGING RESIDENT MS-DOS CHARACTER SETS AND KEYBOARDS

/ J. Hasteley

Do you like the character set and keyboard table your operating system was born with?

MODCON, as most know, is a helpful utility that allows the character set and keyboard table to be changed at run-time but the changes are lost when power is switched off. MODCON also takes up about 24K, a character set 8K, and a keyboard table up to 1.5K of disk space.

SYSELECT and SYSGEN allow custom building of operating systems from scratch, incorporating the character set and keyboard table of your choice. However, these programs are only available as part of a fairly expensive Programmer's Toolkit I or Systems Programmer's Toolkit II.

The following describes how DEBUG, available on PSU Utility Disk #1 and the Victor/Sirius File DEALS FOR READERS White Crane Disk (VM013), can be used to permanently alter the character set and keyboard table in your operating system as if it were configured with SYSELECT or SYSGEN.

Character set (.CHR) and keyboard table (.KB) files are already pretty much in the same format as those which are part of the operating system file MSDOS.SYS. These files consist of a header, data, and, in the case of keyboard tables, a trailing block of zeroes of varying length. Once the headers and trailing zeroes are removed, the remaining data can be overlaid into the appropriate areas within MSDOS.SYS and the altered file saved back to disk.

Before any alterations can be made, however, MSDOS.SYS must first be unhidden. The procedure to do this has been previously covered in both 16 BIT COMPUTING-UK and PSU but is included below for completeness.

Two procedures for doing the above are provided below: one for DOS 1.25 and one for DOS 2.11. The reasons for separate procedures are:

1. File size. Standard configuration MSDOS.SYS 1.25 is 39K bytes long allowing it and the character set/keyboard table files to be loaded separately into a 64K segment for a simple juggling session. Standard configuration MSDOS.SYS 2.11 is 59K bytes long. I chose to directly load in the character set on top of MSDOS.SYS

without changing DEBUG's segment addresses. This is possible with version 2.11 as there is a rather large blank space immediately in front of the existing character set which can accommodate the character set header without overwriting anything of importance. The header is then erased.

2. The locations of character sets and keyboard tables within MSDOS.SYS are different between versions.

3. The location of MSDOS.SYS in the disk directory is different between versions. The DOS 1.25 version is usually the first entry. The DOS 2.11 version will most likely be elsewhere due to the Volume ID and subdirectories.

4. The two versions of DEBUG (for DOS 1.xx and DOS 2.xx) use different labels for the disk drives.

As usual, before tempting the god of botched (BATched ?) operating systems, prepare a sacrificial disk containing the operating system, DEBUG, plus the character set and/or keyboard table of your choice. USE THIS DISK! No stronger warning can be given.

The procedures below contain step numbers and comments for easy referencing. While in DEBUG enter only the items under the "Action" column, NOT the step numbers or the comments. Also, the sacrificial disk MUST be in drive A.

These procedures DO work for the BIOS releases noted. Other releases may have different starting addresses for the character set and keyboard table. If this is the case, a little snooping around with DEBUG will be necessary to find the correct locations. To do this, first examine a typical character set and keyboard table with DEBUG and then scan thru MSDOS.SYS to locate same.

The above method most likely could be used for changing other parts of MS-DOS and would provide a cost effective alternative to the SYSELECT and SYSGEN programs in the toolkits.

Others more knowledgeable are implored to comment on these procedures and let me know if there are serious faults. I am not a programmer by profession, just someone with terminal curiosity about this wonderful machine.

\*\*\*\*\* Unhide MSDOS.SYS 1.25/2.6 \*\*\*\*\*

Step	Action	Comment
1	A>debug <cr>	;invoke DEBUG
2	>l100 1 0 10 <cr>	;load directory sectors
3	>d700	;display the first few directory entries. ; MSDOS.SYS should be the first.
4	>e070b <cr>	;change byte eleven (at 070B) from
5	xxxx:070B 06.00 <cr>	; 06 to 00 to unhide MSDOS.SYS
6	>w100 1 0 10 <cr>	;write directory sectors back to disk
7	>q <cr>	;exit DEBUG
	A>	
8	Open and close the disk drive door to force MS-DOS to reload the directory in which MSDOS.SYS now appears.	

Note: To rehide MSDOS.SYS 1.25/2.6 repeat the above procedure changing the byte at xxx:070B back to 06.

\*\*\*\*\* Change Character Set/Keyboard Table for MS-DOS 1.25/2.6 \*\*\*\*\*

Step	Action	Comment
1	A>debug <cr>	;invoke DEBUG
2	>nfilespec.kb <cr>	;name your keyboard table file
3	>lF000 <cr>	;load keyboard table at offset F000
4	>nfilespec.chr <cr>	;name your character set file
5	>l9ad4 <cr>	;load character set at offset 9AD4
6	>nmsdos.sys <cr>	;name the operating system file
7	>l <cr>	;load it at DEBUG's default offset 0100
8	>mf080 f47f 9320 <cr>	;overlay the existing keyboard table
9	>mf00e f019 9945 <cr>	;overlay keyboard table name/class in banner
10	>m9b70 bb5f 0b70 <cr>	;overlay the existing character set
11	>m9ae2 9aed 996d <cr>	;overlay character set name/class in banner
12	>w <cr>	;write the altered system file to disk
13	>q <cr>	;exit DEBUG
	A>	
14	Push the reset button and observe that the new character set and/or keyboard table appear in the boot-up banner as well as onscreen when you start punching the keys	

Notes: To change character set only, omit steps 2,3,8, and 9.  
To change keyboard table only, omit steps 4,5,10, and 11.

\*\*\*\*\* Unhide MSDOS.SYS 2.11/2.9 \*\*\*\*\*

Step	Action	Comment
1	A>debug <cr>	;invoke DEBUG
2	-l100 0 0 10 <cr>	;load directory sectors
3	-d700	;scan thru the directory with the "d" command ; starting at the beginning (offset 0700) ; until you find the entry for MSDOS.SYS
4	-exxxb <cr>	;change byte eleven (hex value B) in
5	xxxx:xxxB 06.20 <cr>	; the directory entry from 06 to 20 to ; unhide MSDOS.SYS
6	-w100 0 0 10 <cr>	;write directory sectors back to disk
7	-q <cr>	;exit DEBUG
	A>	
8		Open and close the disk drive door to force MS-DOS to reload the directory in which MSDOS.SYS now appears.

Note: To rehide MSDOS.SYS 2.11/2.9, repeat the above procedure  
changing the byte at xxxx:xxxB back to 06.

\*\*\*\*\* Change Character Set/Keyboard Table for MS-DOS 2.11/2.9 \*\*\*\*\*

Step	Action	Comment
1	A>debug <cr>	;invoke DEBUG
2	-nfilespec.kb <cr>	;name your keyboard table file
3	-lf000 <cr>	;load keyboard table at offset F000
4	-nmsdos.sys <cr>	;name the operating system file
5	-l <cr>	;load it at DEBUG's default offset 0100
6	-r <cr>	;write down the value in CX (should be E8C0)
7	-nfilespec.chr <cr>	;name your character set file
8	-l1b84 <cr>	;directly load it at offset 1B84
9	-m1b92 1b9d e72d <cr>	;overlay character set name/class in banner
10	-f1b84 1c1f 0 <cr>	;erase character set header
11	-mf080 f47f e0e0 <cr>	;overlay existing keyboard table
12	-mf00e f019 e705 <cr>	;overlay keyboard table name/class in banner
13	-nmsdos.sys <cr>	;name the system file to be written to disk
14	-rcx <cr>	;change the value in CX
	CX xxxx	; back to the original length
15	:e8c0 <cr>	; of MSDOS.SYS, i.e., E8C0
16	-w <cr>	;write the system file back to disk
17	-q <cr>	;exit DEBUG
	A>	
18		Push the reset button and observe that the new character set and/or keyboard table appear in the boot-up banner as well as onscreen when you start punching the keys.

Notes: To change character set only, omit steps 2,3,11, and 12.  
To change keyboard table only, omit steps 6 thru 10, and 13 thru 15.

\* \* \*

MS-DOS 2.11 COMMAND FORMATS BY FUNCTION CATEGORY / Jack Prichard

[This table supplements the three-part series on MS-DOS 2.11 by Mary Jo Knobelsdorf in VictorTalk, Vol 4, No 2, 3 and 6.]

Command Format	Ref.	E1/E2	Function
<b>DEVICE/CONFIGURATION COMMANDS:</b>			
CAUX[device name]	29		Change auxiliary device
CLS	21		Clear screen
CLST[device name]	29		Change listing device (printer)
CONFIG	1-6	E1	Configure system
CTTY[device name]	29		Change console devices (keyboard and screen)
ECHO ON ECHO OFF ECHO message	37		Echo (display) messages to screen
MODCON sourcefile [savefile]	64	E1	Modify console keyboard and character set
MORE	67	E1	Display one screen at a time
PORTSET[letter][baud rate] [parity][stop bits] [bits/character][/S]	73	E1	Set serial port parameters
132C 132ON 132OFF	D-11	E2	Install 132-column utility; turn on; turn off
<b>DISK COMMANDS:</b>			
CHKDSK[drivename:][filename.ext] [>filename.ext][/F][/V]	18	E1	Check disk for consistency
DISKCOPY[side to side][/C][/E][/Z]	33	E1	Copy disk (double disk machines)
FORMAT[side:][/C][/D][/E][/Fx]	51	E1	Format disk
SDCOPY[drivename:]	88	E7	Copy disk (single disk machines) (supplied only with hard-disk machines)
VOL[drivename:][/C]	118		Display/change disk volume label
<b>FILE COMMANDS:</b>			
CONCAT[files or directories]	23	E2	Concatenate files
COPY[path]filename.ext[/A][/B] +[path]filename.ext[/A][/B]...] [path][filename.ext][/A][/B][/V]	24		Copy file(s)
DEL[path]filename.ext	31		Delete file(s)
EDLIN filename.ext	4-1	E1	Edit file
FGREP[options]expression[files or directories]	44	E1	Search for files with specified string
FILCOM[#][/B][/C][/W]file1 file2	39	E1	Compare files
FIND[/C][/N][/V]"string"[path] filename.ext...	46	E1	Find string in file
MORE	67	E1	Display file, one screen at a time
MV[path]filename.ext[path] filename.ext	68	E1	Move file(s)
PRINT[drivename:]filename.ext [/C][/P][/T]...	75	E1	Print queued file(s) in background
[drivename:]RDCPM DIR[drivename:] [filename.ext]	82	E2	Read CP/M file to MS-DOS file
[drivename:]RDCPM[drivename] filename.ext	"	"	" " " " " " " "
RECOVER[path]filename.ext	84	E1	Recover file(s) from defective disk
RECOVER drivename:	"	"	" " " " " " " "
REN[path]filename.ext filename.ext	87		Rename file(s)
[command]SORT[/R][+n] [<[drivename:]input source] [>[drivename:]output source]	108	E1	Sort file or data

Command Format	Ref.	E1/E2	Function
SEARCH files [matching constraints] [actions]	90	E1	Search files and perform action
SYS source[destination]	112	E1	Copy system file(s)
TAIL[/n][files or directories]	111	E2	Display last lin(s) of file or directory
TYPE[path]filename.ext	116		Display/print (with ALT-P) file
VERIFY [ON] [OFF]	117		Display/set verify file function
WC[/R][/W][path or filename]	119	E2	Count word in file(s)
<b>DIRECTORY COMMANDS:</b>			
CH (CHDIR) [. .][/A][path]	16		Change directory
DIR[path][filename][.ext][/P][/W]	32		List directory
LS[options][files or directories]	59	E1	" " alphabetically
MD (MKDIR)[path]	61		Make new subdirectory
PATH[path];[path]	70		Display or search path thru directory
RD (RMDIR)[path]	86		Remove empty subdirectory
<b>BATCH FILE COMMANDS:</b>			
FOR %%variable IN (set) DO command %%variable	49		Repeat DOS command for each member in set
GOTO label	53		Branch unconditionally
IF [NOT]string1==string2 command	57		" conditionally
IF [NOT] EXIST[path]filename.ext	"		" "
PAUSE[comment]	71		Stop until key pressed
REM (.) comment	85		Display line as remark (comment)
:			Do not display remark when file run
<b>MISCELLANEOUS COMMANDS:</b>			
BREAK [ON] [OFF]	15		Display/set ALT-C interrupt status
CALC	H-1		Load calculator program
COMMAND[path][device][/D][/P] [/C string]	21	E1	Reload COMMAND.COM
CPM	5-1	E2	Load CP/M emulator
DATE[mm/dd/yy] (American)	30		Display/set date
DATE[dd-mm-yy] (European)	"		" "
EXIT	39		Exit from COMMAND.COM to previous program
HI (HISTORY)[n]	55		Display command history or re-enter command
PROMPT[prompt-text]	80		Change MS-DOS prompt
RDCPM[DIR][drivename:][filename.ext]	82	E2	Transfer CP/M files or display CP/M directory
SET[string1=string2]	105		Display/set equivalent value
SORT (see ref.)	108	E1	Sort data
TIME[hh:mm:ss]	114		Display/set time
VER	116		Display MS-DOS version

#### NOTES

CAPS command names; alternate (long) form in parentheses; enter in upper- or lower-case.

[ ] enclose optional parameters.

... ellipsis indicates preceding parameter may be repeated.

/ indicates parameter (such as /C) is a switch.

| indicates data is to be piped; output to left sent as input to right of bar.

< indicates input to come from source other than keyboard.

> indicates output to be sent to file or device (such as printer).

>> indicates output to be sent to end of file named.

All commands are entered with RETURN key.

References, unless otherwise noted, are to pages of Section 7, MS-DOS 2.1 User's Guide, first VICTOR printing Feb, 1984.

E1/E2 indicates external command on disk 1/2 of 2; all other commands are internal.

\* \* \*

# FROM THE BOARDS / Mary Jo Knobelsdorf

Our thanks to the Fuzzy Bear for this short visit to the Electronic Forest. Some time, won't you make your own visit and leave a little honey for the Fuzzy Bear?

[2] #23 (of 23) CASS TYSON, on 07-DEC-86 15:25 (Size: 6016 Bytes) Subject: US SUB-SUB HACKERS FIFTH CLASS WANNA KNOW

I haven't contributed a whole lot to the conversation here on the boards because:

1) I'm really a novice in this whole business, and therefore have not been able to answer anyone's queries in the first place.

2) I hesitate to ask any questions that I think can be looked up somewhere — I feel like I'm burdening others with my stupid questions. (But maybe I'm wrong to feel that way — it gave me a good feeling to be able to explain something about MS-DOS to our new BRAVO! Public Domain librarian.)

3) And probably my biggest hurdle: I \*hate\* to appear ignorant in front of a crowd of informed people — and these messages get read by quite a crowd! (I know, Fuzzy... the dumbest question is the unasked one!)

So, to make a long story short, I've been following (like a first year French student follows the conversation at the next table in a Paris restaurant) your high level technical and software related discussions and saving them on disk somewhere so that someday, maybe, (hopefully) I'll be able to find the answers to my future problems. (It's getting to the point, though, that I've got so much talk on so many different subjects that now I don't know where to find anything specific! Thank you for the Newdisk files index on Newdisk 19 — I \*know\* that will be a Godsend for me someday.)

Meanwhile, I've been dutifully trying to look up the solutions to my trivial problems in the manuals that came with the machine. Unfortunately, as you all know, most manuals are written for people who already know about computers, and not for those who are trying to learn. At the very least, one must know what the problem is before he can figure out where to look it up, right? So I'm going to throw caution to the wind and post some questions that I either can't find the answers to or don't know where to look. (What's really crazy is that many of the people in BRAVO!, who have only recently become Victor owners, are as far behind me as I am behind Dale and Clint and Marty, et al. The other day someone asked me what it means to be 'in' the B drive and what does it mean to 'switch' drives.

Where do you begin to explain something like that?!

Anyway, I have a bunch of questions that I'm sure are elementary for most of you but will be tremendously helpful to us sub-sub hackers 5th class in Baltimore. Here goes:

1) What are the uses (or, really, the advantages) of the DOS utilities (commands?) TAIL, FILCOM, MORE, FGREP, FIND. (Are these the sources of the term 'filtering' data? If not, just what does it mean to 'filter' data?) Most importantly, what are the \*best\* ways to use them?

2) How can I use SEARCH in a batch file that will allow me to backup a bunch of files that I've worked on in a word processing session? I have tried this with only limited success. What I want to be able to do is exit from WP and just type BAK (for instance) and let BAK.BAT search the data disk in drive B and copy (with replace) any files on drive A which are older as well as add any new files on B to A. If there is a simpler way than this, please tell me. The manual is as clear as mud in explaining SEARCH.

3) While working in MULTO, I occasionally have files become "circular" (for want of a better word) meaning that as long as I keep scrolling, the end of the file leads into the beginning: ie. no 'end of file' message. These files usually have some garbage in them somewhere and give a 'read fault error' message after I exit. Why is this happening? How can I prevent it?

4) What are the uses/advantages of 'piping' data? Again, how do you do it?

5) How do you install a different keyboard? I have Marty's DOS2.KB and the .DOC file that explains it, but how do you get it to work? How would I create a personalized variation? Do I need some kind of toolkit to do that?

6) A couple of people in the area recently took advantage of the big deal on HD machines. For them:

How and why do you 'format' a HD? (You do, don't you?)

How do you create a main menu and submenus for using the various programs?

What are the advantages of a HD over a DS machine? (I know there are many but I mean specifically.)

How do you do what Dale mentioned in his editorial on ND 18? That is, back up a whole day's work on one disk at the end of your computing session? Would this be similar to the answer to my question #2?

7) I know exactly nothing about CPM-86. Are there any advantages to becoming familiar with it? Or will MS-DOS do everything it will and then some? Are there useful utilities available only on CPM, for instance?

8) What are the differences, similarities, advantages and disadvantages between MS-DOS and UNIX type systems available for the V9? (There are some, right? XENIX, or something like that?) Are they expensive? Worthwhile? A waste of time? (See next question)

9) I foresee a future in dealing with BM machinery, most likely if I shoot for a full time job with my current company. I'll never stop loving my Mercedes here, but what should I learn (teach myself) to get along better in the Volkswagon (BM) world? Should I learn other operating systems such as CPM? XENIX? Should I familiarize myself with certain programs such as LOTUS or EMIBM? Can anyone who knows the differences/similarities between us and them suggest some possible lines of activity that I might/should consider?

Answers to these questions and any thoughts that might be triggered would be greatly appreciated by me and BRAVO! en toto.

Thanks,  
Cass Tyson

[2] #24 (of 24) BERT HAPPEL, on 08-DEC-86 22:05 (Size: 3155 Bytes) Subject: Some answers for Cass

Cass - Here's a long answer to a couple of the questions you posted.....

#### Question #2 - Backing up

SEARCH is a very powerful and somewhat complicated utility that has been rumored to do nasties if you don't watch it. What I use for the situation you describe (and therefore I think you need it too) is a utility program called BAC.COM. BAC is a public domain program that was first described in PC magazine in August 1985.

BAC is a utility program that will selectively backup files from one drive (or subdirectory) to another. BAC acts upon files that are on the source drive ("A:" in your example) that 1) Do not exist on the destination drive ("B:" in your

example) -OR- 2) Have a newer time or date for the filename as compared to the destination drive.

For your example of backing up new or revised files created by your WP on drive A:, BAC.COM should be on drive A: (or on B: with PATH set properly).

```
A>BAC *.* B:      < Would backup *all*
                  files on A: including
                  the WP program.
```

```
A>BAC *.TXT B:    < Would backup all
                  .TXT files (you will
                  substitute the file
                  extension your WP uses)
```

If the file is the same on both drives, no backup is done. If the destination drive becomes full, BAC prompts you for another disk. The caution at that point is that when backing up later, you must use the same disks in the same order. If you often delete the text files you create on drive "A", BAC will still have a copy of them on your backup disk. This can create the disk full message rather quickly. Of course this command could be put in a batch file called BAK.BAT.

BAC.COM and BAC.DOC are available in file area #4 here on The PUB.

#### Question #5 - Installing keyboard

The quick and dirty answer to this question is to beg, borrow or steal a copy of the PSU disk (probably available through VictorPulse or contact some user like myself who has a copy). This disk has not only the utility programs but also extensive documentaion. Some of the programs on the disk:

KEYGEN KEYboard GENerator, program that allows you to customize your own keyboard.

EFONT Edit FONTS, program that allows you to define charcter sets (have you seen Hebrew, Cyrillic, Olde English?).

MODCON MODify CONsole, program that installs a keyboard or character set file.

These programs are also available on various Toolkits. EFONT suffers from the fact that it only works with DOS 1.25. KEYGEN and MODCON work with any DOS on the Victor 9000 (though there are different versions (updates) of Modcon).

Hope you find this info helpful. From one Advanced Amateur Sub-Hacker to another.....

# FASTTRAX / Franz Hirner

One of the pleasures of running a RBBS (Remote Bulletin Board System) is the chance to participate in the development of software for the Victor 9000. It is much faster to communicate during the development process via modem than it is via regular mail, speeding up the development process considerably. The latest versions of either code or programs can be moved back and forth in a matter of moments, rather than days.

Victor has sold several thousand V9's equipped with hard disks and several thousand more have been retro-fitted with hard disks by Victor dealers or dedicated users. As far as I know there are only two disk optimizing programs available for the Victor 9000. I'm happy to say that BAVUG (Bay Area Users Group) worked with the developers of both packages.

A hard disk is one of the best investments, in terms of both convenience and speed, that you can make. All of your programs and data files are but a keystroke away, once you have a hard disk on your system. You'll wonder how you ever managed with only floppy drives, once you have worked with a hard disk for just a few hours. A hard disk does require periodic backups to protect your data, though. Also, the data files on your hard disk should be optimized from time to time to maintain the speed you've grown accustomed to. Disk optimizing programs come to the rescue.

When does your disk need to be optimized? A brief tour of MS-DOS will provide the answer. MS-DOS manages the files on your disk -- either floppy or hard disk. When you write a file to disk, DOS starts at the "top" of the disk and searches for the first free cluster and writes a portion of your file to that cluster. DOS then continues searching down the disk (File Allocation Table), looking for another free cluster. When it finds one, it writes the next part of your file to that cluster and then repeats the process again until the entire file has written to disk. This is true for both floppy and hard disk. Your file may be scattered in a dozen or more sectors anywhere from the beginning to the end of the physical disk. The longer you have been running the disk and the more files you have added and deleted, the more fragmented the files on the disk will be. Little by little, the performance of your disk is degraded and you wait longer and longer to load a program or file. At this point you need to optimize your disk. You can delete all the programs on the disk and reinstall them one by one. This will automatically place all sectors of a file consecutively on the disk. (However, this is a chore no one wants to undertake on a regular basis.) Alternatively, you can use a disk optimizer program such as FastTrax by Mark Elfield of ME&A.

FastTrax comes with a complete and easily understood manual. The program installation and operation are straight forward. After the first optimization, FastTrax can maintain optimization in just a few minutes if you run the program once a week. Your hard disk will perform like it did when it was new.

With Mr. Elfield's permission, following are excerpts from the user manual.

**FastTrax** optimizes hard disk and floppy disk performance by consolidating scattered pieces of files and subdirectories. FastTrax not only allocates files to contiguous disk clusters, it assigns most files to the least number of disk cylinders, reducing track-to-track seeks. In addition, FastTrax allows the user to assign frequently-used programs and data to the "fast tracks" of the disk (near the File Allocation Table), significantly improving access times to these files. To show the amount of fragmentation on a disk, FastTrax displays an informative visual report.

**FastTrax** operates on both floppy disks and large capacity hard disks, including hard disks with DOS partitions larger than 32MB (up to 4000MB).

**FastTrax** is compatible with existing copy-protection schemes. FastTrax itself is NOT copy-protected.

**FastTrax** requires an IBM PC/XT/AT or compatible (including the Victor 9000 and Victor VI series computers). You should have 256K memory in order to optimize hard disks up to 32 MB. You must have MS-DOS 2.0 or later.

**FastTrax** never moves a file or subdirectory that is marked as "hidden", "read-only", "system", or any combination of these attributes. This is to maintain compatibility with copy-protection schemes. But, since some software companies frequently change their protection schemes, you may wish to "uninstall" copy-protected software before running FastTrax. In this document, disk clusters that FastTrax will not move are called "held" clusters.

**FastTrax** either assigns files and subdirectories in "directory order" or in "command file" order. With directory order, FastTrax assigns all entries in a given directory physically near the directory itself in the order the entries appear in that directory. With command file order, FastTrax follows an optional user-supplied command file.

**FastTrax** offers two "packing" methods that move files and subdirectories to more efficient places on a disk:



### Consecutive packing method

This method assigns files and subdirectories to consecutive clusters, producing contiguous files. This is sometimes called "de-fragmenting" and is a good technique for increasing disk access speed. However, the consecutive method alone is not as effective as the optimized method (described next) because it frequently produces cluster "orphans" -- file segments that needlessly span a cylinder boundary. The consecutive method can only be specified via the command file.

### Optimized packing method

With this method, FastTrax performs "de-fragmentation" as described in the consecutive method, but also attempts to assign each file and subdirectory to the least number of disk cylinders. The optimized method minimizes cluster orphans and serves to further reduce track-to-track disk seeks. This is the normal (default) packing method used by FastTrax; it may also be specified in a command file.

### Comparison of packing methods

The consecutive method is included with FastTrax to offer some functional compatibility with other disk optimizers. However, you may never need it because the optimized method is better. One possible use: If you have an application program that runs faster when its files and subdirectories are in a specific order on your disk, you can use the consecutive method in conjunction with the command file to produce this order.

With the optimized method FastTrax will, if necessary, deviate slightly from the indicated file order to assign a file to the least number of disk cylinders. But, since some disk clusters actually span a cylinder, there is often no optimum assignment for these "black sheep" clusters because FastTrax limits itself when changing the file order. Finally, since FastTrax always assigns files to contiguous clusters, the last file will not be optimized if it crosses a cylinder boundary.

In either method, FastTrax may encounter bad clusters and held clusters when assigning a file (or subdirectory). Although FastTrax bypasses these clusters, it may assign a file around them, resulting in a fragmented file. Fragmentation caused by intervening bad clusters and held files happens much less often with the optimized method because FastTrax alters the file order to avoid fragmentation.

### File location

FastTrax allows the user who demands maximum performance to create a command file that tells the exact location of all files on the disk. The creation is easily done and fully

discussed in the manual. While substantial improvements in disk performance are achieved with the default parameters built into the program, a custom command file for your system will produce the best results.

### Running FastTrax

To run FastTrax, type:

```
FASTTRAX
```

FastTrax will prompt you for the "target" disk to analyze. To bypass this prompt, you may include the drive letter and a colon on the command line (e.g., FASTTRAX C:).

FastTrax then analyzes the target disk and displays a visual fragmentation report. Then FastTrax prompts you to either enter the optimization parameters (described below) or return to DOS.

If you elect to enter the optimization parameters, FastTrax looks for a command file named "FASTTRAX.CMD" in the root directory of the target disk. If found, FastTrax displays the first few lines of this file and asks you if you want to use it; if yes, FastTrax will start the optimization process. Otherwise, FastTrax gives you three optimization choices:

- A - Optimize **all** files and subdirectories on the disk.
- F - Optimize only the **fragmented** files on the disk.
- X - **Exit** to DOS without optimizing anything.

Use the 'A' option the first time you optimize a disk or when the disk is badly fragmented. This option places files at the low end of the disk in "directory" order, and places free space at the high end of the disk.

Use the 'F' option to "touch up" your disk. The 'F' option is much faster, but it may leave a few fragmented files if your disk is nearly full or if the fragmented files are very long. This option looks for a better place for your fragmented files -- in so doing, it may leave a few free space "holes".

There are two other ways to run FastTrax:

```
FASTTRAX [d:] /R          output report only  
                           and return to DOS
```

```
FASTTRAX [d:] /O >file   produce command file  
                           in 'directory' order  
                           and return to DOS
```

A visual report shows the current state of the disk. Each

# CONVERTING TO MS-DOS 3.1

[An anonymous user's experience in converting his floppy-disk Victor 9000 to MS-DOS 3.1; written as a set of instructions that he followed with his comments on the results. (Ed.)]

## I. Preliminary steps using MS-DOS 2.1:

1. Make back-ups of Victor-furnished original 3.1 disks and store them.
2. Copy-protect and store the back-up copies.
3. Print paper copies of both DISKIDS, both directories, and the CONFIG.SYS default file as delivered.
4. From your MS-DOS 2.1 working disk, make paper copy of the CONFIG.SYS file and directory for comparisons with 3.1.

II. Make a new MS-DOS 3.1 working disk. Make it double-sided, exclude the files least likely to be used, and configure it for your printer. This will be your basic, standardized disk to which application programs may be added.

1. Still in 2.1, DISKCOPY the two original 3.1 disks to temporary disks.

2. Print extra paper copies (DIR b:/p) of each 3.1 disk directory to study the uses of all the files and mark those to be deleted (unlikely to be used with present hardware).

3. Delete unwanted files from both temporary disks (i.e., all files relating to hard disks [e.g., DEL \*.cfg], leaving in doubtful ones not understood). I deleted these:

Disk 1: BACKUP.EXE; HDPARK.EXE;  
JOIN.EXE; RESTORE.EXE;  
SHARE.EXE -- (30 files left).

Disk 2: EXE2BIN.EXE; LINK.\*; AUTO\*.\*;  
\*.CFG (22 files left).

Total: 52 files (out of original 83).

4. Put temporary 3.1 (disk 1) in A: and re-boot.

5. FORMAT double-sided new disk in B:, double-sided, using:

FORMAT/m (it won't take /s to transfer SYS).

SYS a: b: (manual doesn't say so, but you have to specify both the source drive and the target drive); the SYS command transferred:

COMMAND.COM; CONFIG.SYS;  
CONFIG.BAT; PORTA.EXE;  
PORTB.EXE; PPORT.EXE.

COPY a: DISKID b: DISKID1

6. Insert temporary 3.1 (disk 2) in A:

COPY a: DISKID b: DISKID2

7. Return temporary 3.1 (disk 1) to A:

COPY a: \*.COM b:

COPY a: \*.EXE b:

8. Insert temporary 3.1 (disk 2) in A:

COPY a: \*.COM b:

COPY a: \*.EXE b:

COPY a: \*.XEC b:

COPY a: \*.BAS b:

**Result:** Double-sided working disk now has 52 files, instead of the 83 total on the two original disks; has 495,616 bytes free. 9. Re-boot with ver. 2.1, using disk containing Spellbinder. I put the new 3.1 working disk in B:, reconfigured as well as I could, but **failed** to get the printer to work. The following compare

1) the successfully-working 2.1 disk;

2) the default values of the delivered 3.1 disk; and

3) the revised 3.1 configuration.

1) My 2.1 successfully working config.:

buffers = 30

break = on

files = 20

country = 1

switchchar = /

device = porta.exe baud(1200) bits(7) stop(1) parity(even)

device = portb.exe baud(1200) bits(8) stop(2) parity(even)

device = pport.exe

shell = \command.com /p

2) 3.1 default config.:

buffers = 15

break = on

files = 20

country = 1

device = porta.exe baud(1200) bits(8) stop(1.5) parity(none)

device = portb.exe baud(1200) bits(8) stop(1.5) parity(none)

device = shell = \command.com /p

3) My new 3.1 config.:

buffers = 20

[compromised between 15 and 30; should make no difference]

break = on

[same as in both]

files = 20

[same as in both. I don't understand it, but it shouldn't matter now.]

country = 61

[changed to Australia from US, to get proper date format; thank God that Microsoft discovered Australia!]

device = porta.exe baud(1200) bits(7) stop(1) parity(even)

[changed to my 2.1 set-up -- should I have let it as in the 3.1 default??? What is 1.5 stop???

device = portb.exe baud(1200) bits(8) stop(2) parity(even)

[changed to my 2.1 set-up -- should I have let it as in the 3.1 default??? What is 1.5 stop???

```
device = pport.exe [same as in both]
shell = \command.com /p [same as in both]
```

The printer refused to work—it just flashed lights and shut down the keyboard, which then woke up again when the printer was turned off. This occurred when trying to TYPE from DOS or when using Spellbinder, doing the usual reset command, and issuing the print command.

However, the new setting for COUNTRY (#61) worked just fine when calling up the directory, setting DATE, and setting TIME. So, DOS is obviously **accepting** the new CONFIG.SYS file! Apparently, it is only the **printer** that isn't working....!

[I never have figured out the meaning of **bits** and **stop** settings, though I know vaguely what "bits" are. I've read purported explanations of bits and stops, but they don't make sense in terms of what to **do** about bit- and stop-settings. The 1.5 is a total mystery—I've never come across that before. I arrived at the various settings in my 2.1 disk by experimenting—the Spinwriter manual has a lot of technical theory, but no instructions about what settings to actually **use** or how to figure out what to use. Hackers seem to know all this by instinct and don't realize that it needs to be explained by **example** or by reference to particular machines.]

The CONFIG.BAT reads; "clst \dev\seriala" — is that OK? That's how it is in my working 2.1 and also in my 3.1.

My printer is the **NEC Spinwriter Model 3515, serial**. It is connected to the Serial Port A.

**SOLUTION:** I have discovered part of the trouble. When printing out the CONFIG.SYS file (with TYPE) it shows Porta as parity None, **even though** the CONFIG.SYS shows Porta as parity Even. According to the Manual Supplementary Information (page 2-5), the PORTSET should pick up the correct setting from the CONFIG.SYS. It evidently does **not!!!** Therefore, I created an AUTOEXEC.BAT file with the line "Portset a, Even". When I rebooted, it worked!!

Question: Why must I have the command in AUTOEXEC.BAT, printing out that chart? Why doesn't CONFIG.SYS pick it up like it is supposed to do?

### III. Remaining steps:

1. Put all the \*.COM and \*.EXE into a **subdirectory** (MD UTILS):

```
Establish a permanent PATH in an AUTOEXEC.BAT
to access them
(PATH=\UTILS):
```

Enter VERIFY ON

Move the files with COPY A:\\*.COM A:\UTILS, then check the subdirectory (CD UTILS) and then DIR.

If the \*.COM files are all safely in the subdirectory UTILS, then go back to the root directory (CD \) and remove them (DEL \\*.COM).

Then transfer all the \*.EXE files to the UTILS subdirectory by the same method, and maybe the \*.BAT files also. (It is convenient for future changes to put the AUTOEXEC.BAT file back in the root directory).

Then, put the Command and Port files back into the root directory, because the Path command will not be entered at boot-up—it must be in the root directory.

```
CD UTILS
```

```
COPY A:COMMAND.COM A:\
```

```
COPY A:PORTA.EXE A:\
```

```
COPY A:PORTB.EXE A:\
```

```
COPY A:PORTSET.COM A:\
```

```
COPY A:Pपोर्ट.EXE A:\
```

Then, **DELEte** those same files from the UTILS subdirectory after first seeing they are safely in the root directory.

2. DISKCOPY this new boot disk to several fresh disks, ready to receive applications programs.
3. COPY Spellbinder and the related word-processing macros to one of the configured working system/utilities disks. Copy other applications programs to the other new boot disks.
4. Set up a paper-copy journal to record all future changes in any working boot/applications disk and decide whether to make the same changes on the other boot/applications disks now or later.
5. Important note for your **data** disks: certain incompatibilities between 3.1 and 1.25 (especially) can be fatal, and some 1.25 command files are no longer needed. Therefore, go through all **data** disks and **DELEte** all \*.com files, \*.exe files, and especially the **chkdsk** file, and the system tracks. To be safe, get rid of everything that is not a strict data file. The most-used command files, such as COPY and DISKCOPY, are memory-resident (from the boot-up) rather than having to be separate external files listed in the directory. Note: as long as you are reviewing the directories of the data disks, you might as well assign **disk volume labels** using the internal VOL/C command.

#### IV. A final note:

Dale Capps (8 Dec 86) writes to say that he set up his hard-disk machine as follows—some of the same strategy could perhaps be used on my double-sided floppies??

#### CONFIG.BAT:

```
echo off
cls
clst \dev\serialb      [hard-disk only]
ports
```

PORTS.BAT is another batch file which sets his port-setting at boot-up:

```
echo off
PORTSET B 1200 Even 1 7
PORTSET A 9600 None 1.5 8
```

Then, his AUTOEXEC.BAT takes over thus:

```
echo off
cls
path a:\;a:\bin;c:\wp
scrnsave          [P.D. utility]
vt-iclk          [Valid Tech. board clock]
echo on
prompt $eyA$eyA$eyA$eyA$eyA$eyB$eyC
               [translation needed—no y in manual]
echo off
speedy4 /H      [P.D. utility]
cls
type menu.dos
prompt $p$g
```

He concluded that the ports must be set each time one boots up. His CONFIG.SYS file reads:

```
buffers = 30
break = on
files = 20
country = 1
switchar = /
device = porta.exe baud(9600) bits(8) stop(1.5)
               parity(none)
device = portb.exe baud(1200) bits(7) stop(1)
               parity(even)
device = pport.exe
shell = \command.com/p
```

... which he says does not seem to obviate the need to set the ports each time.

\* \* \*

#### FASTTRAX (continued from page 9)

square of the graph represents a small group of clusters. If all clusters in a group have a common characteristic (e.g., "free"), the report displays that characteristic. If the clusters have differing characteristics, FastTrax uses this priority table to determine which characteristic to display:

---

cluster is bad	(highest priority)
cluster belongs to a held file	
cluster belongs to a fragmented file	
cluster belongs to a consecutive file	
cluster belongs to an optimized file	
cluster is free	(lowest priority)

---

#### The optimizing process

The optimizing process consists of four phases:

Phase 1 — removes empty directory entries.

(When a file is erased, DOS does not delete the file's directory entry. Rather, DOS marks the entry as "empty" so it can be reused by a new file entry. By removing empty directory entries, DOS can search directories more quickly.)

Phase 2 — determines the final order.

Phase 3 — reorganizes the disk until it matches the final order.

Phase 4 — completes all deferred disk writes and returns to DOS.

Also presented are several suggested methods on how to organize your disk for maximum performance; detailed discussion of the general format of your disk; how DOS maintains directories and files; and, why disk performance slows down.

#### Concluding Comments

If you have a hard disk you should have a disk optimizer program. FastTrax is the best I've seen (for the Victor or the other machine) but FastTrax is the only commercial product I'm aware of that runs on the V9. FastTrax will run, not only on the Victor 9000 computer but also, on most other MS-DOS computers.

FastTrax is available mail order from Mark Elfield & Associates, Inc., 4206 Terrace Street, Oakland, CA 94611 Tel. (415) 652-2231

\* \* \*

## RAM SIZE LIMITS - VICTOR 9000 VS. IBM

Q: I keep reading about how MS-DOS is limited to using 640K of RAM. Is this true? If so, why can the Victor 9000 can accept 896K?

A: The 640K RAM limit you read about is a limit imposed by the IBM-PC, not MS-DOS (or IBM's PC-DOS). The limitation on the amount of allowable free RAM (Random Access Memory) is set by the machine architecture.

Memory for the 8088 processor is divided into sixteen 64K segments numbered 1 through F (in hex). Both the IBM-PC and the Victor 9000 reserve areas of high memory for certain system resources.

IBM reserves segment F (the top 64K of RAM) for ROM (Read Only Memory). This ROM holds the boot ROM routines, ROM BIOS (Basic Input/Output System), and ROM BASIC (cassette BASIC). IBM then reserves segments D and E for plug-in ROM cartridges. These are used only on the IBM-PC/Jr. Segment C is reserved for add-on system ROM for peripherals such as the hard disk in the IBM-PC/XT. Finally, segments A and B are reserved for 128K of Screen RAM. That is where the the screen image is stored for display.

This leaves segments 0 through 9 for user RAM. Ten 64K segments is 640K. This space must hold DOS and any user programs. As you can see, IBM likes to reserve a lot of room for its own use.

The Victor 9000 has a much different design philosophy. It provides for as much user RAM as possible, allocating only two segments for system use. Segment F on the Victor is used for 32K of boot ROM and 4K of Screen RAM. Segment E is reserved for Memory Mapped I/O (communication with peripheral devices). This leaves segments 0 through D for user RAM. Fourteen 64K segments is 896K.

But unlike the IBM, this space is not all available for DOS and the user. The Victor 9000 does not have ROM BIOS, but rather loads the BIOS into RAM at boot time. Also the Victor does not have a character set ROM, but loads the character sets into RAM at boot time. In graphics mode, more than the 4K of Screen RAM is needed, and this too is taken from user RAM.

This use of RAM instead of ROM gives the Victor 9000 superior flexibility. Victor can release a new version of the BIOS without having to replace ROM chips. We, the users, can install our own character sets and keyboard files using MODCON. With 896K of RAM you can easily allocate a 620K RAM disk.

The trade off for this flexibility is that in the same size machine (say 256K), the IBM has more AVAILABLE memory, because it has more built-in system resources. This is why dBase III on the IBM-PC requires 256K, but on the Victor 9000 it requires 320K. The actual requirement is for 200K of free RAM. However, this gap is narrowing because IBM is finding it necessary to put more of their BIOS into RAM like Victor did from the start.

---

### VICTOR 9000 SERVICE

Victor Technologies is offering a group discount (dealer prices) for service and upgrades thru February 28, 1987. For current prices or other info contact your area service manager:

Western Regional Repair Depot  
380 El Pueblo Road  
Scotts Valley, CA 95066  
(408) 438-6680 Ext. 283  
Gary Tyler, Area Manager

Central Regional Repair Depot  
565 Busse Road  
Elk Grove Village, IL 60007  
(312) 364-8950  
Richard Seeman, Area Manager

Eastern Regional Repair Depot  
2166 W. Park Place, Suite C  
Stone Mountain, GA 30087  
(404) 498-6175  
Barry Hartman, Area Manager

The Western region covers the eleven Western states, AK, HI, and Mexico; the Central region covers AR, IA, IL, IN, KS, LA, MI, MN, MO, MS, NB, ND, OH, OK, SD, TX, WI and Canada; the Eastern region covers the remaining states, PR and Caribbean states.

---

### VICTOR PlusPC

If you have trouble writing to Victor format disk on the PlusPC, but have no trouble writing to IBM format disks, the floppy disk controller may have a bad 3090F chip. For help contact your nearest repair depot and request part number 104738-01.

---

Re: VENDORS AND SOURCES, VictorTalk Vol. 4, No. 6. Merton Hill & Associates writes to say "... with considerable sadness ... with the exception of local customers in our immediate area, we are no longer supporting the Victor 9000." They go on to say, "Our customers have been loyal to a fault, and no retailer ever had more honest and reliable customers. In four years, we have not had a single unpaid invoice or rubber check from a Victor owner. Few retailers in any line can claim that kind of payment record from their customers."

\* \* \*

# MS-BASIC, Part 1 / Jack Prichard

Why MS-BASIC? Because it's there! This facetious answer is half correct. We Victor 9000 owners all have MS-BASIC as part of the software that was provided with our machine. The other half of the correct answer is that MS-BASIC is an extremely powerful tool with which we can accomplish an infinity of useful things. Many of these are things that we may not be able to do with our various word processors, spreadsheets, data bases, etc. Thus, it seems that all serious users should have at least some rudimentary knowledge of, and ability with, MS-BASIC. This series of articles is a means of acquiring that knowledge and ability in a reasonably painless manner.

This first article of the series on MSBASIC will describe (in order) how to:

1. Run an existing MSBASIC (.BAS) program.
2. Examine an existing program.
3. Create a simple program with screen output.
4. Save a program to disk.
5. Create a simple program with printer output.
6. Combine two programs.
7. Create a practical program.

## MSBASIC/GWBASIC/VBASICA

There are at least two versions of MSBASIC. (For simplicity, the program file name form is used for the remainder of this article.) The one received with MS-DOS 1.25 is identified as "BASIC-86 Rev. 5.27, created 11/8/82"; that with MS-DOS 2.11 as "MS-BASIC Version 5.28, created 5/24/83". While there are obviously differences between the two because of the different operation systems, there are no readily discernable differences to the user. Thus, the information in this series of articles about MSBASIC applies to both versions unless otherwise noted. There are other closely related versions. GWBASIC is an expanded version of MSBASIC, having commands for graphics and music; it is available with one or more of the Victor Toolkits. VBASICA (Victor BASIC Advanced) is an IBM PC-compatible extension to MSBASIC, similarly having graphics and music capabilities; it is provided with MS-DOS 3.1 for Victor 9000 computers. While most of the material covered in this article is also applicable to these last two versions of BASIC, no attempt is made to cover them at this time. It is hoped to do so at the end of this series of articles.

MSBASIC is Microsoft's version of generic BASIC (Beginner's All-purpose Symbolic Instruction Code), originally developed at Dartmouth College. MSBASIC, unlike most other high-level programming languages is an "interpretive" language, as opposed to a "compiled" language. That is, as a

program in MSBASIC is being run the computer interprets the language into machine language, essentially line-by-line. Thus, it runs more slowly than do compiled languages, which have previously been translated into machine language by a one-time compiling run. MSBASIC does have the advantage of being simple to modify, and rerun, whereas a compiled language program requires recompilation after a modification is made to the original high-level language program.

Note: Where speed of running is important (as in a program entailing much computation, sorting, or other similar time-consuming operations) MSBASIC can be compiled and linked to obtain a much faster-running machine-language program (one with the extension .EXE or .COM). However, the programs to accomplish this are not furnished with the Victor 9000 computer, but must be purchased separately. The use of the MSBASIC compiler was well covered by R.W.Johnson. (See VictorTalk, Vol 3, No 5, May 1985.)

## PROGRAMMING HELPS

A key part of this series is the inclusion of tables of commands, statements, and functions arranged in a particularly useful way. Rather than simply being arranged in alphabetical order, they are arranged by categories of tasks that they perform. Thus, when you want to do some particular task but you have forgotten (or perhaps never knew) the appropriate command, you can find it in a short list under the appropriate category. This is certainly much easier than wading thru a long alphabetically-arranged list. Further, the tables include not only the "name" of the command, statement or function, but also two highly useful additional pieces of information: the format for its use and page number references (in the Victor documentation) to details on its purpose and examples of its use. The table of commands and statements is included in this article; the table of functions will be included in a later article.

As you proceed thru the several exercises in this article, use the following steps:

1. Read thru the first exercise.
2. Reread the first exercise, noting each new command or statement introduced. (For simplicity this will be shortened to "command".)
3. Look up each command in Table 1. (Almost all are in the first column of the first page of the table.)
4. Note the format(s) used with the command.
5. Read the referenced portion of the Victor documentation.
6. Carry out the instructions for the first exercise on your computer.

7. Proceed to the next exercise and repeat the above steps.
8. Similarly, continue with each exercise until done.

Before proceeding with the exercises, however, the two modes of operation to be used will be described:

In the direct mode, commands and statements are not preceded by a line number and are executed immediately; while the results may be stored in memory, the commands and statements themselves are lost upon execution.

In the indirect mode, commands and statements are preceded by a line number and are stored in memory as part of a program.

### RUNNING AN EXISTING MSBASIC PROGRAM

In this section and the following one (Examining an Existing Program), enter all commands in the direct mode; that is, without line numbers. Also, end all commands with a RETURN; however, for simplicity and clarity, RETURNS are not shown here. Enter commands in either upper case or lower case letters; for clarity, upper case is used here.

Select a program written in MSBASIC for running and examining. Your Public Domain library is a good source. The program will have the suffix .BAS. (However, GWBASIC programs (which also have the .BAS suffix) will generally not run in MSBASIC. For convenience of reference, the selected program will be referred to as "TEST.BAS" in the following exercises.

To run the selected program, if at system level (as indicated by an MS-DOS prompt such as A>), type:

```
MSBASIC TEST
```

For this to work, MSBASIC.COM must exist in the default drive or a proper path to it must have been previously provided and TEST.BAS must exist in the default drive. If, for example, MSBASIC.COM is on the A: drive and TEST.BAS is on the B: drive, proper operation would be assured regardless of which was the default drive, by typing:

```
A:MSBASIC B:TEST
```

Note that in either case it is unnecessary to use the suffix .BAS.

Alternatively, this same result could be obtained by two separate steps. Type:

```
MSBASIC
```

Then, after MSBASIC is loaded (as indicated by the legend and the MSBASIC prompt Ok) type:

```
LOAD "B:TEST.BAS",R
```

This would load and run TEST.BAS.

If B: were the default drive, the command could be further abbreviated:

```
LOAD "TEST.BAS",R
```

In the event you do not want to immediately run the program

,R would be omitted. Thus, type:

```
LOAD "B:TEST.BAS" or LOAD "TEST.BAS", as appropriate.
```

These could be further abbreviated as:

```
LOAD "B:TEST" or LOAD "TEST", as appropriate.
```

Note carefully the use of double quotes. These are required with all commands involving a filename (CHAIN, FILES, KILL, LOAD, MERGE, NAME, OPEN, RUN, SAVE).

When ready to run, type:

```
RUN
```

If, during the running of a program you want to temporarily halt, type ALT-C by simultaneously pressing the ALT key and the C key (either upper or lower case). The computer will respond by displaying:

```
Break in line nnnnn
```

MSBASIC returns to command level (Ok prompt). It is then possible to display or print the value of variables, etc., as will be described later. When ready to resume running the program from the point at which it was stopped, type:

```
CONT
```

To restart at line nnnn, type:

```
GOTO nnnn or RUN nnnn
```

To restart from the beginning, type:

```
RUN
```

When done with MSBASIC and you want to return to the system (MS-DOS) level, type:

```
SYSTEM
```

The return to system level is then indicated by the MS-DOS prompt (A>, or whatever other prompt symbol has been selected previously).

Unless you have already done so, turn to Table 1 on the following two pages. Look up each of the commands you have just used. Then, unless you are already familiar with their use, read the referenced pages in the documentation supplied with the Victor 9000 computer.

'82 = "User's Guide for MS-BASIC", first Victor printing December, 1982 and "Reference Guide for MS-BASIC". These two parts were issued together in a looseleaf binder identified as "BASIC 86 User's Guide". Most page number references are to Chapter 2 of the Reference Guide. For brevity the 2- portion of these page numbers is omitted; other pages are identified as in the guides; UG stands for User's Guide.

'83 = "MS-BASIC", first Victor printing April, 1983. This manual, issued in a looseleaf binder also identified as "MS-BASIC", contains essentially the same material as the earlier manual but with a single set of consecutively numbered pages, I-VII and 1-122.





	Page Ref.	
	'82	'83
<b>DATA I/O</b>		
DATA<list of constants>	13	22
INPUT[;][<"prompt string">;] <list of variables>	37	37
INPUT#<file number>, <variable list>	39	38
[LET]<variable>=<expression>	41	40
LINE INPUT[;][<"prompt string">;] <string variable>	42	40
LINE INPUT#<file number>, <string variable>	43	41
LPRINT[<list of expressions>]	48	43
LPRINT USING<string exp>; <list of expressions>	48	43
NULL<integer expression>	54	46
OUT I, J where I and J are integer expressions, range 0-255	59	49
also UG 2-4		
PRINT[<list of expressions>]	61	49
PRINT USING<string exp>; <list of expressions>	63	51
PRINT#<file number>,[USING <string exp>;]<list of exps>	67	54
READ<list of variables>	71	57
RESTORE[<line number>]	75	60
WAIT<port number>, I[, J] where I and J are integer expressions	82	65
WIDTH[LrPINT]<integer exp>	84	66
WRITE[<list of expressions>]	85	67
WRITE#<file number>,<list of expressions>	86	67

<b>DATA I/O (Random File)</b>	UG4-6	93
FIELD[#]<file number>,<field width> AS <string var>	29	32
GET[[#]<file number> [,<record number>]	32	34
LSET<string variable>= <string expression>	49	43
PUT[#]<file number> [,<record number>]	69	56
RSET<string variable>= <string expression>	49	43

#### PROGRAM LINKAGE

CALL<variable name> [(<argument list>)]	3	16
also UG 2-3	104	
CHAIN[MERGE]<filename> [,<line number exp>] [,ALL][,DELETE<range>]]	4	17
also see note p5		
DEF SEG[=<address>]	UG 2-1	
DEF USR[<digit>]=<integer exp>	17	25
PEEK(I) (function)	3-16	80
POKE I, J where I and J are integer expressions	60	49
USR[<digit>][(<argument>)]	UG 2-5	108
UG 5-7		
RG 3-23	25	

#### DATA MANIPULATION

DEF SEG[=<address>]	UG 2-1	25
MID\$(<string exp1>,n[,m])= <string exp2>	51	45
PEEK(I) (function)	3-16	80
POKE I, J where I and J are integer expressions	60	49
RANDOMIZE[<expression>]	70	57
SWAP<variable>,<variable>	80	63

#### ERROR HANDLING

ERROR<integer expression>	27	31
IF ERR=<error code> THEN...	26	31
IF ERL=<line number> THEN...	26	31
ON ERROR GOTO<line number>	55	46
RESUME	76	61
RESUME [0]	76	61
RESUME NEXT	76	61
RESUME<line number>	76	61

#### Notation used in above formats:

CAPS Capital letters - portions of commands or statements that must be entered, exactly as shown.  
 [ ] Square brackets - enclosed entry is optional.  
 < > Angle brackets - user entered data.  
 | Vertical bar - separates two alternate items.  
 . . . Ellipses - entry may be repeated as many times as needed.  
 All other punctuation (commas, semi-commas, colons, slash marks, and equal signs) must be entered exactly as shown.

## EXAMINING AN EXISTING PROGRAM

When in MSBASIC, to display the listing of a MSBASIC program that has been loaded into memory, type:

LIST

The entire listing will then be scrolled on the display. Scrolling can be stopped by pressing ALT-S; it can be restarted by pressing ALT-S again. To examine segments of the listing, it may be more convenient to use one of the following forms of the LIST command:

For only line 50, type:

LIST 50

For only thru line 100, type:

LIST -100

For lines 400 thru 500, type:

LIST 400-500

For all lines from 800 on, type:

LIST 800-

A period stands for the current line number. Thus, to list all lines up to the current line, type:

LIST - .

Listings may be output to the printer by using corresponding versions of the command LLIST.

## CREATING A SIMPLE PROGRAM WITH SCREEN OUTPUT

As was seen from the listing of a MSBASIC program, such programs consist of a series of sequentially numbered lines of commands, statements, and/or remarks. To prepare for creating a new program, first clear memory of any existing program (unless that program is to be a part of the new program). To clear memory, type:

NEW

The operation of this command can be checked by again listing the contents of memory. You will find nothing there, as indicated by the immediate appearance of only the command level prompt, Ok

Before creating a program, try the following PRINT and LPRINT commands in the direct mode. Text is displayed by the command PRINT followed by the text to be displayed in double quotes. Type:

PRINT "PRINT displays text in quotes"

Similarly, text is printed by the command LPRINT followed by the material to be printed in double quotes. (Be sure your printer is turned on and on line, else the computer will hang up until the printer is ready.) Type:

LPRINT "LPRINT prints text in quotes"

Now write a simple one-line program to do the first of these tasks just accomplished in the indirect (program) mode. Type:

10 PRINT "PRINT displays text in quotes"

RUN

You have now created a program, run it and seen its result.

## SAVING A PROGRAM TO DISK

Type:

SAVE "B:TEST1.BAS",A

You have now saved a program as a .BAS file. The .BAS suffix identifies the file as in the BASIC language. The added ,A signifies that the file is to be stored in ASCII form. This allows the file to be merged with other files (as will be immediately demonstrated); also, to be read directly under MS-DOS using the TYPE command.

## CREATING A SIMPLE PROGRAM WITH PRINTER OUTPUT

Type:

NEW

20 LPRINT "LPRINT prints text in quotes"

RUN

SAVE "B:TEST2.BAS",A

You have cleared memory, created a second program, run it (and observed the result) and saved it as a .BAS file.

## COMBINING TWO PROGRAMS

To demonstrate how two programs can be merged, type:

MERGE "B:TEST1.BAS" or simply MERGE "B:TEST1

RUN

SAVE "B:TEST3.BAS",A

NEW

LOAD "B:TEST3

RUN

You have merged the two programs into one program, run it (and observed the results), saved it as TEST3.BAS. You then cleared memory, loaded the new program and ran it (again observing its results).

Note that two programs are MERGED in accordance with their line numbers. If both programs have one or more common line numbers, the line from the MERGED program overwrites the corresponding line of the program previously in memory.

## CREATING A PRACTICAL PROGRAM

As the finale for this article, we will create a practical program, using a number of the commands already introduced, plus a few new ones. Let us assume you want a program to control the various fonts available on your dot-matrix printer. First, you may look thru the Public Domain disks that you have to see if there is a program that will do as you want. Assume that you can not find what you want and decide to write your own program. From the instruction manual for your printer determine the control codes needed to achieve the desired

results. For an Epson MX-80 printer the "character designation codes" and their corresponding control codes are:

```
SO      CHR$(14)      Enlarged print on
DC4     CHR$(20)      " " off
SI      CHR$(15)      Condensed " on
DC2     CHR$(18)      " " off
ESC E   CHR$(27) CHR$(69) Emphasized " on
ESC F   CHR$(27) CHR$(70) " " off
```

Note, however, that enlarged printing can be turned on for only one line at a time as the line feed character (LF) also turns the mode off. Thus, enlarged printing can only be obtained by combining the control code and the text involved in a single LPRINT command. For example, enlarged printing of the word TEST will result from execution (in either the direct mode or in the indirect mode) of:

```
LPRINT CHR$(14) "TEST"
```

Try this, and variants of it, to assure yourself that this is true.

Because of this limitation, enlarged printing is not included in the menu of print modes. Instead, double-strike printing is included. (Oddly, this is undocumented in the Epson Operation Manual.)

```
ESC G   CHR$(27) CHR$(71) Double-strike print on
ESC H   CHR$(27) CHR$(72) " " off
```

The desired result might be achieved in two different ways: Each of these control codes might be put into a separate program (with a separate batch program to run it from the system level). Alternatively, one program could allow selection from a menu.

For the separate program solution, create and SAVE the program B:PRTDSN.BAS (standing for PRinT Double-Strike on):

```
10 LPRINT CHR$(27) CHR$(71);
20 SYSTEM
```

Note the semicolon at the end of line 10; its purpose is to prevent an automatic line feed being sent to the printer. (More about this later.)

Write a batch file, B:PRTDSN.BAT, to run the .BAS file from the system level, using the system command COPY CON, as follows:

```
COPY CON B:PRTDSN.BAT
A:MSBASIC B:PRTDSN
ALT-Z (Simultaneously press the ALT key and Z)
```

Then, to invoke the double-strike mode from the system level, type:

```
PRTDSN
```

This will call up the batch file PRTDSN.BAT, which in turn calls up MSBASIC, loads and runs PRTDSN.BAS, setting the desired mode, and returns to system level.

Similar .BAS and .BAT files could then be written to invoke the remaining five "character designation codes".

The alternative approach, using one program with a menu, might be written as follows:

```
10 REM - PRIMODE.BAS-Controls printer mode
20 PRINT "Select one of of following:"
30 PRINT " 0 - Return to system level"
40 PRINT " 1 - Double-strike print on"
50 PRINT " 2 - Double-strike print off"
60 PRINT " 3 - Condensed print on"
70 PRINT " 4 - Condensed print off"
80 PRINT " 5 - Emphasized print on"
90 PRINT " 6 - Emphasized print off"
100 INPUT;"",N
110 IF N=0 THEN SYSTEM
120 IF N=1 THEN LPRINT CHR$(27) CHR$(71);
130 IF N=2 THEN LPRINT CHR$(27) CHR$(72);
140 IF N=3 THEN LPRINT CHR$(15);
150 IF N=4 THEN LPRINT CHR$(18);
160 IF N=5 THEN LPRINT CHR$(27) CHR$(69);
170 IF N=6 THEN LPRINT CHR$(27) CHR$(70);
180 SYSTEM
```

An alternative coding for the last seven lines, using the command ON . . . GOTO:

```
120 ON N GOTO 130, 140, 150, 160, 170, 180
130 LPRINT CHR$(27) CHR$(71);: SYSTEM
140 LPRINT CHR$(27) CHR$(72);: SYSTEM
150 LPRINT CHR$(15);: SYSTEM
160 LPRINT CHR$(18);: SYSTEM
170 LPRINT CHR$(27) CHR$(69);: SYSTEM
180 LPRINT CHR$(27) CHR$(70);: SYSTEM
```

In line 100, the semicolon is used to avoid echoing a carriage return/line feed sequence when the carriage return is typed to input the selection. The pair of double quotes is used to indicate a blank prompt, and the comma to suppress the question mark that would otherwise be displayed. In lines after 110, the semicolon is used to avoid echoing a carriage return/line feed sequence when the printer operation is completed. For reasons not understood, this desired result is not obtained with the first version (using IF . . . THEN command), while it is obtained with the second version (using ON . . . GOTO command). However, both versions properly set the desired mode.

The lines of a program may be typed in any desired sequence when line numbers are manually typed. MSBASIC will automatically arrange them in numerical sequence. Alternatively, lines can be automatically numbered by typing:

```
AUTO
```

This will produce the line sequence 10, 20, 30, 40, . . . . If a different starting line number, line number increment, or both

is desired, the general format may be used:

AUTO[<line number>[,<increment>]]

(The symbolism used here is the same as used in the Victor documentation and in Table 1.) For example, AUTO 100, 20 will produce the sequence 100, 120, 140, 160, . . . . The automatic line numbering can be stopped by ALT-C, which returns to command level.

If typing errors are discovered after a line is entered, wait until you are done typing the entire program before attempting corrections. Then go back and correct your errors using the command EDIT <line number>. Read the referenced pages in the Victor documentation. You will find most editing can be done with a few simple commands, listed here with their mnemonics:

- I Insert
- X Extend
- D Delete
- C Change
- L List (remainder) of line
- H Hold (prior characters) and add
- Q Quit edit without changes
- A Again (restore and re-edit)

The remaining commands (S, K and E) are used only occasionally.

When satisfied that the typing is correct, save the program as B:PRTMODE.BAS (standing for PRinT MODE). Then try using the program. If it works as intended, great! If not, look it over carefully and make necessary corrections until it does run correctly. Then save it again.

In checking the operation of your program you have probably discovered that some combinations of modes are usable. Thus, some ten different-appearing fonts can be obtained. In order, from smallest and lightest to biggest and darkest, the fonts are:

CONDENSED  
CONDENSED DOUBLE-STRIKE  
NORMAL  
DOUBLE-STRIKE  
EMPHASIZED  
EMPHASIZED DOUBLE-STRIKE  
ENLARGED  
ENL. DOUB. -STR.  
ENL. EMPHASIZED  
ENL. EMPH. D-S

The combinations not listed do not work. As a final exercise, try duplicating the above sampler (of course with such variations as are required with your particular printer).

## SUMMARY/PREVIEW

This article has introduced a vast subject. While it has only scratched the surface, hopefully, it may have helped you get over the initial fear of the unknown. If you have faithfully slugged thru the several examples, you can, at least, run and view existing MSBASIC programs. Further, you have some faint knowledge of what an elementary program consists and how to create and save such a program. Not too bad a beginning! Even more, you have a powerful tool in Table 1 that capsulates considerable essential information in convenient form. With it you have a good start into proceeding on you own into the futher intricacies of MSBASIC.

However, it is our intention to provide considerable further hand-holding assistance. Part 2 of this series will extend the notions already introduced on controlling a printer's font capabilities. Its central theme will be the creation of a program to print a typical business form in a specified format - a bit beyond the capabilities of most spreadsheet or data base programs. In doing this, more commands will be introduced, making further use of Table 1. Also, along the way, a variety of "hints and kinks" on writing programs will be introduced.

In later parts of this series, the subject of files, both sequential and random, will be delt with at some length. At an appropriate point, a Table of Functions will be introduced. This will be in the form of Table 1, giving each function, its format, and references to the Victor documentation. As different MSBASIC subjects are introduced, it is intended that all categories of commands, statements, and functions will be covered.

In addition to the key functions already introduced (running, listing, creating and saving programs), there are other important functions yet to be covered: debugging and modifying programs. Frequently a Public Domain program can be found that almost (but not quite) does what you want it to do. If the program is written in MSBASIC (or some other version of BASIC, for that matter), it may be entirely practical to modify it so that it does exactly what you want. This approach to solving a particular problem may just be a bit simpler than reinventing the wheel, so as to speak!

If you haven't been previously involved in programming, you may soon develop a glorious new feeling. When things begin, at last, to fall into place, you feel the master of your computer, rather than its slave. Of course, that's not really the case, but it is exhilarating to occasionally believe you're on top!

\* \* \*

## QUESTIONS, QUESTIONS, QUESTIONS / John Knobelsdorf

By the time this article is published in VictorTalk, a lot of you will have heard that I have retired, and Mary Jo and I will be moving back to Texas. Why are we moving back there? Because we have a home on some rural property in the hills of central Texas, and we will be close to our children and grandchildren.

We have enjoyed our work with Victor\*Group, the work with the public domain, the work with VictorTalk, and most importantly, our association with all of the friends we have found among all you users of the Victor 9000. There seems to be something particularly nice about all V9000 owners and operators, from those in Scotts Valley, to Canada, the East Coast, Europe, Australia and return.

Our enjoyment with this work and the contacts we have made domestically and internationally has led to our agreement to continue handling the public domain library for Victor\*Group from our location in Texas. We will still be able to visit with you via modem, bulletin board, hardcopy correspondence, telephone and diskette, and we are looking forward to doing that.

We hope to organize a Texas Regional sub-chapter of Victor\*Group, and will invite everyone we can find to join in our effort.

The Victor\*Group public domain library continues to expand. In January 1987 we released four more new disks. They are #78, 79, 80 & 81. You will find the usual brief description of these disks in paragraphs that follow.

We suggest again, that the first thing you should do when you obtain a public domain disk is look at the DISKID (disk identification) to get an idea of the kinds of files that are on the disk and what the various files and programs will do for you.

In the case of the last several public domain disks, the file name of the disk identification is DISKIDnn.TXT, where "nn" is the disk number. On earlier disks, the disk identification file name might be simply DISKID.TXT or DISKID.DOC or README or anything else that might have seemed appropriate to the "librarian of record" who was describing the contents of the disk.

If you are uncertain of the name of the disk identification file, look at a directory listing of the files on the disk to determine the correct name of the disk identification.

My reason for suggesting that you read the disk identification

files, particularly for new disks, is that the library will now include files and programs which are in different format than we have been using.

Beginning officially with PD #78, some, many or all of the programs and files on a public domain disk will be in compressed format. It will be necessary for the user to de-compress the file or program before it can be used. As the telephone information recording states, "please make a note of it".

What is a "compressed file"? How do you recognize it? How does one de-compress a file? All of those are very good questions! The answers may be found in an article named Compressed Files which appeared in VICTORTALK, Vol. 4, No. 5. The utility programs which are used to compress and/or de-compress a file were listed at the end of the article, and are available on various of the public domain disks (i.e. #31,36,46,49,57,60,67,68,70 & 73). A copy of the article, Compressed Files, is also included on PD #73 for your convenience.

Why are we going to have compressed files on public domain disks? There are several reasons. The biggest and best reason is that almost all compressed files occupy less disk space, and we are thus able to supply you with more files and programs per disk. Other reasons relate to the time and expense involved with obtaining the files, tracking file names, sorting the files onto new disks, and developing the new disk's identification file.

Other than new or revised versions of compress/de-compress utility programs which will be put on disk the same as any other new file or program, we will not include any compress/de-compress programs on a new disk simply for the users convenience of access. Why? Why? Why? Three good questions, and three good answers. The de-compress programs would occupy disk space that could be used for other new files and programs. They are readily available to the user on other PD disks, and should be kept even more available on the users own utility disk or in a "bin" on hard disk. And, since any compressed file on a public domain disk should be moved to a work disk before it is de-compressed (to avoid possible complications - as explained by the Compressed File article), the de-compress utility would also have to be moved to the work disk, and thus, there should not be any additional great burden placed on the user to have to call up the de-compress utility from the user's source.

If after reading the Compressed Files article, and reading the documentation files furnished with most of the compress/de-

compress utilities, and reading and following the directions included in some of the compress/de-compress utility programs, anyone has any difficulty using the compressed files on the new public domain disks, he/she may call on me for assistance. I may ask you first if you have done your homework, but I will be glad to help.

Now we go on to the brief description of the current new disks.

**DISK #78.** Has a set of Pascal Routines & Utilities - from Bert Happel. They are: BEEP.PAS, Like Beep in Basic; BITCHG.PAS, Change bits in integers; CLRLINE.PAS, Clear line from screen; EXEC.PQS, Allows calls to DOS; GETDATE.PAS, Get current System Date; GETKBD.PAS, Ascii/Scan code from keyboard; HELP.PAS, Pascal source code for HELP; INDENT.PQS, Reformat Turbo Source code; MOUSE.PAS, Mouse routines; OPTICS.PAS, Pascal source code for OPTICS; PASCLIB.LBR, Hispeed Ft Pt Sublib MS Pascal & A/L; PRINT.PAS, Print files with page break/number; PTOOLDAT.LQR, Date tools; PTOOLENT.LQR, Data entry tool; PTOOLSQR.DQC, Extended doc for PTOOLSQR.LQR; PTOOLSQR.LQR, Data entry-use with PTOOLENT; PTOOLTIM.LQR, Time tools; PTOOLWI2.LQR, Windowing tools version 2; PTOOLWIN.LQR, Windowing tools; REPSTR.PAS, Like STRING\$ in Basic; TOOLKIT.PAS, Tools; TURBUTIL.PAS, utilities from BYTENET.

TPTUTOR1.ARC, a public domain tutor for Turbo Pascal. Files in this archived file are: TP1READ.ME, PRINT1TL.BAT, PROG1.PAS, PROG10.PAS, PROG2.PAS, PROG2A.PAS, PROG3.PAS, PROG4.PAS, PROG5.PAS, PROG6.PAS, PROG7.PAS, PROG8.PAS, PROG9.PAS, TEST1.PAS, TL01.TXT, TL02.TXT, TL03.TXT, TL04.TXT, TL05.TXT, TL06.TXT, TL07.TXT, TL08.TXT, TL09.TXT, TL10.TXT,

TPTUTOR2.ARC, Part 2 of Turbo tutor. Files in this archived file are: TP2READ.ME, PRINT2TL.BAT, PROG11.PAS, PROG12.PAS, PROG12A.PAS, PROG13.PAS, PROG13A.PAS, PROG14.PAS, PROG15.PAS, PROG15A.PAS, PROG15B.PAS, PROG15C.PAS, PROG16.PAS, PROG17.PAS, TIME.INC, TL11.TXT, TL12.TXT, TL13.TXT, TL14.TXT, TL15.TXT, TL16.TXT, TL17.TXT.

**DISK #79.** This disk is primarily a set of BASIC Tutorials - Donations from Bert Happel. Also on the disk are: EMIBM.COM, an IBM emulator that is used to run BIO.COM and PCDISK.COM; BIO.COM, biorythm program; BIO.TXT, support file for BIO.COM; PCDISK.COM, a program to catalog files on your disks; PCDISK.NOT, notes re use of PCDISK - read this before using; CATALOG.BAT, Happel's batch file to use PCDISK; SR.ASM, source code for SR.COM; SR.COM, program to rename files and/or sub-directories; SR.DOC, documentation for SR.COM; DOS31USE.DOC, a text re use of

PC-DOS 3.1; ILUV9000.TXT, Happel loves his V9000; VDT9000.DOC, an article re effects on one's vision of VDT use; VT O&E.TXT, odds and ends notes re use of V9000; BMASCII.DOC, info re converting Benchmark files to ASCII files; CHK4BOMB.DOC, doc for CHK4BOMB.EXE; CHK4BOMB.EXE, use to check a "suspect" file for a worm or virus; MAPPER.DOC, doc file for MAPPER.EXE; MAPPER.EXE, displays a map of the IBM-PC's 1mb RAM; KRMITARC.DOC, documentation for use with KERMIT.ARC; KRMITLBR.DOC, documentation for use with KERMIT.LBR; USEANSI.ARC, archive form of USEANSI.TXT on how to use ANSI 3.64; XEBEC.ARC, text file re use of Xebec's hard disk controller; WSSI300.ARC, a disk library manager - current version 10/25/86; FIDOFILS.DOC, A list of files used with a Fido BBS.

**DISK #80.** A set of graphics programs for the Okidata and other graphics printers, and provided to us by E. Harding of Fresno, CA. They are: VDRAW.EXE, the drawing program; VDRAW.DAT, the data file used by VDRAW.EXE; OKIFONT.EXE, used to design character sets for Okidata 92 printer; GREEK.OKI, a character set produced by using OKIFONT.EXE; ITALICS.OKI, 3840 character set produced by using OKIFONT.EXE. There are a number of help files and character sets on this disk. The character sets used by VDRAW.EXE are: FUTURE.CHR, THIN.CHR, MED.CHR, HINPROP.CHR, SCRIPT.CHR, MEDPROP.CHR, GREEK.CHR, SMPROP.CHR, OCR.CHR, LGPROP.CHR, MEDVERT.CHR, LGVERT.CHR, SMPROPV.CHR, SMVERT.CHR, SMALL.CHR, THINVERT.CHR, LGPROPV.CHR, MEDPROPV.CHR, THNPROPV.CHR, SCRIPTV.CHR, GREEKV.CHR, FUTUREV.CHR, GRAPHICS.CHR, USER1.CHR, USER2.CHR, USER3.CHR, LARGE.CHR, NORMVERT.CHR, NORMPROP.CHR, NORMPRPV.CHR.

A set of Graphics Routines (MS-BASIC) provided to us by J. P. Hastey, Jr. They are: GRAFPAK.ASM, the assembly language source code; GRAFPAK.EXE, the executable program; GRAFPAK.DOC, documentation file; GRAFPAK.CHR, a sample character set with 19 fill patterns; PAKDEMO.BAS, an MS-BASIC demo program; PAK.BAT, a batchfile that runs the PAKDEMO.BAS.

CHRKB.DOC, an article written by J. P. Hastey, Jr. that describes methods for changing character sets and keyboards. PC-TOUCH, a Typing Tutor program from the BAVUG bulletin board. FREEWILL.ARC, statutory will forms for California residents to prepare a "fill in the blanks" statutory will. (Use ARCE to unarchive this file).

**DISK #81.** There are 55 picture files (JPHxx.SCR) that were converted from a Victor Technologies CP/M-86 slide show demo to the MS-DOS format by James P. Hastey, Jr. for use

(continued on page 24)

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
P.O. Box 50893  
Palo Alto, CA 94303

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
City _____ State & Zip _____	_____
Country _____ Phone _____	Order shipped _____

Circle diskette numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60. See Vol. 4, No. 2 for a brief description of disks 61-64; Vol. 4, No. 3 for disks 65-71; Vol. 4, No. 6 for disks 72-77; and Vol 5, No. 1 for disks 78-81.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34*	35*	36	37	38	39	40
*(34 & 35 are two disk set)									
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56*	57*	58	59	60
*(56 & 57 are a two disk set)									
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	—	—	—	—	—	—	—	—	—

1987 Victor\*Group Member: No \_\_\_\_\_ Yes \_\_\_\_\_ Date joined \_\_\_\_\_

Number of diskettes ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of diskettes ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage and Handling (domestic)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

1987 Membership fee \$35 (receive all 1987 issues of VictorTalk). \$ \_\_\_\_\_

+3.00 per month for remaining months and remaining issues of VictorTalk if joining after 1/31/87. (Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.)..... \$ \_\_\_\_\_

QUESTIONS (continued from page 22)

with KNEE TOE GRAPHICS. Instructions to access these picture files are on the disk.

From the BAVUG bulletin board: 3-D.ARC, a 3-D Graphics Generator; TGDEMO.ARC, a Turbo Graphics Demonstration.

So much for the brief descriptions. Remember, read the various DISKIDnn.TXT's for additional information.

\* \* \*

More on Warning Tones

R. W. Johnson provides the following comment on "Warning Tones" by Joe Zmuda (VictorTalk, Vol. 4, No. 6):

On my Victor 9000 HD, in MS-DOS 1.25, ^G produces nothing, and ECHO ^G produces "Command not found" error.

The following program was written in MSBASIC:

```
100 FOR I=1 TO 4
110 PRINT CHR$(7);
120 FOR J=1 TO 75:NEXT J
130 NEXT I
140 SYSTEM
```

This program was saved in ASCII as "BEEP",A; it was then compiled into an executable file (.EXE using BASCOM and LINK. It was renamed as ^G.EXE (not ALT G, MSDOS wouldn't take that, but using the ALT 7 key followed by the G key). It still won't beep with true G but it will with ALT 7 G.

The nested J loop was necessary to provide some delay for the tone to complete and thus give a longer beep. If the time is increased it will produce four distinct beeps. Without line 120 it gives only one short beep, no matter what is done with the I loop.

\* \* \*

VictorTalk

For all inquiries, write to P. O. Box 50893  
Palo Alto, CA 94303

Telephone: (415) 820-5622

Editor: Jack Prichard

Contributors: J. Haste, Franz Hirner, R. W. Johnson, John Knobelsdorf, Mary Jo Knobelsdorf, Jack Prichard.

Subscriptions: Nancy Weaver

VictorTalk is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in VictorTalk. Copyright 1987 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

VictorTalk is an independent publication not affiliated in any way with Victor Technologies, Inc. Victor is a registered trademark of Victor Technologies, Inc.

Submit Articles on MSDOS disks in ASCII or Spellbinder format, with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

Yes, I want to join.

Enclosed is a check/money order for \$35.00 (foreign membership, \$40 in U.S. dollars) for my 1987 Victor\*Group membership.

Name \_\_\_\_\_  
Title \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_

Make checks payable to:

VictorTalk

P. O. Box 50893  
Palo Alto, CA 94303

City \_\_\_\_\_  
State \_\_\_\_\_ ZIP \_\_\_\_\_  
Country \_\_\_\_\_  
Phone \_\_\_\_\_



## BAVUG RBBS / Franz Hirner, Sysop

Several good things have happened on BAVUG during the past month or so. If you're not yet into modems, PC-Pursuit, and other communications things, now is the time to jump in and get your feet wet.

February was a relatively slow month on the board. Perhaps this was the result of the failure of our 60 MB Priam hard disk. The system is back and running on a 40 MB Priam until the 60 MB drive gets back from repairs. Seems there was a problem with the "embedded servo" which requires major surgery. It should be back in another couple of weeks. Meantime, the board is hardly the worse off except a few files are missing from the download areas. [See the FIDO Utilization Report on page 4. - Ed.]

### Microsoft Windows

There has been a lot of conversation on the board this past month regarding Microsoft Windows. Microsoft has been pushing hard to make their Windows program the defacto graphics interface for the 16 bit industry (and even 32 bit now). There are many new graphics chips (from Intel, Texas Instrument, and Hitachi just to name a few) that have been in the hardware people's hands for about six months now to develop new cards for the other machine. It seems that the other machine will soon have graphics cards available that will allow screen resolution matching the Victor 9000 or even a bit better in some cases. All this and at a "reasonable" price to boot. The only problem is all the screen formats are different. This presents a tremendous problem to the software developers. If Microsoft has their way, Windows will come to the rescue.

Windows will present a uniform graphics interface between the hardware and the program. Hardware suppliers will supply a driver for their particular product to tie it into Windows. The software folks will write one program based on the Windows screen protocol, and IBM's compatibility problems will be solved. According to the San Francisco Chronicle, IBM announced last week that MS Windows will be their standard Graphics Interface.

What do we care about the way IBM's problems are solved? Why do we care about new software like Page Maker that's not being written for the Victor anyway? Simple. If Microsoft has their way and establishes a standard, almost all new software that is graphics oriented (and more will be graphics oriented if there is a standard) could be run on the Victor 9000 if the necessary drivers were written. Imagine, new software releases for years to come would run on the V9.

First of all, what Windows will NOT do: It won't make standard IBM software run on the Victor! It's also not the best multitasker, since it's not preemptive (like DESQview).

That said, there's much that it will do!: Any Windows application (and there are a few pretty good ones) would run on the Victor. You could do multitasking on the Victor (even if it's not as good as DESQview, it still works). You would have the benefits of windowing for well-behaved applications. Plus, MS clearly intends Windows to be its interface for the future.

Just think — Aldus Pagemaker (a Windows application) running on the Victor with a its superb bit-mapped display! There are also some excellent graphic programs that would also be able to run on the Victor. Basically it would open up a significant new world of advanced, quality software. And Victor's owners could even write their own Windows applications with fancy windowing — that would also run on PCs or other Windows machines. Like generic MS-DOS except far more sophisticated, able to take advantage of the superior Victor display.

As for performance — especially in single-program mode — Windows is pretty darn good for such a powerful device-independent graphics package. Yes, it would be a bit slow, but only because of the crunching required to update the graphic display — since all the output is bit-mapped (although there is a rumor of a text-mode version also). Still, I'd think it would be well worthwhile — a real way to take advantage of your graphics and get access to some powerful software.

Windows shines in that it has a bit-mapped display interface much like the Macintosh, with all of the advantages. It's really a natural for graphic software and WYSIWYG displays. It's also designed for cooperative multitasking and true device independence — so it's a really powerful environment for programs designed to use its facilities!

GEW (from DRI) has some of the graphic attributes of Windows — but it lacks the multitasking program support. And Windows has the advantage that Microsoft is behind it 100% as its machine-independent interface for the future. After quite a delay, it looks like Microsoft is finally getting some key developers to support Windows (like Aldus). And don't forget that Gates has promised that Excel (the Mac spreadsheet that it probably better than any other on ANY machine) will be released in a Windows version RSN.

Windows could keep the Victor alive for a long time to come even with only a few key applications!

This sounds too good to be true. What are the drawbacks? Speed for one. The new 32 bit machines are running at 16 MHz. and higher. That's fast enough to not only do everything in graphics mode, but support multitasking at the same time. We wondered just how windows might perform on the Victor 9000. The closest machine available is an IBM PC with the Hercules Graphics card (720,384 resolution). There are a few less pixels to draw in, but the IBM is almost 10% slower than the V9 (4.7 MHz vs. 5.0 MHz) so the comparison is about a push. Single applications running in Windows are tolerable speedwise if the IBM/Hercules measurements are any indication. Multitasking is bearable, but really would only be useful if you were to start two programs running their tasks and then go to lunch expecting them to be finished when you returned. The extra overhead of the Windows interface would most likely be tolerable in view of the new software that would be made available, especially when you consider that Windows would be run only for Windows software by most users.

What does it take to develop Windows for the V9? Microsoft has a fairly well documented development kit for the drivers with lots of sample code for existing hardware. Drivers would have to be written for the screen, keyboard and the mouse serial port if the mouse was implemented. This is no small task, but it is doable if there is enough interest and two or three people want to help with the code.

Are you interested in Windows? If so what are the drivers worth to you? Should the group get involved? If you have any feelings pro or con regarding Windows please leave a message on the board or write me c/o Victor Talk.

### Victor Disk Performance

There are several programs available in the other world that measure a computer's disk performance. Victor has not had such a program available that I know of. As a result, there has been no easy answer to the question "Whose drive is faster?"

Well, now DSKBENCH is available to measure your disk's speed. [While the source code was provided for publication, space was not available. - Ed.] For those of you who don't know a compiler from a whatsit, DSKBENCH can be downloaded from BAVUG.

The results of several tests on various machines that I have access to are presented for your evaluation. I did not pull the covers off the computer in order to get the exact model of drive but the results are still interesting. If you will time your own drive and provide the information to either BAVUG via modem or to VictorTalk we'll try to make a tabulation for

every one's information.

Comp./ Drive Combo.	Block I/O		Sector I/O		DOS Random	
	Read (kb/s)	Write (kb/s)	Read (kb/s)	Write (kb/s)	Sector Read average (ms)	Eff. Rot. Speed (rpm)
1	20.9	8.4	2.2	0.8	470.0	252
2	72.1	81.9	13.1	9.0	86.7	1800
3	76.7	76.7	13.1	9.2	117.1	1800
4	76.7	75.1	13.4	9.4	53.3	1800
5	72.1	80.1	13.4	9.2	50.0	1800
6	270.3	206.0	73.2	63.2	2.3	38250
7	64.9	36.1	22.5	12.3	118.0	3602

### Computer/Drive Combinations:

- 1 Victor 9000 w/Tandon 100-4 1.2 MB Floppy Drive
- 2 Victor 9000 w/Rodime 30 MB Hard Drive
- 3 Victor 9000 w/Tandon 10 MB Hard Drive
- 4 Victor 9000 w/Atasi 40 MB Hard Drive
- 5 Victor 9000 w/Priam VT-150 40 MB Hard Drive
- 6 Victor 9000 w/White Crane RANDISK (500K)
- 7 IBM PC w/Miniscribe 20 MB Hard Drive

### PKARC

PKARC is a combination file compression and library program used by many RBBS callers. In fact, most of the programs on BAVUG available have been ARC'ed. ARC'ing allows us to both include all files necessary for a given program or application to run and to compress those files into a smaller space to save storage capacity on our disks and to reduce your online time to download. You will have to unARC the file after you download it though before you run it. The original ARC utility was named "ARC". It's still used by many boards. PKARC is faster than ARC by far and we will be using it to ARC programs in the future. Next time you're on the board be sure to download a copy of PKARC.

If you upload files to RBBS systems that don't use PKARC you'll want to be careful not to Squash files or they won't be compatible with the older ARC which most people still use. Otherwise, users of the older and slower ARC won't be able to unARC your files.

You can modify PKARC to make not Squashing the default with debug. Just follow the example below. After the modification you will HAVE TO USE THE SWITCH to enable Squashing.

Here's how to use DEBUG to reverse the logic of the "/oc" switch in PKARC >>v2.0 only<< to disable "squashing" by default:

```

debug pkarc.com
-u f20 f25
280D:0F20 833E6CEB00  OMP  Word Ptr [EB6C],+00
280D:0F25 7529      JNZ  0F50
-a f25
280D:0F25          JZ   f50
280D:0F27
-u f20 f25
280D:0F20 833E6CEB00  OMP  Word Ptr [EB6C],+00
280D:0F25 7429      JZ   0F50
-w
Writing 4430 bytes
-q

```

With this change, you'll only get "squashing" if you use the switch. By default it will be 100% compatible with ARC v 5.2.

### "C" Programming

BAVUG has been digging into Microsoft "C" lately. MSC is the only language that will allow the programmer complete access to the Victor 9000 and full use of the powers of both DOS 2.1 and DOS 3.1 (What, you're still running DOS 1.25? Come on into the modern world!). I wanted to move Wall Street Weekly into a new environment capable of downloading price data on the stocks I follow directly from the Dow Jones mainframes. "C" appears to be the best tool to use for this major revision in the program. "C" has the ability to get almost as close to the hardware as assembler or to program at a much higher level similar to Basic or Pascal. It's an ideal tool for the task at hand.

The first chore was to develop a graphics interface in "C", which is just about done. Next come communication routines (anyone have any sample code?) and finally the interface to the Dow Jones mainframe will have to be written. Slowly everything is falling into place.

The graphics routines will be released into the public domain as soon as a manual can be written. If you'd like to see a small sample of the power of MSC, download the file V9\_DEMO.ARC from BAVUG. The program does not do anything except display some of the capabilities of the Victor 9000/MSX package.

For those of you new to "C" (like me) here are a few tips that have been passed along by John Navas:

I try to write code that is easy for an >>experienced C programmer<< to understand. For example:

I often use macros to improve code readability, even when they are complex. Once the macro is carefully debugged, it can be used to produce source code that is clear and easy to follow — certainly easier to follow than long sequences of obscure statements.

I try to keep all functions logically consistent and short, choosing arguments and return values with care. I practice data abstraction, hiding messy internal details from outside code. I always try to use local rather than global variables. I hone my functions so I can use them over and over again.

I take the time to understand what kind of machine code will be generated for typical statements so I don't work against the compiler — but I also try to get the compiler to do as much of the work for me as possible!

I avoid subscripting \*except\* where it makes sense, because pointers are usually much more efficient.

I try to use the power of the language, and in C that means structures, unions, enumerations, casts, prototype declarations, etc., not just simple statements.

I try to make my programs as portable as possible, following the AT&T guidelines in "The C Programmer's Handbook."

I try to learn something new from each program I write! (You might also check out the "C rules" that I recently uploaded.)

Thanks John for the good advice. The "C" rules file that John mentions are available for download from BAVUG. They may be printed here in the next issue. As usual, the RBBS is at least six weeks ahead of the printed media.

There's much more going on on BAVUG. Stop by and see (and contribute).

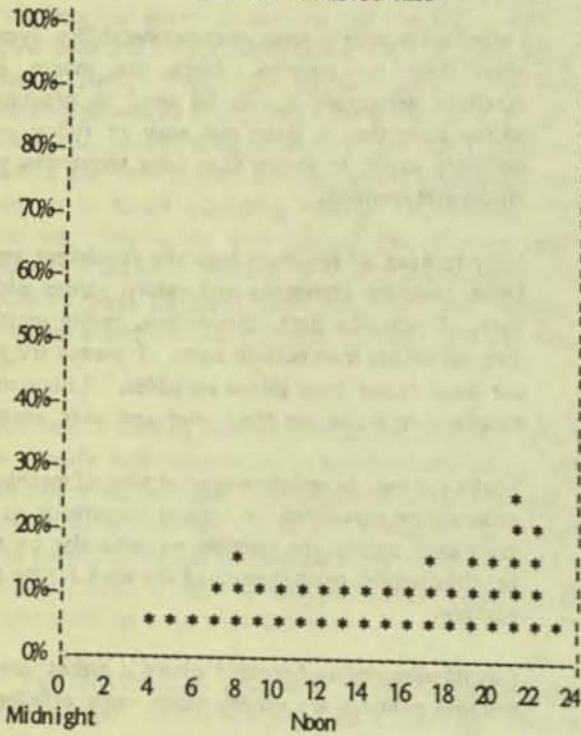
Any comments can be directed to me at BAVUG or through Victor Talk.

FIDO UTILIZATION REPORT FOR  
BAUG - Node 125/625

Files Downloaded

(Files not starting with C: have A:\FIDO\ omitted)

From 31 Jan 87 to 28 Feb 87  
Percentages Averaged over this 27 day period  
Report Printed on: 2/28/1987 at 14:11  
Data From: A:SYSCP.LOG



File Name	Frequency	File Name	Frequency
APPLIC\DOGT01A.ARC	1	UPLOAD\C-MULTSK.ARC	1
APPLIC\NEWEMIBM.ARC	2	UPLOAD\EMIBMWP.ARC	2
APPLIC\VKERM28.ARC	1	UPLOAD\FEDTAX86.ARC	2
APPLIC\XMSCARD.ARC	1	UPLOAD\PC-TAX86.ARC	1
APPLIC\XMODEM.LBR	1	UPLOAD\RLL INFO.ARC	1
DBASE\DB24.PAT	1	UPLOAD\WAITE.ARC	1
DBASE\DB3PLUS.PAT	1	UPLOAD\WATERMSR.ARC	1
DBASE\DB3PNEW.PAT	1	UPLOAD\XMODEM.ARC	1
DBASE\DBA1186.ARC	1	UTILITY\ARCS12.DOC	4
DBASE\DB ADV.ARC	1	UTILITY\ARCS12.EXE	2
GENERAL\CRITICAL.BAS	1	UTILITY\BACOPY.ARC	1
GENERAL\CUBE.ARC	1	UTILITY\CACHE2.ARC	2
GENERAL\FREBWILL.ARC	1	UTILITY\COVER.ARC	1
GENERAL\MUZAK23.ARC	1	UTILITY\DISK.ARC	2
GENERAL\SETRAM.ARC	3	UTILITY\LUS7.EXE	1
GENERAL\SKULL.ARC	1	UTILITY\MKSET.ARC	1
GENERAL\SPACEWAR.ARC	1	UTILITY\PROG4A20.COM	1
GENERAL\XMSCARD.ARC	1	UTILITY\STRIP.EXE	1
PATCHES\CBS-CVT.ARC	1	UTILITY\TYPER.ARC	1
PATCHES\DB3PNEW.ARC	1	UTILITY\UNDELETE.COM	3
PATCHES\DRIVER.ARC	1	UTILITY\USQ2.EXE	1
PATCHES\MGRAPH.ARC	2	VICTECH\CSK-V1N2.TXT	1
PATCHES\PC.ARC	1	VICTECH\CSK-V2N3.TXT	1
PATCHES\TPTUTOR1.ARC	1	VICTECH\CSK-V2N4.TXT	1
PATCHES\TPTUTOR2.ARC	1	VICTECH\DEBUG.ARC	3
PATCHES\XMODEM.ARC	1	VICTECH\LINE25.ARC	1
PICKUP\ALLFIL.ARC	1	VICTECH\MEMMAP.5B	1
PICKUP\CIRCLE.C	2	VICTECH\RS232TXT.ARC	1
PICKUP\CRULES.ARC	18	VICTECH\SYSINT.ASM	1
PICKUP\FP.C	1	C:\C\BALD.C	1
PICKUP\FPFJH.ARC	2	C:\C\BANK.LBR	1
PICKUP\GRAF10.ARC	1	C:\C\C-MULTSK.ARC	1
PICKUP\GRAF11.ARC	2	C:\C\CFLOW*.*	1
PICKUP\GRAF11A.ARC	1	C:\C\CPRETTY.C	1
PICKUP\GRAF12.ARC	1	C:\C\CTUTOR.ARC	1
PICKUP\GRAFMAKE.ARC	1	C:\C\DIAL.C	1
PICKUP\LATER.ARC	1	C:\C\GCIRCLE.ARC	1
PICKUP\MGS.ARC	1	C:\C\ICBM.C	2
PICKUP\NAVAS.ARC	1	C:\C\MSCTOOLS.ARC	1
PICKUP\NAVAS1.ARC	1	C:\C\SCCS.ARC	1
PICKUP\NAVAS2.DOC	1	C:\C\STAYRES*.*	1
PICKUP\NAVAS3.ARC	2	C:\C\STAYRESC.ARC	1
PICKUP\NAVAS4.ARC	1	C:\C\TKD.ARC	1
PICKUP\NEWPIXS.ARC	3	C:\C\USTIA.ARC	1
PICKUP\SHADOW.ARC	1	C:\C\YACC.ARC	1
UPLOAD\ALLFILES.DOC	6	C:\NEWS\ATTRAG21.ARC	1
UPLOAD\ANSIVID.C	3	C:\NEWS\FNEWS403.ARC	1
UPLOAD\ASMG2.ARC	2	C:\NEWS\FNEWS404.ARC	1

Calling Baud Rates

300 Baud - 76 ( 27%)  
1200 Baud - 170 ( 61%)  
2400 Baud - 31 ( 11%)

Utilization Statistics

Total Up-Time- 38880 Min  
In-Use-Time- 3745 Min  
FIDNET (est)- 1620 Min  
Avg Utilization- 10.1%

Total Calls This Period - 277 (about 10 calls per day)  
Incomplete Logins - 8 (about 0 per day)  
Average Call Duration - 13.5 (+/- 15.0) min.  
FidNet active from 1:00am to 2:00am.

File Upload/Download Activity

Transfer Type	Total this period	Avg/Day
Uploads	21	0.8
Downloads	144	5.3

SYSREPT, Version 1.52, April 28, 1986  
Copyright 1985 by Alfred Anderson, FIDO 14/61  
This software may be distributed, without cost, among FIDO user groups. [Format changes have been made to this report so as to fit the VictorTalk page format. - Ed.]

# SPELLBINDER/dBASE II DATA EXCHANGE / Lexisoft

The dBASE II Data Base Management System requires that the data be enclosed in quotes (" or '), that fields be delimited by commas (,) and that records be separated by carriage returns. The following commands will prepare your SPELL-BINDER data for entry into dBASE II.

## INSIDE SPELLBINDER:

1. Read into your workspace the SPELLBINDER data file.
2. COMMAND: **500f9/S/^/[RET]**

This marks the end of each record. Later we will replace the mark with a carriage return. The number **500** tells SPELLBINDER how many times to carry out this command. Increase or decrease this number to correspond with the number of records in your file. The number **9** indicates the number of lines per record minus one. If you have 15 lines per record then substitute the number 14.

3. COMMAND: **T/SA/</^,"/[RET]**

This places a quotation mark (") at the beginning and end of each field and removes the carriage returns that separate the fields and replaces them with commas. Fields with no data will simply have two quotation marks with nothing between them. This works as a place holder to the dBASE II system.

4. COMMAND: **T/S//^/[RET]**

This inserts a quotation mark at the very beginning of the first field.

5. COMMAND: **T/SA/^/"<"/[RET]**

This replaces the marks with a carriage return so that each record is separated from the next by a carriage return.

6. COMMAND: **GD[RET]**

This saves your file on disk.

## INSIDE dBASEII:

Using the dBASE II Command **CREATE** you can define a form that looks just like the format used in SPELLBINDER's example file "cuslist.dem". You will be asked to define the type of data that will be inserted into each field. We defined each field to be "C" for Characters or "N" for Numbers. Our form "CUSTOMER" looked like this:

FLD	NAME	TYPE	WIDTH
001	CUESORT	C	50
002	NAME	C	50
003	COMPANY	C	50
004	ADDRESS	C	50
005	CITY	C	50
006	STATE	C	50
007	ZIP	N	15
008	FIRSTNAME	C	50
009	PHONE	N	20
010	INFO	C	50

The dBASE II Command **APPEND** is the method for translating straight ASCII data into database data. All SPELLBINDER files are straight ASCII files.

USE CUSTOMER

APPEND FROM <spellbinder data file> DELIMITED

The USE Command tells dBASE II which file to put the data in. In the example above we told dBASEII to put the incoming data in the "CUSTOMER" file. The APPEND Command transfers the data that you prepared in SPELLBINDER into dBASE II data. DELIMITED advises dBASE II that the data is in the comma delimited format.

To convert dBASE II data files into SPELLBINDER files use the dBASE II COPY facility. This will convert database data into straight ASCII files.

USE CUSTOMER

COPY TO <filename> DELIMITED

## INSIDE SPELLBINDER:

To read this file in SPELLBINDER add the .TXT extension to the filename. Carriage returns should only appear at the end of each record. If there are carriage returns within the records remove them using the following instructions:

1. COMMAND: **T/SA/.A/^A/[RET]**

The first field of our record is the cuesort line. This will put a mark at the beginning of every record by searching for the first two characters of the cuesort line.

2. COMMAND: **T/SA///[RET]**

This searches and removes the quotes from around the DELIMITED dBASE II data.

3. COMMAND: **T/SA/<///[RET]**

This removes all the carriage returns.

4. COMMAND: **T/SA/^/</[RET]**

This replaces the marks with carriage returns. Now each field is delimited by a comma (,) and each record is separated by a carriage return.

5. COMMAND: **T/DI[RET]**

This removes the leading carriage return. This is important because it would destroy the line count necessary for running the mail merge cuesort.

[Editor's Note: This article was provided by Lexisoft's Venita Kent-Basham, Assistant Manager - Technical Support.]

\* \* \*

# MS-BASIC, Part 2 / Jack Prichard

The first article in this series, "MS-BASIC, Part 1", (VictorTalk Vol. 5, No 1, Jan-Feb 1987) showed how to run and examine existing MSBASIC programs and how to create simple programs. The examples given printed the various fonts available on a dot-matrix printer (specifically the Epson MX-80). This second article will extend these concepts to show how to create a program to produce a selected number of forms in an arbitrarily specified format.

The format specified in this example comes from a real application. The owner of a small photographic business had the following problem:

1. He was running out of his preprinted billing statements.
2. He had purchased a sizable stock of window envelopes for use with these statements.
3. He was contemplating moving, so did not want to invest in a quantity of professionally preprinted forms with his then current address.
4. He would soon be using a partially computerized accounting system (written in MSBASIC, as an exercise by the author).

Thus, it was decided to write a program which would duplicate his existing billing form as closely as practical, since his existing preprinted forms had filled his needs quite satisfactorily. The program would allow both for use with the computerized billing and for interim use with manual billing. The example used here is taken directly from that design (with a few changes "to protect the innocent ... and the guilty").

In programming a form there are two extremes of procedure that might be used. The first corresponds to freehand sketching where the design is created largely "by eye" on the screen. The second corresponds to mechanical drawing or drafting where the lines are carefully measured and the dimensions recorded. The first procedure is more applicable to designs such as the FIDO sketch used as the legend by BAVUG RBBS (Bay Area Victor Users Group Remote Bulletin Board System) (See VictorTalk, Vol. 4, No. 4, 1986.) The second is possibly more suitable for the present problem. However, both procedures will be illustrated here.

The statement form that was developed is shown in Figure 1. It closely resembles the original preprinted form. In discussing the programming to produce this form, we will first describe the freehand sketching style which is illustrated in Figure 2. We will then modify the program to illustrate the more mechanical or symbolic form of programming which is illustrated in Figure 3. The programs and comments are intended to fill two needs: the specific one of showing how to

write a program that will produce the desired output (statement form) and the more general one of illustrating some useful practices to provide readable and maintainable programs.

## GENERAL PROCEDURE TO BE FOLLOWED

1. Unless you really intend to "wing it", add a grid to the form to be duplicated (whether a preprinted form, a computer printout, or a hand-drawn sketch). This is intended to aid in the location of lines and columns. The grid can be in the form of a penciled-in grid, or a transparent overlay, etc. For normal computer printers use the standard 6 lines per inch and 10 characters per inch. However, these can be readily changed to meet special requirements and the capabilities of your printer.

2. Review the form to determine the most appropriate character fonts to be used for each item. In the example form it was decided that all items would be in normal font except the following would be as indicated:

"Statement"	Expanded (or large) characters
Name (in heading)	Emphasized characters
Telephone number	" "
Instructions	Condensed "
Closing items	Emphasized "

3. Write MSBASIC statements (similar to those of Part 1) to define strings for turning on and off each font to be used.

4. Write MSBASIC commands for each line of the statement form to print the desired format.

## FREEHAND SKETCHING OR FACIMILE STYLE

Following are comments on the program of Figure 2, on a line-by-line basis.

Line 10 gives the program name, date of last modification, and some idea of what it does.

Lines 20-80 define strings used to turn on and off the various character fonts to be used. As with all variable names, mnemonics are used to facilitate understanding their meaning. With printers other than the Epson MX-80 these definitions may have to be changed to fit the particular printer to be used.

Line 90 defines a constant for later use in the print subprogram. Defining it here in the heading or early part of the program makes it easy to find should change be necessary

**STATEMENT**

**RON WOODWARD PHOTOGRAPHY**  
 2716 Davis Street  
 RICHMOND, CALIFORNIA 94800

(415) 726-0465

Please detach and return with your remittance

\$

DATE	CHARGES AND CREDITS	BALANCE
	BALANCE FORWARD >>>	

Pay last amount in this column^^  
**THANK YOU**

**RON WOODWARD PHOTOGRAPHY**

Figure 1. STATEMENT FORM

and keeps it out of the repetitive loop of the subprogram.

Line 300 provides for keyboard input of the number of forms to be printed. Note the use of the comma to suppress the question mark that would otherwise be displayed on the screen.

Lines 310-350 provide instructions for paper alignment in the Epson MX-80 printer so that the line printed by line 780 falls exactly on the perforation. (The purpose of this is so that the top portion of the statement may be detached by the customer, along the perforations, for mailing with the payment.) Again, with other printers some change in alignment procedure may be required. Note, that to facilitate entering the word "PRINT", the cursor is positioned immediately below the "P" in "PRINT". Also note that until the word "PRINT" is correctly entered, the instructions are repeated upon each entry of a carriage return.

Lines 360-380 constitute the main program, they simply call the print subprogram the requisite number of times to make the specified number of copies. Note that line 370 (the portion of the loop between FOR and NEXT) is indented for clarity, a good practice to follow.

Line 390 terminates program execution. Although END is not actually necessary in many programs, in this program it is necessary. Without the END statement, execution would continue with another (extra) copy being printed and execution would then terminate with an error message, "Return without GOSUB".

Line 600 is a remark line (with the lead apostrophe substituting for REM) to mark and name the subroutine that follows. A round line number was chosen to make reference simple and to leave a number of line numbers unused for possible later additions.

Line 610, like the majority of other lines in the subprogram, is simply a LPRINT command with a facimile of the line enclosed in double quotes. Note that the first double quote is aligned with those for all succeeding lines to further the facimile appearance of the screen and the program listing. With the normal screen width of 80 characters, the characters beyond the 80th would wrap around to the next line, somewhat modifying the desired facimile appearance of the screen. In principal, this could be avoided by the use of the Victor 9000's 132-column display. However, the small size of the characters makes the use of this feature of questionable value. Using a printer that allows close character spacing, the desired facimile appearance can be obtained, as in Figure 2.

Line 620 provides three blank lines in a one-line statement. While it does reduce the program line count, it also somewhat distorts the desired facimile appearance by forshortening the

facimile vertically and intruding into the facimile space to the right of the vertical line-up of double quotes. The desired end result could also have been obtained by a FOR...NEXT statement similar to lines 820-840. Alternatively, a one-line FOR...NEXT statement could have been used:

```
620 FOR N=1 TO 3: LPRINT: NEXT N
```

Line 630 prints "STATEMENT" in expanded or large characters. Note that the blank spaces are of the same width as the characters (double normal width) so that the facimile of the listing again fails to exactly duplicate the final printed statement format. Also note that the use of LC.OFF\$ is actually unnecessary in that the large character mode would be turned off by the line feed character.

Lines 710-740 were entered in this simple format for facimile reasons and for correspondence with Figure 3, wherein provision has been made for inserting name and address data in the window space.

Line 770 (like line 630) distorts the facimile appearance because neither the blank spaces nor the characters are of the same width as in the actual statement printout.

Lines 820-840 are the more customary way of providing multiple identical lines using the FOR...NEXT command. This, too, distorts the facimile appearance of the listing, as the 3-line listing represents 22 lines in the actual printout.

Line 860 (like lines 770 and 630) also distorts the facimile appearance because of the different character widths in the actual printout.

Line 900 is the form-feed character to position the printer paper for the start of the next statement form. Note that it is followed by a semicolon. This is to avoid getting an extra, unwanted line feed, as would otherwise occur.

Line 910 concludes the print subprogram and returns control to the program line following the GOSUB command (line 180, in this case).

It is seen that the facimile format used in writing this program is of some assistance, even though wrap-around occurs in some lines of the screen display.

## ENTERING/TESTING/EDITING - FACIMILE STYLE

Try entering this program into your computer (with necessary changes to accommodate your particular printer, and with such other changes as you may want to try). When done, SAVE the program as STMFRM.BAS (StAtement FoRM), in ASCII, as was explained in Part 1 of this series. (Hopefully, you have previously SAVED your partial program as work



10 REM STIMFRM.BAS 3/14/87 PRINTS BLANK STATEMENT FORMS IN NUMBER SPECIFIED BY KEYBOARD ENTRY

20 REM FOLLOWING CODES FOR EPSON MX-80 PRINTER:

30 LC.ON\$= CHR\$(14) 'LARGE CHARACTERS ON

40 LC.OFF\$= CHR\$(20) ' " " OFF

50 EC.ON\$= CHR\$(27)+CHR\$(69) 'EMPHASIZED " ON

60 EC.OFF\$= CHR\$(27)+CHR\$(70) ' " " OFF

70 CC.ON\$= CHR\$(15) 'CONDENSED " ON

80 CC.OFF\$= CHR\$(18) ' " " OFF

90 NEL=22 'NO. BLANK LINES

300 INPUT "NUMBER OF FORMS TO BE PRINTED=" , NMAX

310 PRINT "ALIGN PRINTER PAGE WITH CENTER OF 9TH HOLE DOWN FROM PERFORATION"

320 PRINT "AT MATCH-MARK."

330 PRINT " \* \* \* \* \* TYPE PRINT TO PROCEED \* \* \* \* \*"

340 INPUT " " , A\$

350 IF A\$ <> "PRINT" THEN 110

360 FOR NF=1 TO NMAX

370 GOSUB 600

380 NEXT NF

390 END

600 \*\*\*\*\* SUBPROGRAM - PRINT STATEMENT \*\*\*\*\*

610 LPRINT "-----"

620 LPRINT: LPRINT: LPRINT

630 LPRINT LC.ON\$ " STATEMENT" LC.OFF\$

640 LPRINT EC.ON\$ " RON WOODWARD PHOTOGRAPHY"

650 LPRINT EC.OFF\$ " 2716 Davis Street"

660 LPRINT " RICHMOND, CALIFORNIA 94800 "

670 LPRINT: LPRINT

680 LPRINT EC.ON\$ " (415) 726-0465"

690 LPRINT EC.OFF\$ " . "

700 LPRINT

710 LPRINT

720 LPRINT

730 LPRINT

740 LPRINT " . "

750 LPRINT: LPRINT

760 LPRINT " \$ \_\_\_\_\_ "

770 LPRINT CC.ON\$ " Please detach and return with your remittance"

780 LPRINT CC.OFF\$ " |-----"

790 LPRINT " DATE | CHARGES AND CREDITS | BALANCE"

800 LPRINT " | BALANCEFORWARD>>>|"

810 LPRINT " |-----"

820 FOR N=1 TO NEL

830 LPRINT " |-----"

840 NEXT N

850 LPRINT " |-----"

860 LPRINT CC.ON\$; " Pay last amount

in this column^^^"

870 LPRINT CC.OFF\$; " " EC.ON\$ "THANK YOU"

880 LPRINT " RON WOODWARD PHOTOGRAPHY"

890 LPRINT EC.OFF\$ " |-----"

900 LPRINT CHR\$(12);

910 RETURN

Figure 2. PROGRAM TO PRINT STATEMENT FORM - Facimile Style

proceeded to limit your risk of loss in the event of power failure, or whatever.)

Now try running the program. If it runs properly you've done a terrific job! More likely than not it may hang up and present you with an error message. If the meaning of the error message is not obvious, look up the meaning in the Victor documentation. (In '82 documentation, Appendix A, "Summary of Error Codes and Error Messages"; similarly, Appendix G in '83 documentation.) The more likely errors are "syntax errors" due to misspelling of commands and "type mismatch" due to omission of \$ from a string variable name.

Edit the program as necessary to get it to run properly and produce the desired printout. Remember to frequently SAVE the program as you proceed, to minimize potential loss of your fixes.

## MECHANICAL DRAWING OR SYMBOLIC STYLE

Thus far, no new commands, statements or functions have been introduced beyond those used in Part 1 of this series. Now we have need for a new command, LPRINT USING. Before proceeding, look up this command in Table 1 of MSBASIC, Part 1 and read the referenced Victor documentation. Similarly, we will be using several new functions in addition to CHR\$, previously introduced. The new functions are (in order of their use): SPACE\$, STRING\$, SPC. Look up these functions in TABLE 1 - MSBASIC FUNCTIONS BY USAGE CATEGORY. The first two will be found under the string functions category, the last one under miscellaneous functions. Again, read the referenced Victor documentation.

Figure 3 is a listing of the program as modified to illustrate the use of the new functions and to show how variable data may be introduced with the LPRINT USING command. Note that lines 20-90 and 300-390 are identical to those in Figure 2; hence, they will not be commented upon further here. (Lines 300-390 have not been reprinted to save space.)

Comments on the program of Figure 3 by line number follow:

Lines 10-16 provide identification, as before, with some additional notes.

Lines 100-140 define a number of strings using the new functions. Note the use of the plus sign to concatenate the multiple parts of the definitions. These definitions involve counting spaces and characters, rather than creating a facimile of the line. As has been done elsewhere, corresponding parts of the definitions have been aligned vertically to simplify reading and checking for errors. Each

of these definitions are used more than once. Those definitions that are used only once are defined within the subprogram portion. In line 130 note the use of the back slash to denote the beginning and end of the space reserved for the variable data.

Lines 150-180 are examples of variable data to show how they can be introduced into the statement for computer billing. (See lines 670, 710-740, 770, and 810.)

Line 670 completes printing the line started by line 660 (which ends in a semicolon to prevent a linefeed at this point). Note how the space to be occupied by the date is specified by the two back slashes which mark the beginning and end of the space reserved for the date.

Line 770 introduces the use of the special characters \$\$ to format the dollar value of the balance.

Line 800 is shown in the same format as was used for the corresponding line in Figure 2, as the symbolic form seemed less simple:

```
800 LPRINT SPC(20);"DATE";SPC(3);" | ";SPC(10);  
"CHARGES AND CREDIT";SPC(7);" | ";SPC(4);  
"BALANCE"
```

LINE 810 concatenates a variety of formatting characters in its string expression. Note the first semicolon separates the list of expressions from the string expression.

## ENTERING/TESTING/EDITING - SYMBOLIC STYLE

As you have already done with the program of Figure 2, enter the program of Figure 3 into your computer. Since many lines are the same in the two programs, it is suggested that you start with the program of Figure 2. Modify this program as necessary by adding the new lines and either editing or overwriting the lines that are different. Note that, while many lines correspond directly, not all do. When you SAVE the revised program, be sure to change the name so as to avoid overwriting your original (working) program. For example, call the revised program STMFRM2.BAS.

When you think you have a program that should run, try it. Again, you will likely find that it does not run properly or give exactly the desired output format. Edit it, as necessary, until it does do exactly what you intended.

Now, with these exercises accomplished, try designing "from scratch" a form that will meet your particular need. This should be possible without the use of any commands, statements, or functions beyond those you have already used. Good luck!!!

\* \* \*

```

10 REM SIMFRM2.BAS 3/15/87 PRINTS BLANK STATEMENT FORM
12 REM ILLUSTRATES ADDITION OF SOME VARIABLE DATA AS EXAMPLE
14 REM As shown here, the program will run, producing one printout
16 REM For complete program, add lines 300-390 of Figure 2
20 REM FOLLOWING CODES FOR EPSON MK-80 PRINTER:
30 LC.ON$= CHR$(14)          'LARGE CHARACTERS ON
40 LC.OFF$= CHR$(20)        ' " " " OFF
50 EC.ON$= CHR$(27)+CHR$(69) 'EMPHASIZED " ON
60 EC.OFF$= CHR$(27)+CHR$(70) ' " " " OFF
70 CC.ON$= CHR$(15)         'CONDENSED " ON
80 CC.OFF$= CHR$(18)       ' " " " OFF
90 NBL=22                   'NO. OF BLANK LINES
100 L1$=STRING$(16,45)+"!"+STRING$(63,45)
110 L2$=SPACE$(16) +"!"+STRING$(63,45)
120 L3$=SPACE$(27) +"! " +SPACE$(36) +"! "
130 L4$=SPACE$(23) +"! \" +SPACE$(36) +"! \"
140 L5$=SPACE$(23) +"!." +SPACE$(36) +"!."
150 DT$="3/9/87"            'EXAMPLE DATA
160 NM1$="NAME": AD$="ADDRESS"
170 CT$="CITY, STATE, ZIPCODE"
180 BF=27.85: BAL=14.75
600 ***** SUBPROGRAM - PRINT STATEMENT *****
610 LPRINT L1$
620 LPRINT: LPRINT: LPRINT
630 LPRINT LC.ON$; SPC(29); "STATEMENT"; LC.OFF$
640 LPRINT EC.ON$; SPC(22); "RON WOODWARD PHOTOGRAPHY"
650 LPRINT EC.OFF$; SPC(26); "2716 Davis STREET"
660 LPRINT SPC(21); "RICHMOND, CALIFORNIA 94800";
670 LPRINT USING SPACE$(15)+" \" \"; DT$
680 LPRINT: LPRINT
690 LPRINT EC.ON$; SPC(27); "(415) 726-0465"
700 LPRINT EC.OFF$; L5$
710 LPRINT USING L4$; NM1$
720 LPRINT USING L4$; NM2$
730 LPRINT USING L4$; AD$
740 LPRINT USING L4$; CT$
750 LPRINT L5$
760 LPRINT: LPRINT
770 LPRINT USING SPACE$(65)+"$$$$###.###"; BAL
780 LPRINT CC.ON$; SPC(51); "Please detach and return with your remittance"
790 LPRINT CC.OFF$; L2$
800 LPRINT "          DATE |          CHARGES AND CREDITS          | BALANCE"
810 LPRINT USING SPACE$(27)+" \"! \"+SPACE$(34)+"! \"! \"$$$$###.###\";SPACE$(18)+"BALANCE FORWARD>>>\"; BF
820 LPRINT SPACE$(27)+" \"! \"+SPACE$(36)+"! \"! \"+STRING$(15,45)
830 FOR N=1 TO NBL: LPRINT L3$: NEXT N
840 LPRINT L2$
850 LPRINT CC.ON$; SPC(92) "Pay last amount in this column^^"
860 LPRINT CC.OFF$; SPC(54); EC.ON$ "THANK YOU"
870 LPRINT SPC(26); "RON WOODWARD PHOTOGRAPHY"
880 LPRINT EC.OFF$; L1$
890 LPRINT CHR$(12);
900 RETURN

```

Figure 3. PROGRAM TO PRINT STATEMENT FORM - Symbolic Style

TABLE 1. MS-BASIC FUNCTIONS BY USAGE CATEGORY  
(See notation list and notes at end of table)

	Page Ref.		
	'82	'83	
<b>MATHEMATICAL FUNCTIONS</b>			
ABS(X)	2	69	Absolute value of X
ATN(X)	3	70	Arctangent of X in radians
CDBL(X)	3	70	Convert X to double-precision number
CINT(X)	4	71	Convert X to integer by rounding
COS(X)	5	71	Cosine of X in radians
CSNG(X)	3	72	Convert X to single-precision number
EXP(X)	7	73	e to the power X
FIX(X)	7	74	Truncated integer part of X
INT(X)	12	77	Largest integer $\leq X$
LOG(X)	14	78	Natural logarithm of X
RND[(X)]	18	81	Random number between 0 and 1
SGN(X)	18	81	1 for $X > 0$ ; 0 for $X = 0$ ; -1 for $X < 0$
SIN(X)	19	82	Sine of X in radians
SQR(X)	20	83	Square root of X; $X \geq 0$
TAN(X)	22	84	Tangent of X in radians
<b>STRING FUNCTIONS</b>			
ASC(X\$)	2	69	Numerical value of ASCII code of 1st character of X\$
CHR\$(I)	4	70	ASCII code of I
CVI(<2-byte string>)	6	72	Convert string to integer
CVS(<4-byte string>)	6	72	Convert string to single-precision number
CVD(<8-byte string>)	6	72	Convert string to double-precision number
DATE\$	-	72	Date as string: mm-dd-yyyy
FRE(X\$)	8	74	Number of free bytes in string space
HEX\$(X)	8	75	Hexadecimal value of rounded value of decimal X
INKEY\$	9	75	One-character string read from keyboard
INPUT\$(X[, [#]Y)	10	76	String of X characters from keyboard or file number Y
INSTR([I,]X\$,Y\$)	11	76	Position of 1st match of Y\$ in string X\$, starting at Ith character
LEFT\$(X\$,I)	12	77	Leftmost I characters of X\$
LEN(X\$)	13	78	Length (number of characters) of X\$
MID\$(X\$,I[,J])	15	79	Middle J characters of X\$ starting with Ith character
MKI\$(<integer expression>)	15	79	Make string of integer
MKS\$(<single-precision expression>)	15	79	Make string of single-precision number
MKD\$(<double-precision expression>)	15	79	Make string of double-precision number
OCT\$(X)	16	80	Octal value of rounded value of decimal X
RIGHT\$(X\$,I)	17	80	Rightmost I characters of X\$
SPACE\$(X)	19	82	String of X spaces
STR\$(X)	21	83	String representation of X
STRING\$(I,J)	21	83	String of length I, all characters have ASCII code J
STRING\$(I,X\$)	21	83	String of length I, all characters are 1st character of X\$
TIME\$	-	85	Time as string: hh:mm:ss
VAL(X\$)	23	85	Numerical value of X\$

TABLE 1. (Continued) MS-BASIC FUNCTIONS BY USAGE CATEGORY  
(See notation list and notes at end of table)

	Page Ref.		
	'82	'83	
<b>MISCELLANEOUS FUNCTIONS</b>			
DATE	-	-	Numerical day of year
EOF(<file number>)	6	73	Returns -1 (true) at end of sequential file
FRE(0)	8	74	Number of bytes in memory not used by MSBASIC
FRE(" ")	8	74	Forces garbage collection, then returns number of free bytes
INP(I)	UG 2-4	9 75	Byte read from port I
LOC(<file number>)	13	78	Record number just read or written in random file
LPOS(X)	14	78	Position of line printer print head in printer buffer
PEEK(I)	16	80	Byte read from memory location I
POS(I)	17	80	Cursor position (I=dummy)
SPC(I)	20	82	Prints I spaces (only with PRINT and LPRINT)
TAB(I)	22	84	Spaces to position I (only with PRINT and LPRINT)
TIME	-	-	Numerical seconds after midnight
USR[<digit>](X)	UG 2-5	23 85	Calls assembly language routine with argument X
VARPTR(<variable name>)	24	86	Address of first byte of <variable name>
VARPTR(&<file number>)	24	86	Address of buffer assigned to <file number>

\*\*\*\*\*

**Notation used in formats:**

- CAPS Capital letters - portion of functions that must be entered exactly as shown.
  - ( ) Parentheses - enclose arguments to functions.
  - [ ] Square brackets - enclosed entry is optional.
  - < > Angle brackets - user entered data.
- All other punctuation must be entered exactly as shown.  
 X and Y Represent any numeric expressions.  
 I and J Represent integer expressions.  
 X\$ and Y\$ Represent string expressions.

**Page references are to following Victor documentation:**

'82 = "User's Guide for MS-BASIC", first Victor printing December, 1982 and "Reference Manual for MS-BASIC". These two parts were issued together in a looseleaf binder identified as "BASIC 86 User's Guide". Most page number references are to Chapter 3 of the Reference Manual. For brevity the 3- portion of these page numbers is omitted; other pages are identified as in the guide; UG stands for User's Guide.

'83 = "MS-BASIC", first Victor printing April, 1983. This manual, issued in a looseleaf binder also identified as "MS-BASIC", contains essentially the same material as the earlier manual but has a single set of consecutively numbered pages, I-VII and 1-122.

Note: Double-precision functions are supported only by the MS-BASIC compiler.

# UPGRADING THE VICTOR 9000 / Roger Habeck

[The author, a "small Victor dealer", composed this letter for one of his customers. While some of the comments are specific to that customer, many are general and applicable to other Victor users. The 2.5-MB RAM disk mentioned is the Western Automation V2 board with expansion module stuffed with 256K DRAMs. He says that they have updated their device drivers to work with MSDOS 3.1 and "they seem to have done a good job". - Ed.]

Dear Vic User:

I decided to write this letter to both organize my thoughts about your computer system and to provide you with a reference for future use. I have divided the letter into two parts, first a discussion of the current V9 situation and second, those areas in which I feel improvements can be made in your existing system. [Improvements are presented in three groups: hardware enhancements, updated software, and a composite enhancement (a very large RAM disk) - Ed.]

## THE CURRENT V9000 SITUATION

**THE VICTOR 9000:** As you know, the V9 has been discontinued as has its follow-on machine, the VI. After an initial shock during which we saw a few less sophisticated users jump out of their V9s, the situation has stabilized and a strong market exists for used machines. Granted, prices have fallen drastically, but that's more a function of the overall decline in computer prices than a decline in V9 values. (A used Victor still sells for nearly the same price as a new IBM clone.)

**SOFTWARE:** There are somewhere between 100,000 and 150,000 V9 users around the world, and since they tend to be more sophisticated than the average, a strong user support network exists. As a result, after a low point a year or so ago, we are seeing an increase in supported and updated software for the V9. Currently several spreadsheets, a number of word processors, two or three integrated packages and a reasonable number of accounting and specialized programs are available, as well as two good emulator programs that allow the V9 to run much IBM-format software. Most general users such as yourself do not need to worry about software availability for a while and I doubt that this will ever be a major concern.

**REPAIRS:** The situation as regards repairs is also good. Victor Technologies continues to offer depot services to dealers for repairs and exchanges. There are also a number of independent depot level repair facilities (most staffed by former Victor employees). Therefore, I don't see a significant problem in the service area for some time.

**EQUIPMENT UPGRADES:** The situation here is less clear. The 8088 CPU chip in the V9 runs at 4.77 megahertz and handles one byte of data at a time, the newer chips such as the 80286 in the IBM AT can run as fast as 12 megahertz and handle two bytes of data at a time. In other words, the newer chips can be five times faster in certain types of operations. This situation will continue to evolve as faster chips have been introduced and the rate of development seems to be increasing.

There has been some discussion and rumor of an add-on board that would support one of the high-speed, high-capacity microprocessor chips. Boards of this type are currently available for the IBM PC and many compatibles. At the present time, I do not know of such a board being available. However, I expect one to be developed as some of the hardware specialists look for untapped market niches.

## HARDWARE ENHANCEMENTS

Following, I have listed a number of inexpensive steps that can be taken to improve the overall performance of your installation and some general recommendations for procedural changes.

**BOOT ROM UPGRADES:** There have been a number of changes made in the built-in start-up program of the V9. This program is contained in the boot (start-up) ROM (read only memory) chip(s) on the main motherboard. The latest version provides a number of improvements and I recommend that you bring all of your machines up to this standard. You can determine the number of machines that need to be updated by noting their memory display on start-up. The new boot ROMs display memory in standard numbers, while the old ROMs display memory in hexadecimal notation. For example, a machine with 640K of RAM would display A000 on boot-up with the old ROMs and 640K with the new ones.

**CPU UPGRADE:** I have been experimenting with a new CPU chip for some time in several of my machines, and feel that I can make the recommendation to change without reservation. I can remove the 8088 CPUs from your machines and replace them with NEC V20s operating at 8 megahertz. This will almost double CPU speed and provide a noticeable improvement in overall performance. In addition to the installation of the chips, the MSDOS utilities FORMAT and DISKCOPY must be patched to operate at the higher CPU speed.

**RAM EXPANSION:** The one single enhancement to your equipment that will return the greatest dividend in improved

performance is to expand the amount of Random Access Memory. The V9 will use up to 896K of RAM. However, many software packages will not use this much, so I suggest that you increase each machine to 640K, the current DOS standard maximum. I am aware that you have upgraded several machines as inexpensive memory boards became available. However, since this is no longer a likely occurrence, I think it's time to go ahead and do the remainder. [A greater expansion of RAM is recommended in the last section. - Ed.]

## UPDATED SOFTWARE

**OPERATING SYSTEMS:** One area of substantial concern to me is your current use of several versions of MSDOS at your site. If the steps are taken to upgrade the boot ROMs and RAM as described above, you will be able to use MSDOS 3.1 on all of your machines and eliminate the very real danger of DOS conflict. Many MSDOS users do not realize that the same command name in different versions of MSDOS does not mean the command is the same. For example, the CHKDSK command of version 1.25 is destructive to files if run under MSDOS 3.1. The damage caused by using utilities from one version of MSDOS with another is highly random and unpredictable, as is running conflicting versions of COMMAND.COM. It has been my experience that when there are several versions of MSDOS in use at one site the corruption of files and disks is just a matter of time. It only takes an instant for someone to grab a disk with a needed utility, insert it in a running machine using a different version of MSDOS, and trash everything in sight. By the way, this is not a V9 problem, it can occur with any configuration of equipment when multiple versions of MSDOS are in use.

**APPLICATIONS PROGRAMS:** A frequently overlooked area of performance improvement is the upgrading of applications programs. In your case, you are running a large application using Multiplan. However, the version you are using is many revisions old. The first versions of most applications programs that were available for the V9 were applications that had been upgraded from the old 8-bit, Z80 type machines. Because these machines could only use 64K of RAM, the programs were designed to reside on disk and only load the absolute minimum of the program into RAM to save as much space as possible for applications. In the early days of the V9, IBM PC, et al, the situation was not much different. RAM was expensive; V9s were sold with as little as 128K (and 64K in the case of the IBM PC).

As the cost of RAM declined, the minimum RAM configuration considered acceptable gradually increased, until today, hardly a machine is sold with less than 640K on the motherboard. Software companies were slow to modify their software to take advantage of this increased RAM capacity because they had a large installed base of older machines and

users who had not kept the capacity of their machines up to date.

Currently, most up-to-date software packages are able to check the amount of RAM available and load as much of the application and data files as will fit. This means that the current version of Multiplan, version 2.0, will use all RAM available. If your machine has 640K of RAM it would load your whole Multiplan program and the data file into RAM at one time. This would reduce the time required to run your application by 50% or more. The same comments apply to your very early version of Benchmark.

I have noticed that many users assume that because they aren't having a problem with a particular piece of software there is no reason to update it. While in some cases this may be correct, the software companies usually charge a very reasonable fee for updates. I would at least check to determine if an update would improve performance.

**CHANGING APPLICATION PROGRAMS:** One often overlooked possibility of improvement is to change the application program entirely. Many software programs that enjoyed wide acceptance some years ago have fallen by the wayside. The software company may have chosen a poor direction or simply made a series of poor decisions as to how the product would be used.

A good example is Wordstar word processing by Micropro; at one time Wordstar ruled the roost. But, through a combination of arrogance and attempting to dictate to the market place, they have lost much of their following. Those who still use Wordstar seem to fall into two groups. There are those, like myself, who just don't want to change because they don't want to take the time to learn a new program. And there are those who feel they can't change because they have large files of existing data that might be lost in the conversion.

With very few exceptions, most datafiles from one software program can be transferred to another similar program. I don't know of any word processor that can not, with some effort read the files of any other. The same holds true of most spreadsheets, data bases and even many specialized packages such as accounting, inventory and the like. There is seldom a good reason to plug along with hopelessly outdated or limited software just because one is afraid to convert.

Many of the latest versions of the more widely used software packages include utilities to simplify the transportation of data from one software package to another. Even the difficulty of converting from one type of machine and operating system to another is often vastly overstated. With care and planning and some judicious experimentation before jumping in whole hog, converting to a more powerful or friendly

software package can be fairly simple.

#### COMPOSITE HARDWARE/SOFTWARE ENHANCEMENT

I have left this option for last as it is the most technical, requires the expenditure of a significant amount of money and demands some operator retraining and minor programming.

While great attention has been paid to faster and more powerful microprocessors, far less attention has been paid to the portion of the system that takes up most of the time, loading data on and off the disk. The high power CPU chips are desired because of the growth of the size of application being run on micros today. We really don't need more power for word processing, most spreadsheet activity, or most routine day to day type of activity. The powerful chips are needed for the big data bases, large interactive accounting packages and the like.

The problem is, that despite the speed of the chip, input and output to and from the hard disk drive is agonizingly slow, regardless of the speed of the hard disk. These big applications are far too large to load into RAM; in fact, in many cases the program itself will take up most of the available space in RAM. This means that sorts and indexes have to be done in a small amount of workspace and many calls to the disk must be made for data. It's like having a car capable of 200 miles per hour but driving only in town where the longest straight stretch is two blocks.

There is a way out of this dilemma that works for many applications. That is to use a very large RAM disk. A RAM disk is extra RAM installed in the computer which can be configured so the operating system will see it as an additional disk drive. On my personal machine my floppy drive is drive B, my 10-MB hard disk is drive A, and my 2.5-MB RAM disk is drive C. RAM disks have been around for some time, but because of the high cost of RAM were usually limited to small sizes to expedite word processing and the like.

The RAM drive I am recommending is 2.5MB in size. When coupled with 640KB of RAM this provides over 3.5MB of program and data space which operates at electronic, rather than mechanical, speeds. In a particular accounting application, this system (using a fast CPU chip) showed remarkable improvement in performance. The processing time per record (which included updating files and reindexing a data base) dropped from 60 seconds plus to less than 8 seconds; normal screen response went from one or two seconds delay to nearly instantaneous.

I have also successfully used a 1.2-MB RAM disk in a dual floppy machine. If the data files will fit on one floppy and the program and operating system on another, that old floppy

machine (that you don't use so much any more) will blow your socks off. An additional note: I consider the use of an uninterruptable power supply absolutely necessary if you are doing any complex work in a RAM disk, as the slightest power interruption will cause you to loose everything.

I hope that this letter will help you see where you are and how you can get more benefit out of your system.

\* \* \*

#### TECHNICALLY YOURS MEMBERSHIP

As a result of Victor\*Group's sponsorship, members may obtain an individual membership in Technically Yours for \$11.95 (rather than the normal cost of \$29.95 per year) including MicroVision Magazine. In their words:

Technically Yours is a growing company designed to help computer users. We are a particular benefit to users who feel abandoned by the dealer or manufacturer who sold them their computer. The founders of Technically Yours come from a Victor background, and we appreciate the V9000 user's need for help.

With support as our primary focus, Technically Yours has become affiliated with MicroVision Magazine so that our members receive this excellent source of information as part of their membership; we have developed our Purchasing Assistance (PA) program to help members find the products that they are looking for, at the best possible prices; we regularly send out information so that our members can keep up-to-date with the equipment and software that they have or intend to get; and we continue to expand our list of affiliates in the U. S. and Canada so that we can refer members to repair and consulting services.

To join or for futher information, write:

Technically Yours Enterprises, Inc.  
P.O. Box 1414,  
Aptos, CA 95001  
1-408-688-5201

\* \* \*



## PLUS PC WITH HARD DISK IN VICTOR FORMAT / Wayne Crouch

The following is an explanation of my discoveries concerning hard disk access with a PLUS-PC system.

I started out with the desire to be able to read IBM formatted (360K) floppies on my Victor 9000. I specifically did not wish to lose any of the significant capabilities I already had. Since I could not find any other way to gain access to IBM disks, I purchased a PLUS-PC kit to do this. Well, it has been quite a road to travel, but I am almost there - though it took a lot of hard work and frustration to get to this point. I want to document how I got there, in case the information is of use to others.

After installing the Plus-PC, which was a chore (due mostly to the documentation provided by VICTOR), I was horrified to discover that the Plus-PC operating system and/or hardware were operating my hard disk in a strange manner. Instead of the normal rapid "click" noises from the hard disk, I now heard LONG low "moans". It appeared that the disk was being "single cylinder" stepped on every head motion. The noise was bad enough (it evoked extreme concerns for the continued health of my drive), but the resulting access time was totally unacceptable.

What had been a very fast hard disk based system, was now SLOWER than a floppy system! It sometimes took as many as five seconds to find a file. This made the Plus-PC completely worthless in my system. I had to either find a solution or get rid of the Plus-PC. Fortunately, I found such a solution, no thanks to Victor. Let me add that what follows seems straight forward and logical. However, it is the result of several weeks of hard work, and of exploring quite a few blind alleys.

First, I need to establish the normal operating sequences for the Plus-PC, at least as I understand them. The Plus-PC overrides the standard Victor boot ROMs. When cold starting the machine, or following a "reset" from the back panel, the Plus-PC boot ROMs gain control.

The Plus-PC boot ROMs first run various diagnostics, which include code to check the machine configuration for disk drives. This code has a "feature" in that, if it finds that a hard disk controller exists, it loops until it is able to access drive "0". When it gains access (drive powered up & ready), it then performs a "reset" on the drive, which will move the heads to cylinder zero. It does not appear that the contents of the drive matter AT THIS POINT, just its existence and status. The poorly thought out part of this is that the code must complete its sequence BEFORE it will display anything or make any sounds on the speaker. The moral here is that, if you have a hard drive, it MUST be powered up and ready

before you get anything at all out of the system.

Once past the initialization logic, the screen clears (possibly with some initial snow and/or flickering) and the speaker "beeps". The memory test is then performed, with the results displayed at the bottom of the screen. The amount of memory found is displayed in a format unique to the Plus-PC. Two numbers are shown. The second is the actual amount of memory found, which MUST not exceed 768K - else the system will not boot properly. The first is the smaller of (actual amount minus 64K) and (640K). The 64K reflects the fact that the Plus-PC emulation software "steals" 64K for its own uses. The 640K apparently reflects compatibility with the IBM-PC 640K limit on RAM. At this point the Plus-PC boot ROMs begin attempting to boot the system. They try booting in order from: Left Floppy, Right Floppy, and Hard Disk "0". If the Plus-PC operating system (Floppies or PLUS-PC formatted Hard Disk) or an IBM operating system (Floppies only, I think) is found, it boots successfully. If no valid operating system is found, the standard "X" appears on the monitor.

Now consider what happens if a standard Victor OS is present on a boot disk you try to use. First, if the hard disk is formatted as other than a PLUS-PC hard disk, the Plus-PC boot ROMs effectively ignore it. They will perform the earlier mentioned "reset", but appear to thereafter ignore the existence of the hard disk. I assume that the Plus-PC boot code reads the hard disk "label" sector to determine this. Therefore, a hard disk containing a standard Victor OS (which will, necessarily, be formatted in standard VICTOR style) will not boot normally on a Plus-PC.

If a floppy diskette contains a V9000 OS, it will boot - in a most interesting manner. The Plus-PC boot code appears to read the floppy "label" sector and determine that it contains a non Plus-PC (and non IBM) OS. At that point, the boot code performs some special "magic". The Plus-PC boot ROMs are "mapped out", and control reverts to the standard Victor (F3F7) boot ROMs. At this point, the machine reverts to normal Victor mode operations. The standard boot ROMs now do their thing, and the operations from that point appear EXACTLY the same as on a standard Victor. The floppy disk based Victor OS then boots successfully.

Once the machine has been booted, the hard disk can be accessed through the OS, with no problems. The only access problems lie with the boot ROMs, not with the OS.

So one can access a hard disk, formatted to standard Victor mode, on a Plus-PC machine. What is more, it is possible to

boot with such. This relies on the boot ROM mode switch described above. First boot (say DOS 3.1) from a floppy disk. Then, make the floppies not ready (open the doors), and use the KEYBOARD boot request (Alt/Lock/.) function. Since the standard (F3F7) boot ROMs are still in control, they first fail booting from the floppies (which are not ready), and then succeed in booting from the hard disk. Note that, for this to work, you MUST use the keyboard re-boot, or a back panel "reset", because a "power off/on" will put the Plus-PC boot ROMs back in control! Once you have performed the above sequence, you may reboot at will, as often as you wish, from the hard disk, as long as you do not cycle the power off/on or depress the reset button.

It is even possible to shorten the above sequence a bit, by proceeding as follows. First, with a standard Victor OS boot floppy in place, and the hard disk powered up and ready, initiate a re-boot by any technique you wish (keyboard, or reset, or power cycle). Let the boot sequence proceed through the memory test to reading the floppy "label". The Plus-PC boot ROMs, if active, will then map themselves out, thereby turning control over to the standard Victor boot ROMs. At this point, the Victor boot ROMs will go into their memory/device tests. This will take a few moments. While this is going on, and the screen is blank, open the floppy drive door. When the Victor boot ROMs try to read the floppy, they will fail, and will, after a delay, proceed to boot from the hard disk. The exact timing of the above is not nearly as difficult as it seems. With a little practice, it becomes easy. If you mess up, no lasting harm is done, merely try again.

Using this technique, it is quite practical to boot a standard Victor OS, such as 3.10 from a Victor formatted hard disk. The only limitation is that it does require operator intervention and the initial use of some Victor OS floppy boot disk. The OS version is not important for this purpose, as it is only used to fool the Plus-PC boot ROMs.

One word of caution. Since the standard Victor (F3F7) boot ROMs will stay in control until a (back plane) "reset" is performed, or until power is cycled, if you have been running a standard Victor OS and wish to boot from a Plus-PC or IBM OS, you can NOT do so via the keyboard. Attempts to do so will result in an "X" display, since the F3F7 boot ROMs do not recognize PLUS-PC or IBM OSs. You must use a "reset" or cycle the power to return control to the Plus-PC boot ROMs.

Thus, I have now achieved a workable compromise. I can use the same machine as a Plus-PC (floppy access only, but either Victor or IBM format), or as a standard Victor 9000 (floppy and/or hard disk). I can not access the hard disk while running a Plus-PC OS, but such is of little use anyway. I can transfer data between modes via floppies. And, I am limited to 768K of RAM instead of 896K, which to me, is an acceptable price to pay.

One lingering "bug" exists - no matter what I do, I can't get my 256K Plus-PC machine to work reliably with my VALID TECH (+640K) memory board. As per the instructions provided by VALID, I have set the switches all to "Off", limiting the board to the 512K maximum allowed with the PLUS-PC. I have even tried depopulating the board to 512K by removing the chips from the two 64K rows. No matter! When I try and boot with the board in place, I get various not very interesting displays, but the bottom line is that the machine will not boot.

Sometimes I get 192/256 for the memory size line from the Plus-PC boot ROMs. This value would be correct, if the VALID board were not installed. Sometimes I get 192/768, which makes no sense at all. Sometimes I get various other bad values, but in no case does the boot succeed. It "hangs" at various points, sometimes requiring a complete power off/on sequence to clear it.

I have also tried taking the system back to a standard (non Plus-PC) Victor configuration, and the board seems to work fine in such a setting. I suppose that it is possible that I have an intermittent problem with the board, and that it is just coincidence that it chooses to work in the Victor mode. However, I have no way to exhaustively test this possibility. I really want to continue using this board, but have run out of ideas. I would appreciate help from anyone.

\* \* \*

---

#### SHEET LABEL PRINTING WITH SPELLBINDER /Boardman Moore

If you need multiple copies of mail labels, the three- across, thirty-three-labels-per-sheet format can save both time and money. Master copies can be distributed and labels reproduced on any copy machine, a feature not possible with the fanfold format. The advantage may be illusory, however, if your 3.5-inch fanfold label data cannot be accommodated on the 2.83-inch sheet label. Using condensed printing (17.1 characters per inch) with the SPELLBINDER MLABEL MACRO solves the space problem, but introduces another; the space between columns is reduced by a similar ratio.

LEXISOFT's Assistant Manager - Technical Support, Venita Kent-Basham helped with this procedure:

1. Set screen line width to 159 (L159 in Command mode).
2. Read the file MLABEL.WPM into your SPELLBINDER workspace.
3. Change line 246 (if necessary) so as to read:  
:SS 15 \$G 3 3 3 3 \$G 3 3 3 3 3 3 3 3 3 3 3
4. Save as MLABEL2.WPM to distinguish this macro from the original.
5. When using the new macro, change the "WIDTH OF LABEL IN CHARACTERS ?" from 33 to 48 (2.83 inches x 17.1 characters per inch).

\* \* \*

# VICTOR 9000/VICKI BIOS LISTING / Victor Technologies

Victor Technologies, Inc.  
380 El Pueblo Road  
Scotts Valley, CA 95066-4269

## VICTOR 9000/VICKI BIOS Listing Order Form

The VICTOR 9000/VICKI BIOS Listing is a diskette containing the commented source in PL/M and 8086 Assembly Language of the MS-DOS 2.11 or 3.1 Basic Input Output System for the VICTOR 9000 and VICKI computers. This material is copyright by Victor Technologies, Inc. It is made available on the following basis:

1. That the information contained in the BIOS Listing is not to be used to develop software or hardware which directly

competes with products marketed and sold by Victor Technologies, Inc., or its subsidiaries.

2. If this information is used by the purchaser to make modifications to the BIOS distributed by Victor Technologies, Inc., the purchaser understands that Victor Technologies, Inc., its subsidiaries and its distributors will not be responsible for support of the modified BIOS.

3. Purchaser's customers must be so informed in writing that Victor Technologies, Inc., its subsidiaries and its distributors will not be responsible for support of the modified BIOS or of any software package running in an environment with a modified BIOS.

4. This information is made available solely to the signatory below and is not to be distributed without the written permission of Victor Technologies, Inc.

I agree to the four conditions above and would like to order \_\_\_\_\_ copies of the VICTOR 9000/VICKI BIOS Listings at fifty dollars (\$50.00 US) each, plus \$2.50 shipping and handling charge.

Signed \_\_\_\_\_

Date \_\_\_\_\_

Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zipcode \_\_\_\_\_

Check one

Method of Payment: \_\_\_\_\_  
Check \_\_\_\_\_ Money Order \_\_\_\_\_ Visa/MasterCard 2.11 \_\_\_\_\_

Visa/Master Card Number \_\_\_\_\_ 3.1 \_\_\_\_\_

Expiration Date \_\_\_\_\_

Authorizing Signature \_\_\_\_\_

This offer good only in the United States of America.

\* \* \*

# FROM THE BOARDS

[FROM THE BOARDS is assembled from messages appearing on various (generally Victor-specific) bulletin boards, particularly The PUB, by John and Mary Jo Knobelsdorf. Only minor editing is done to correct obvious typographical errors and to eliminate some superfluous material. An effort has been made toward a degree of standardized formatting and the elimination of excess space to the extent practical without degrading legibility. Garbles that cannot be readily rectified are allowed to remain. Believing that considerable worthwhile material is contained in these excerpts, our intention is to include this feature in each issue of VictorTalk, as space will allow. - Ed.]

## WORDSTAR

#33 PAUL LAUSIER, Subject: Wordstar

Just when I get proficient at Wordstar and get on the Pub for some hot tips, all I find are people talking about Wordperfect! I'm not too bright, it took me a long time to subdue WS, and I don't look forward to learning another word-processing program. But I would if I were convinced it would be worth it. Is Wordperfect so superior that I should make the switch?

#34 BILL WITOWSKI, Subject: Wordstar v. WPerfect

WPerfect has features you'll eventually want that are not provided by any other word processor. This opinion comes not just from me, but the latest review of WPerfect 4.2 found in a recent INFOWORLD. If you are \*really\* stuck on the Wordstar keyboard, then WPerfect can give you a Wordstar keyboard, a copy of which (I think) is in the files section of this BBS. Try it, you'll like it.

#35 BARRY COHEN, Subject: Re: Wordstar

Paul, I'm one of the dinosaurs still using WS. I have an office full of people trained and I won't switch until a program comes along that does the typing for us!

I may be able to give you some tips, and some neat utilities I've written. One checks print control characters to make sure that every one has a mate. One extracts an address from a letter and prints it on an envelope.

Get in touch if you're interested.

#36 PAUL LAUSIER, Subject: Wordstar

Dear Barry, from one dinosaur to another! Thank you for your response. I would certainly be interested in your WS utilities, especially the one which picks the address from a letter and prints the envelope.

My partner and I have been running Wordstar in our law office for nearly four years, but we don't qualify as hackers of any grade, so we hope to gain an education from the Pub. We have managed a few enhancements which give the lie to the term "Wordstare", which was used by a member of the great unwashed a while back. Can't imagine using that term for Wordstar running on Ramdrive with Drive C as the default, especially after changing the value in location 2D2, as described in the first issue of Microvision. The author indicated there were other worthwhile modifications but quit before revealing them.

I have no way of knowing whether this will interest you, but from letters and pleadings that come in from time to time, I know others have encountered the same problem we did. Our NEC printer would sometimes widen the spaces between letters, especially on a line with many short words. Not enough to make the line illegible, but enough so you knew it wasn't right. NEC had no solution, Micropro suggested .UJ0 at the beginning of every document, which did the trick but was also a pain. I had one key keygened for just .UJ0. One day, because Install runs so quickly in Ramdrive, I decided to run through the menu of printers, one by one, to see if there was a better match. Sure enough, Centronix 353 seems to be perfect. No funny spacing, no more need for .UJ0. Now if I can just get WS to produce the fractions on the 1/2 1/4 key, my life will be complete. But am I right, the ASCII codes for 1/2 and 1/4 are out of Wordstar's range?

Keep those cards and letters coming!

#38 DAVID RENSIN, Subject: Wordstar

Anyone see the latest spec sheet for the new Wordstar 4.0? Looks like someone at Micropro had a copy of WordPerfect laying around. Too bad no Victor user will ever be able to use WS 4.0. In fact, there's the press release ---

PKARC/PKXARC.COM, Download Problem & XMODEM

#116 SYSOP, Subject: PKARC/PKXARC.COM

The PUB is now using the latest versions of Phil Katz's FAST!

archive create and extract utilities. Any .ARC files dated 1 February 1987 or later may require the use of PKARC.COM ver 2.0 or PKXARC.COM ver 3.4 for successful "decompression".

These archive utilities allow the tightest file compression and the fastest unARCHiving. Phil Katz's newest routines also allow comments to be added to the files as they are archived.

PKARC and PKXARC are downward compatible with previous versions and can also be used to unpack .ARC files created with SEAware's (Software Enhancement Associates) ARC.EXE, ver 5.12, and earlier.

---

Again, you must have PKARC.COM version 2.0 or PKXARC.COM version 3.4 or later in order to successfully unpack some .ARC files placed on this BBS after 1 February 1987.

---

#124 SYSOP, Subject: Download Problems

Bill Witowski was having problems downloading TSB-PPC.ARC using Crosstalk's XModem. It would stop at block 9. I've taken the ARC file apart and tried rearranging the files. Now it stops at Block 27.

Wow, what progress!

At least five people successfully downloaded TSB-PPC.ARC. Were you all using Kermit?

If someone can't help the Fuzzy Bear solve this problem, I may just have to give up.

I'm ARCing with the latest version of PKARC. It looks like the downloads are hanging on an interior end of file marker. Each of the text files is created with WordStar in the nondocument mode. I'm sure that others as well as myself have downloaded text files...so why all of a sudden are we having problems. I'd hate to have to rebuild File Section #2 from scratch; it's already taken me many bear hours of labor to get as far as I have.

I'm stumped...ideas anyone?

#125 MARTY MOLESKI, Subject: XModem Glitch

Fuzzy, (it's not your .ARC file that's causing Xt's XM command to hang; it's bad phone lines or bad karma. XModem can handle all 256 ASCII characters in a file, CTRL Z included, because it sends packets and use all kinds of conventions to determine when the transmission is finished.

Do not rework Files 2; get the victim to try Vicmini instead.

#126 SYSOP, Subject: Re: XModem Glitch

Marty, it's not bad phone lines because:

- 1) I can download other files without incident.
- 2) I've placed the file (both before and after repacking) in three different directories, and XModem terminates at the same block in the download each and every time you attempt it on any of the three files — first time around on Block 9 and second time around on Block 27. Bill Witowski's attempts also terminated at Block 9. Someone else just reported a similar problem at Block 7xx in ADVENTUR.ARC. I would advise everyone to be wary of downloading \*HUGE\* files until we sort this problem out.

**dBASE III tool for VICTOR**

#93 JOHN NEWTON, Subject: dBASE III tool for VICTOR

I've been doing some dBASE III development on my Victor then running it through CLIPPER to test it, as I don't have a VICTOR version of dBASE III. To create .DBF file required that I create them in the office then transfer them to my VICTOR. This is time consuming to say the least. So I started looking for something that might help me. I found a number of PD programs that almost did what I wanted. Well to make a long story short I kludged a number of PD programs together to do what I need. The results is DEDIT. I placed it in the the UPLOAD directory and the file includes a batch file to modcon CLIP.KB the call the program. I had to create a special keyboard so that the cursor keys work. Now all I need to do is figure out how to get the programs to read the VICTOR function keys. Maybe someone out there can help. I've also included the source code, for those so inclined to tinker and improve, please keep a record of updates in the DEDIT.PRG file.

So improvements that I plan to incorporate include reading and creating index files, modifying the .DBF structure and maybe a report generator. Just because I plan on doing it doesn't mean that some of you should wait, get started now, it might be months, maybe even years before I get it done. (Looks like I should take a typing class.)

As to who gets credit for DEDIT, well there are a lot of files from different places, and I don't remember who all's I used. All I did was to kludge it together, so it may seem crude, but it works. Have fun with it.

\* \* \*

# LIBRARY REPORT / John Knobelsdorf

This will announce the release of three more V\*G public domain disks. They are #82, #83 and #84. They are each packed full of important, useful and fun files. As in the past, some of the files on these diskettes are in compressed form. Use NUSQ (on PD #73) to "Unsqueeze"; use LUE (on PD #73) to "Unlibrary"; use PKXARC (on PD #83) to "Unarchive".

Recent V\*G public domain disks have referred to ARC512.EXE for archive file creation, and ARCE118.COM for archive file extraction. Various of the SYSOP's and other persons who do a lot of uploading and downloading of files via telecommunications have begun using some similar programs for file handling. The "new" programs were developed by Phil Katz; they do the job faster and more efficiently.

The new programs are PKARC.COM ver. 2.0, and PKXARC.COM ver. 3.4, and they are on PD #83. It is suggested that you read PKXARC.DOC, and use PKXARC.COM to unarchive PKARC.ARC to obtain PKARC.DOC and PKARC.COM.

PKARC and PKXARC are downward compatible with previous versions, and can also be used to unarchive .ARC files created with ARC512.EXE.

Beginning with disk PD #83, most of the .ARC files and programs that appear on V\*G public domain disks will have been archived with PKARC.COM. It is recommended that you begin using PKXARC.COM for all your unarchiving activities.

Disk #82. This disk contains a group of files provided to us by Merle Witcraft of Carmichael, CA who has modified them to run on the V9000. These files came from a set of statistical programs called "EPISTAT" written by Tracy L. Gustafson of Round Rock, Texas for the IBM-PC. (Note: This set of programs is an updated version of the package by the same name on PD #60.)

To use these files: 1. Create a self-booting disk containing the operating system of your choice. 2. Copy the complete set of VBASICA files (released by Victor Technologies with their version of MS-DOS 3.1) to it. 3. Copy this set of files to your disk. 4. Call up VBASICA and run "EPISTAT".

The EPISTAT files are DATA-ONE.BAK, EPISETUP.DAT, EXAMPLE, PRINTDOC, SS1, SS2, and the following .BAS files (.BAS omitted here for brevity): ANOVA, BAYES, BINOMIAL, CHISQR, CORRELAT, DATA-ONE, EPIMRG, EPISTAT, FILETRAN, FISHERS, FORTRANS, HISTOGRM,

LNREGRES, MCNEMAR, MHCHIMLT, MHCHISQR, NORMAL, NUMBER, POISSON, RANDOMIZ, RANKTEST, RATEADJ, SAMPLSIZ, SCATRGRM, SELECT, T-TEST, XTAB.

A set of files were gathered by John Pietro of San Francisco from Victor Circle NewsDisk #9, "The Month of the Batch". These programs contain various utilities for enhancing batch processing on the V9000, along with a complete tutorial demonstrating the use of the utilities. Note: Read the file BATCH.DOC for further information about the 31 files from "The Month of the Batch".

A number of game programs were downloaded from Victor\*Group's BAVUG bulletin board. (Note: Be sure to de-compress the .ARC files before trying to run them. The .BAS programs will run using VBASICA or GW-Basic. The .EXE programs are executable by themselves.)

Following is a listing of the games on this disk. ACYDUCY.ARC, a game of Acey Ducey; ADVBAS.ARC, create your own adventure games; AIRTRAX.ARC, Air Traffic Controller game; CENTI.ARC, centipede game for V9000; FOOTBALL.BAS, a game of football; GCIRCLE.ARC, Great Circle navigation; GOLF.BAS, a golf game; LANDER.ASC, a Lunar Lander game; LUNAR1.BAS, a different lunar lander game; MONOPOLY.ARC, a game of Monopoly; PACMAN.BAS, the game of Pacman; SPACEWAR.ARC, a graphics space game by Bill Seiler.

Disk #83. The above-referenced PKARC and PKXARC programs for file archiving and extraction. The latest versions of VICMINI (ver 3.4) and VKERM229 (ver 2.29) that are available for the V9000, and other communications related files.

There are various Victor Information Files and Technical Bulletins which were provided to field service personnel, and are for use with the V9000, the Plus-PC and the VI. There is a wealth of information there, and to find what you want, TYPE or LIST the three index files (FETB.LST, NCSCTB.LST, and TSB.LST). The indices contain the subjects and filenames for specific bulletins. The .ARC files may be unarced to obtain the individual bulletins.

There are a couple of games, REACTION and SWITCH. These are stand-alone versions of the games of the same name on Victor Circle NewsDisk 22.

Michael Wishnietsky, Sysop of the PUB (BBS) at the Edge of

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
P.O. Box 50893  
Palo Alto, CA 94303

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
City _____ State & Zip _____	_____
Country _____ Phone _____	Order shipped _____

Circle diskette numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60. See Vol. 4, No. 2 for a brief description of disks 61-64; Vol. 4, No. 3 for disks 65-71; Vol. 4, No. 6 for disks 72-77; Vol 5, No. 1 for disks 78-81; Vol. 5, No. 2 for disks 82-84. A catalog supplement, "Victor\*Group Public Domain Library, Part II, Disk #61 through #84", will be available soon.

1	2	3	4	5	6	7	8	9	10		
11	12	13	14	15	16	17	18	19	20		
21	22	23	24	25	26	27	28	29	30		
31	32	33	34*	35*	36	37	38	39	40		
			*(34 & 35 are two disk set)								
41	42	43	44	45	46	47	48	49	50		
51	52	53	54	55	56*	57*	---	59	60		
					*(56 & 57 are a two disk set)						
61	62	63	64	65	66	67	68	69	70		
71	72	73	74	75	76	77	78	79	80		
81	82	83	84	---	---	---	---	---	---		

1987 Victor\*Group Member: No \_\_\_\_\_ Yes \_\_\_\_\_ Date joined \_\_\_\_\_

Number of diskettes ordered \_\_\_\_\_ x \$5.00 (members only).... \$ \_\_\_\_\_

Number of diskettes ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage and Handling (domestic)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

1987 Membership fee \$35 (receive all 1987 issues of VictorTalk). \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.)..... \$ \_\_\_\_\_

LIBRARY REPORT (continued from page 22)

the Electronic Forest provided a few FUN FILES which he shares with us. Titles are: The Ultimate Customer Service Program; How it all started; Our Lawyer Writes; MIT "Hacker's Dictionary" ca. 1982; and Characteristics of the Real Programmer.

Last, is a list of Victor User Groups as of 03/09/87.

Disk #84. A copy of PKXARC.COM was added at the last minute for your convenience, at the insistence of "a user". You will not find this file listed in the DISKID84.TXT file, because there was not enough room left on the disk (zero bytes free) to permit editing the diskid. The PKXARC.COM file will be listed in the regular directory of the disk.

Several "Additional Diskid Notes" which came from Victor Technologies. These "Notes" are for MULTIPLAN; MS-BASIC; MS-DOS 1.25a BIOS; MS-DOS 2.11 BIOS; PMATE; and WORDSTAR.

Some Newsletter Columns from the Victor Users Group of Denver, including a few of their Dr. DOS articles.

There are several games, all of which (I am told) WILL run on the V9000 without difficulty or emulation. If anyone finds that to be an untrue statement, please let me know.

Programmers and How-To-Do-It Files, five PMATE utility files, sixteen files of various type for use in ways too numerous to describe.

The Librarian's move to the hills of central Texas is scheduled for April 24, 1987. Our gate and entry off of the local ranch road is between two creeks that can and have blocked travel to town. "If the good Lord's willing, and the creeks don't rise", there will be a couple more new disks released in June '87. Wish us luck. \* \* \*

VictorTalk

Address inquiries to: P. O. Box 50893  
Palo Alto, CA 94303

Subscriptions: Nancy Weaver

Editor: Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

Contributors: Wayne Crouch, Roger Habeck, Franz Hirner, John Knobelsdorf, Mary Jo Knobelsdorf, Boardman Moore, Jack Prichard.

VictorTalk is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in VictorTalk. Copyright 1987 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

VictorTalk is an independent publication not affiliated in any way with Victor Technologies, Inc. Victor is a registered trademark of Victor Technologies, Inc.

Submit Articles on MSDOS disks (single-sided, please) in ASCII or Spellbinder format, with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

Yes, I want to join.

Enclosed is a check/money order for \$35.00 (foreign membership, \$40 in U.S. dollars) for my 1987 Victor\*Group membership.

Name \_\_\_\_\_  
Title \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_

Make checks payable to:

VictorTalk

P. O. Box 50893  
Palo Alto, CA 94303

City \_\_\_\_\_  
State \_\_\_\_\_ ZIP \_\_\_\_\_  
Country \_\_\_\_\_  
Phone \_\_\_\_\_



## A LOOK BACK / John and Mary Jo Knobelsdorf

[This article, written for the Anniversary Issue of Victor Circle NewsDisk24, seemed particularly appropriate to include in VictorTalk. It contains much of interest and value to both old and new users of the Victor 9000. Although the authors have moved out of the Bay Area, they will continue to contribute their talents to our group, both in articles in VictorTalk and the continued stewardship of the Victor\*Group Public Domain Library. Perhaps we will also be honored by occasional presentations when they visit the Bay Area. We are happy to publish this article with their permission. - Ed.]

In late 1983 we moved from Texas to California and purchased our first Victor 9000. At that time Victor computers were rather pricey but we had shopped for about 6 months and found nothing that we thought was comparable so we sprang for the Victor. Several years and many Victors later, we know that we made the right decision.

After we entered the Victor world, we had the opportunity to interact with some wonderful people and we learned a lot. Now we want to share some of our experiences and a few observations.

We joined Victor\*Group, the San Francisco bay area Victor Users Group, in 1983 after we bought our computer. At that time there were three chapters, one in San Francisco, one in Walnut Creek, and one in Fremont. Each chapter met once a month and we made it a point to go to all three meetings if we could. We noticed that there were two categories of members, those who said very little and those few who said a lot. We said very little because we knew very little and we did not want anyone to know how lost we were. John Kimber, the program chairman at the Fremont meetings, used to have each member give their name, the software they used and their primary uses for the V9000. We gave some fuzzy answer because our primary uses were so meager. Now we could give better responses, but we no longer meet in Fremont and John Kimber is no longer program chairman.

We had some computer experience in our background, Mary Jo had a lot more than John, but we simply did not know what to do with the computer we owned. Probably one contributing factor was that any particular work that we needed or wanted to do could be done with a red Big Chief Tablet and a fat, black pencil, or if it was really complicated we could add our Sharp hand calculator.

It now seems strange that we would KNOW that we "needed" a V9000, but couldn't really decide how to put it to good use

beyond some vague notion about writing letters, putting the budget on a spreadsheet, and maintaining a mailing list.

We started with an excellent word processor, Spellbinder, from Lexisoft in Davis, CA, and our Victor Operator's Manual. We spent all our spare time, and a lot of non-spare time, learning about MS-DOS 1.25, our hardware, and Spellbinder. As new users, we had to develop an understanding of how the hardware uses the operating system and the software, and what our role was. Simply stated, what did the Victor do? What did MS-DOS 1.25 do? What did Spellbinder do? And what were we supposed to do? The manuals were full of stuff we didn't understand. Did that matter? Finally, we became comfortable doing word processing on the V9000.

Our first suggestion for the new V9000 owner is, read enough of the MS-DOS manual and your first software manual to get some work done on the computer. Go back and read the manual again. This time start with the Appendices. They often have vital information and excellent summary listings. Mix working on the computer and reading the manual. Later, start at the beginning and re-read every page of the manual. Every time you read something that you do not understand, write a note/question on a yellow post-it and put the note on that page in the manual. After you finish studying the manual, go back and remove as many post-its as you can. Hopefully, you will eventually be able to remove all of the post-its in the manual.

After we learned that Victor would no longer make the V9000, we started buying used Victors so we would always have a spare one. Mary Jo had started to do some instructing and consulting, and this helped justify the cost of our purchases. We have worked with many systems and many configurations. Mary Jo now rents or leases equipment (computers, printers, modems, etc.), with or without training, to persons who want to learn about computers, or who have a temporary need for a computer. She also does a bit of consulting, and buying and selling of equipment.

We hear a lot about a 10Meg hard disk not being so much memory anymore. For us, a 10 Meg HD is sufficient. Given a choice, a double-sided floppy disk system with an external hard disk is the best of both worlds. All of the software and data can be put on the hard disk, leaving the two floppy disk drives available for disk and file copying, transfer, and backup operations. Our original purchase included a dual floppy computer and an external Univation 12Meg hard disk. It has never been a problem and is still working fine.

Probably one of the worst things that has happened to us was the tremendous decrease in the price of disks. Back in the olden days, we drove 20 miles to a computer store that had a special on Maxell SS disks, 3 boxes (30 disks) for \$100.00. Believe me, when we paid prices like that for disks, we were careful about how we used them. In fact, Mary Jo's first letters and resume are still on one of those disks. Now we pay very little for disks, use them with abandon and often find that we have multiple versions of the truth on the various disks. This is a poor work practice and we must someday regroup, rearrange, and get our act together. Try to keep from getting into this mess. It is easy to get into and difficult to get out of.

To compound the problem, we were fortunate/unfortunate enough to be in Victor's back yard when they decided to get rid of all of their V1's as well as their V9000's. We purchased V1's for gifts for our two sons and their families and added a few V1's to our list. Mary Jo "could always use them to teach software that was IBM specific." A V1 is nice, is very IBM compatible, and is useful if your software won't run on a Victor 9000. Our preference, the V9000.

Yes, we have worked with the Plus-PC also. It is so much like the V1 that it is difficult for us to distinguish one from the other. The major difference is that an IBM disk must be formatted on another computer before it can be used on the Plus-PC, but the V1 can do its own formatting of IBM disks. Other than that, there is very little difference and both computers are very easy to use. Conceptually, the V1 is two computers in one case. When you are running Victor software you go to the Victor mode (press three keys on the keyboard to go to Mode V) and run your program. When you are running IBM software you go to the IBM mode (press three keys on the keyboard to go to Mode I) and run your program. The really neat part is that \*THE COMPUTER DOES NOT CARE WHETHER THE FILES ON THE DISK ARE IN VICTOR FORMAT OR IBM FORMAT AND NEITHER DO YOU\*. Just put the disk in the computer and use it. The V1/Plus-PC checks to determine the format of the disk, reads the files and uses them. This is a tribute to Chuck Peddle and the staff that designed this hardware and software. They did an impressive job, even if it did not sell.

We plan to continue to use V9000's and available software. We have purchased enough machines to keep some going for a long time. We have enough software to keep us learning for a lifetime. When we need IBM compatibility, we will enlist the assistance of the V1 or Plus-PC. For us however, the IBM compatibility is for consulting and training.

We have put together a check list that we use when we are checking a used V9000. We must admit that we have both diagnostic disks and System Test and Burn In disks but we do

not put them to good use. This is what we do:

1. Boot with MS-DOS 1.25 from each floppy drive, first with the on/off switch, then with the reset button.
2. Boot with MS-DOS 2.11 from each floppy drive, first with the on/off switch, then with the reset button. Only the very early models will not boot with MS-DOS 2.11.
3. Boot with MS-DOS 3.1 from each floppy drive, first with the on/off switch, then with the reset button. Some V9000's with old ROMs will boot with 3.1, some will not.
4. Notice the value of RAM, and the way it is written when the system boots. A computer with new ROMs and a value of 896k is worth a lot more than one with old ROMs and a RAM value of 2000 or 4000.
5. Check the sound with some program that has sound. We use GWBASIC and MUSICBOX.BAS (on Victor\*Group's public domain disk #6 and #7).
6. Format a disk in each floppy drive. Set format's zone and error switches (format /z /e <ret>). If there are a lot of soft errors, there may be a problem with the alignment of the floppy drives. With a dual floppy system, diskcopy a disk from each drive to the other drive, again with the zone and error switches set.
7. Print a file. A handy ASCII file to print is CONFIG.SYS on the MS-DOS 2.11 boot disk. We take an MX-80 printer, so we test only the parallel port. We take our chances with the two serial ports.
8. Copy a file from every drive to every drive, including the hard disk and ramdrive to double check read/write capabilities.
9. If the computer has an internal harddisk, boot from the harddisk. Use chkdsk on the harddisk to determine the size of the harddisk. Read and write files to and from the hard disk and use HDPARK.EXE. To tell the truth, this will not tell you much about the harddisk other than it is working at that moment.
10. Run some of your favorite software from each drive and the hard disk (if there is one). We use Lotus 123 to view a graph on the screen, run a screen dump to the printer, and edit a small file with Edlin.
11. Use one of the diagnostic disks and run a RAM test.
12. If possible, turn on the computer for 24 hours and repeat these steps again.

13. Check the monitor glare screen for cuts and scratches, especially at the bottom left of the screen. Use the internal command, CLS, to clear the screen and notice how long it takes for the text to vanish.

14. See if the keyboard has all 10 function keys and the ALT values on the front of the numeric keys.

We have assembled a set of disks that are in a box labeled "Disks for Test and Travel." We have both double- and single-sided disks in the box. We have a Vicki boot disk, a Plus-PC boot disk, a VI boot disk as well as the various MS-DOS boot disks. You never know when you will need them, and disks don't cost very much.

Naturally, when we buy a used V9000 we know the rule is "buyer beware, and do not expect guarantees." If you use this check list, please realize that your computer may or may not be ok. If you want more assurance, take it to a service department and have them check it for you.

A few words about software. We are stockpiling software that we might use in the future. Sometimes we get software and manuals with used computers. We have gone to the computer stores that used to sell Victors and negotiated with them for any remaining V9000 software.

We use MS-DOS utilities and a number of public domain utilities a lot. We use Spellbinder and Fancy Font for word processing, Lotus 123 for spreadsheet applications, and dBASE II for database applications. We have an MS-BASIC and an MS-FORTRAN compiler for custom applications. We have a lot more software but we haven't read the manuals and we haven't tried to use the software so we won't try to evaluate it.

If you plan to keep or sell your V9000, you might want to consider purchasing one of the remaining copies of LOTUS 123, Release 1A, from Victor to include with your computer sale. That will make your offer more attractive. When all of the copies of Release 1A are gone, there will not be any more to purchase and there is no reason to expect a Release 2 of LOTUS that will run on the Victor.

If you want a real learning experience, work with some public domain software. There are a lot of useful utilities for use with various applications programs, some graphics applications, fun games, music, and trash out there. We spent three days "reworking" several games and educational programs so our five-year-old granddaughter could run them on a VI. Mary Jo had worked with all of those programs on the V9000 and knew that they were ok. With the VI, they all had to be tested again, some worked in Mode I, some worked in Mode V, some worked with MS-BASIC, and some worked with VBASICA. It was worth the time. Valerie can use the

computer, she knows that the F6 key is a shortcut and saves typing the word "files", and she knows to look for files that end in .BAS to run. She also knows that Mode V stands for Mode Valerie! And she is learning to type by using the computer, and she loves it.

If you want to learn BASIC or PASCAL or other languages, study your manuals or take a class. Then study some of the programs in the public domain that are written to run with that interpreter. Programs that end in .BAS are supposed to run with MS-BASIC, GWBASIC or VBASICA. Programs that end in .PAS are supposed to run with MS-PASCAL or Turbo PASCAL.

If you are a new user with a used V9000 and you want to get started with a minimum of cost you might want to consider the following "starter" software package:

1. A good word processing program. WordPerfect is a good choice for price, support and popularity. Our biased preference is Spellbinder but it is expensive and not as commonly used as WordPerfect.

2. MS-DOS 2.11 or MS-DOS 3.1 from Victor (Caution: you may need new ROMs, also available from Victor, to use MS-DOS 3.1.)

3. PC-File III from the public domain (Victor\*Group public domain disk #77). This is a good first relational data base management system.

4. For programming, use your MS-BASIC that came with the V9000 or VBASICA that comes with MS-DOS 3.1.

5. To get on the V9000 bulletin boards, an inexpensive modem (Avatex has a good one) and Vicmini or Kermit from the public domain.

- - - - - with a little more money to spend - - - - -

6. A spreadsheet, Report Manager, Multiplan, or Lotus 123.

7. When you need a powerful database management system, consider dBASE II (it is not too hard to learn to use) or dBASE III with Guy Gordon's IBM emulator (expensive).

Most of the software available for the Victor 9000 will run with 384k or less of RAM. Tony Casano's Knee Toe Graphics must have 384k and every V9000 owner should spend some time with Knee Toe Graphics. GWBASIC and VBASICA need 256k. Lotus 123 will run with 256k but large worksheets need 384k. For use and/or for resale considerations, 384k is "needed." If you have a computer with less than 384k, one way to upgrade inexpensively is to try to find someone willing to sell a used

memory upgrade board. Probably the least expensive memory upgrade is from Marvin Krull, a former Victor employee. We know Marvin personally, and know of his reputation from both current and former Victor employees. We would not hesitate to let him work on any computer we own. In fact, we watched him make a RAM upgrade on one of our computers. He is a master working on hardware. He will furnish the parts and labor and take a 128k computer to 512k for \$140.00. If you have a 256k computer he will take it to 896k for \$210.00. It is hard to beat a deal like that. (Marvin Krull, 9985 Empire Grade, Santa Cruz, CA 95060, (408) 427-2760.) Of course you would have to mail your CPU to Marvin. If you park the head of the hard disk and use the cardboard inserts in the floppy drive(s), the CPU should be safe to ship. Incidentally, don't use a disk for shipping, use the cardboard inserts. Notice that the cardboard insert has a tab on each end. The tab that is on the end that is inserted in the drive first secures the drive head and keeps it from shifting forward on the rails. It is especially important to transport the Vicki with these cardboard inserts in place. The Vicki handle is on the back of the computer and the drive heads are vertical and in position to travel forward on the rails toward the front of the computer, if the correct cardboard insert has not been put in place prior to picking up the Vickie by the handle. This is very hard on the drives and often damages them.

A fringe benefit of our owning a Victor 9000 has been working with special people from all over the world. Many of the V9000 owners we met at user group meetings we hope to keep as lifelong friends. Since we were so close to Scotts Valley, Victor Technologies' home, we have been able to visit the area frequently and meet some special and helpful Victor employees. Our work with the public domain and VictorTalk put us in touch with another group of special people, many of whom are members of other user groups and work with Victor specific publications and the public domain. The bulletin boards and Victor Circle expanded our contacts even more. You might guess that we have spent a lot of time working with Victor 9000 users and Victor 9000 computers. Our conclusion is that it was worth it.

We appreciate the opportunity offered by this Anniversary Issue of Victor Circle NewsDisk24 to express our sincere thanks to each and every one of you for all of your guidance, assistance, and camaraderie. May you all enjoy long and healthy and bountiful lives, and reap the benefits of Victor Circle membership. Viva NewsDisks!

\* \* \*

[If you are not already a member of Victor Circle and receiving the monthly-issued NewsDisk, see the note in the next column, listing the current distributors. - Ed.]

\* \* \*

## VICTOR CIRCLE NEWSDISKS

As gleaned from the VictorCircle NewsDisk24 (April 28, 1987), NewsDisks are currently distributed by the following:

CHRIS DeSALVO, 19 Ridge Road, Boulder, CO 80303:  
Colorado, Kansas, Montana, Nebraska, North Dakota,  
Oklahoma, South Dakota.

BILL DODGE, 62 Beaudry St., Marlboro, MA 01752:  
Connecticut, Maine, Massachusetts, New Hampshire, Rhode  
Island, Vermont.

ISAAC FELDHEIM, 10 Waverly Place, Monsey, New York  
10952: New Jersey, New York.

JAMES HUGHES, P.O. Box 1146, Smithfield, NC 27577-1146:  
Alabama, Arkansas, Florida, Georgia, Idaho, Louisiana,  
Mississippi, North Carolina, Ohio, Oregon, South Carolina,  
Tennessee, Texas, Washington.

DOUG JEATT, Pucks Blarney, Crouch Lane, Winkfield,  
Windsor, Berkshire, England SL4 4TF: Great Britain.

TONY POMPA, Box 1352, Laurel, MD 20707: Delaware,  
District of Columbia, Maryland, Pennsylvania, Virginia, West  
Virginia.

ANN REID, P.O. Box 3244, Scottsdale, AZ 85257:  
Alaska, Arizona, Hawaii, Nevada, New Mexico, Utah,  
Wyoming.

DAVID RENSIN, 4611 Vesper Avenue, Sherman Oaks, CA  
91403: California.

MIKE SIM, 253 Hindmarsh Drive, Rivett ACT 2611, Australia:  
Australia.

BOB SWENDSEN, 152 Aberdeen Drive, Dundee, IL 60118:  
Illinois, Indiana, Iowa, Kentucky, Michigan, Minnesota,  
Missouri, Wisconsin.

JOSE VASCONCELOS, M.D., The Brandon Clinic, P.O. Box  
280, Brandon, Manitoba, CANADA R7A 5Z2: Canada.

Many distributors will supply previous NewsDisks as well as the most current one. Though their procedures vary, you can't go wrong by sending your distributor floppy disks, formatted SS, and a stamped, self-addressed mailer of sufficient size to hold all the NewsDisks you are requesting. Some will provide floppies, mailers and/or postage at cost.

\* \* \*

# MS-BASIC, Part 3 / Jack Prichard

The first two articles of this series showed how to run and examine existing MSBASIC programs and how to create simple programs involved with the display and printing of formatted information. In this third article, we will develop the more general application of programming involving calculation and logical control of programs. A number of additional commands, statements, and functions will be introduced by way of short example programs.

To facilitate your review of material covered thus far, Table 1 gives page references to these first two articles for the various commands, statements, and functions previously introduced. Further, the tables of all MSBASIC commands, statements and functions have been revised to include references to the Victor VBASICA documentation. They follow as Tables 2 and 3. VBASICA documentation is generally improved over the earlier MSBASIC documentation and is now widely available. It was provided by Victor as part of the upgrade to MS-DOS 3.1 and as part of the documentation for the Victor VI computer. Tables 2 and 3 list, by usage category, all the MSBASIC and MSBASIC-compatible VBASICA commands, statements and functions. The tables replicate those published in Parts 1 and 2 of this series except for the addition of a third column of page references to the VBASICA documentation.

In the BASICA documentation, functions have been included with commands and statements in one long, alphabetically-arranged chapter. The number of entries has almost doubled with the addition of many entries for use with the expanded capabilities for graphics, sound, communications, etc. Fortunately, the table of contents covers (almost) all of the entries. Also, a fairly complete index is included (for the first time). Unfortunately, as is too often the case, a number of omissions and typographic errors occur. Those that have been discovered in the course of preparing this article are listed in Table 4. Hopefully, these several tables will ease your access to the Victor documentation.

Where practical, successive programs will be similar and will be generated by making minor modifications of the original program. This will simplify keying-in the programs and will give practice in use of the MSBASIC editor (introduced in Part 1). For ease of identification, the portion of a program that is changed from the prior program is shown in bold type.

In the examples, after the program listing, the display is shown for one or more runs. (If you want a printed output rather than a display, simply change the PRINT commands to LPRINT commands.)

## SUGGESTED PROCEDURE

To gain some further experience in MSBASIC programming, it is suggested that you follow this procedure for each of the examples shown, in order:

1. Read through the example program.
2. If any old material is not understood, look up the command, statement, or function involved in Table 1. Refer to where it was introduced in the prior Part 1 or Part 2. Review the material there and, if necessary, refer back to your Victor MSBASIC (or VBASICA) documentation and re-read the pertinent information. Use the Tables in the prior articles or Tables 2 and 3 in this article as an index.
3. Load MSBASIC and key in the example program. For successive program modifications, make them using the MSBASIC editor as described in Part 1.
4. LIST the program and make any necessary corrections using the editor.
5. When satisfied that the program appears correct, try running it.
6. If the program runs satisfactorily, save it to disk and proceed to the next example. If it does not run properly, re-edit it until it does run satisfactorily.
7. Proceed to the next example.

## ARITHMETIC OPERATORS

In the first example, the only new element is the introduction of the arithmetic operators + and \*. It is suggested you read the pertinent section on operators and their precedence in your Victor documentation. ('82 p. 1-10; '83 p. 8; '85 p. 2-16.) [For document identification ('82, '83, '84) see end of Table 3.]

```
10 REM Perimeter Program
20 INPUT "LENGTH"; L   'L=length
30 INPUT "WIDTH"; W   'W=width
40 LET P=2*L+2*W      'P=perimeter
50 PRINT "PERIMETER ="; P
60 END
Ok
RUN
LENGTH? 15
```

```
WIDTH? 10
PERIMETER = 50
Ok
```

Adding line 55 causes the program to rerun, rather than terminating after one run.

```
10 REM Perimeter Program
20 INPUT "LENGTH"; L 'L=length
30 INPUT "WIDTH"; W 'W=width
40 LET P=2*L+2*W 'P=perimeter
50 PRINT "PERIMETER ="; P
55 GOTO 20
60 END
Ok
RUN
LENGTH? 15
WIDTH? 10
PERIMETER = 50
LENGTH? ^C
Break in 20
Ok
```

Adding line 15 inserts space between display of runs.

```
10 REM Perimeter Program
15 PRINT
20 INPUT "LENGTH"; L 'L=length
30 INPUT "WIDTH"; W 'W=width
40 LET P=2*L+2*W 'P=perimeter
50 PRINT "PERIMETER ="; P
55 GOTO 15
60 END
Ok
RUN
```

```
LENGTH? 15
WIDTH? 10
PERIMETER = 50
```

```
LENGTH? 30
WIDTH? 25
PERIMETER = 110
```

```
LENGTH? ^C
Ok
```

## DATA AND READ STATEMENTS

Adding line 12, changing line 20, and deleting line 30 the modified program reads data from DATA statement.

```
10 REM Perimeter Program
12 DATA 15,10,30,25
```

```
15 PRINT
20 READ L,W 'L=length, W=width
40 LET P=2*L+2*W 'P=perimeter
50 PRINT "PERIMETER ="; P
55 GOTO 15
60 END
Ok
RUN
PERIMETER = 50
```

```
PERIMETER = 110
```

```
Out of DATA in 20
Ok
```

Adding lines 30 and 35 provides printout of input parameters.

```
10 REM Perimeter Program
12 DATA 15,10,30,25
15 PRINT
20 READ L,W 'L=length, W=width
30 PRINT "LENGTH ="; L
35 PRINT "WIDTH ="; W
40 LET P=2*L+2*W 'P=perimeter
50 PRINT "PERIMETER ="; P
55 GOTO 15
60 END
Ok
RUN
```

```
LENGTH = 15
WIDTH = 10
PERIMETER = 50
```

```
LENGTH = 30
WIDTH = 25
PERIMETER = 110
Out of DATA in 20
Ok
```

## UNCONDITIONAL BRANCHING

The unconditional switching (including branching) statements and commands have all been previously introduced. (These are: END, GOSUB . . . RETURN, GOTO, and SYSTEM.)

## CONDITIONAL BRANCHING

The conditional switching (including branching) statements and commands already introduced are: FOR . . . NEXT, IF . . . THEN, and ON . . . GOTO. To complete the introduction of the program switching category of statements and commands,

the following five examples use variants of the IF statement. These programs all give exactly the same results (and for that reason the results of the runs are given only for the first program). They differ only in the form of the IF commands (conditional branching) used: IF ... GOTO; IF ... GOTO ... ELSE; IF ... THEN with branch line number; IF ... THEN with statement; IF ... THEN ... ELSE with statements.

```
5 REM      IF...GOTO example
10 PRINT
20 INPUT "X ="; X
30 IF X>=0 GOTO 50
40 PRINT "NO REAL SQUARE ROOT": GOTO 10
50 PRINT SQR(X): GOTO 10
Ok
RUN
```

```
X =? 2
1.414214
```

```
X =? -2
NO REAL SQUARE ROOT
```

```
X =? ^C
Break in 20
Ok
```

```
5 REM      IF...GOTO...ELSE example
10 PRINT
20 INPUT "X ="; X
30 IF X>=0 GOTO 50 ELSE 40
40 PRINT "NO REAL SQUARE ROOT": GOTO 10
50 PRINT SQR(X): GOTO 10
```

In this case the ELSE is superfluous, as the prior example shows. (If the IF condition is not met, control passes to the next line.)

```
5 REM      IF...THEN example with branch
10 PRINT
20 INPUT "X ="; X
30 IF X>=0 THEN 50
40 PRINT "NO REAL SQUARE ROOT": GOTO 10
50 PRINT SQR(X): GOTO 10
```

```
5 REM      IF...THEN example with statement
10 PRINT
20 INPUT "X ="; X
30 IF X>=0 THEN PRINT SQR(X): GOTO 10
40 PRINT "NO REAL SQUARE ROOT": GOTO 10
```

```
5 REM      IF...THEN...ELSE example with statements
10 PRINT
20 INPUT "X ="; X
```

```
30 IF X>=0 THEN PRINT SQR(X) ELSE PRINT "NO REAL
SQUARE ROOT"
40 GOTO 10
```

(Text continued on page 12)

TABLE 1 - COMMANDS, STATEMENTS, AND FUNCTIONS PREVIOUSLY INTRODUCED (A=left column; B=right column)

COMMANDS AND STATEMENTS

	Part 1	Part 2
<b>Program Running</b>		
ALT-C	15B	
CONT	15B	
LOAD	15B	
RUN	15B	
SYSTEM	15B	
<b>File Operations</b>		
LOAD	15A	
MERGE	18B	
SAVE	18B	
<b>Program Creation</b>		
AUTO	19B	
EDIT	20A	
LIST	18A	
LLIST	18A	
MERGE	18B	
NEW	18A	
REM or '	19B	
<b>Program Switching</b>		
END		9
FOR ... NEXT		9
GOSUB ... RETURN		9
GOTO	15B	
IF ... THEN	19B	
ON ... GOTO	19B	
SYSTEM	15B	
<b>Program Initialization</b>	---	---
<b>Data I/O</b>		
INPUT	19B	
LPRINT	18A	
LPRINT USING		11
PRINT	18A	
<b>Data I/O (Random File)</b>	---	---
<b>Program Linkage</b>	---	---
<b>Data Manipulation</b>	---	---
<b>Error Handling</b>	---	---
<b>FUNCTIONS</b>		
<b>Mathematical Functions</b>	---	---
<b>String Functions</b>		
CHR\$	19A	
SPACE\$		11
STRING\$		11
<b>Miscellaneous Functions</b>		
SPC		13

TABLE 2. MS-BASIC COMMANDS AND STATEMENTS BY USAGE CATEGORY  
 (Including those of VBASICA common to MS-BASIC and a few MS-DOS commands usable in MS-BASIC)

[See notation list at end of table; reference identification at end of Table 3]

	Page Ref.				Page Ref.		
	'82	'83	'85		'82	'83	'85
<b>PROGRAM RUNNING</b>				<b>PROGRAM SWITCHING</b>			
ALT-C	UG 1-4	3		END	24	30	50
CONT		11	21 31	FOR<variable>=x TO y [STEP z]			
LOAD<filename>[,R]		47	43 104	* [<loop statements>]			
RUN[<line number>]		77	61 172	* NEXT[<variable>]			
STOP		79	63 185	[,<variable>]			
SYSTEM	UG 6-4	103	189	where x, y, and z are numeric expressions.	30	33	65
TROFF		81	64 194	GOSUB<line number>			
TRON		81	64 194	* RETURN	33	35	74
<b>FILE OPERATIONS</b>							169
CLOSE[[#]<file number>[,#] <file number...>]]		8	19 23	GOTO<line number>	34	35	75
FILES[<filename>]	UG 6-3	102	62	IF<expression> THEN			
KILL<filename>		40	39 94	<statement(s)> <line number>			
LOAD<filename>[,R]		47	43 103	[ELSE			
LOF(<file number>)	UG 6-3	103	108	<statement(s)> <line number>]	35	36	78
MERGE<filename>		50	44 112	IF<expression>GOTO<line number>			
NAME<old filename> AS <new filename>		52	45 116	[ELSE			
OPEN<mode>,[#]<file number>, <filename>,[<reclen>]		57	47 128	<statement(s)> <line number>]	35	36	78
RESET	UG 6-3	103	166	ON<expression>GOSUB			
SAVE<filename>[,A ,P]		78	62 173	<list of line numbers>	56	47	121
<b>PROGRAM CREATION</b>				ON<expression> GOTO			
AUTO[<line number>[,<increment>]]	2	15	4	<list of line numbers>	56	47	121
DELETE[<line number> [-<line number>]]		18	26 44	SYSTEM	UG 6-4	103	189
EDIT<line number>		20	27 49	WHILE<expression>			
LIST[<line number>]		44	41 101	* [<loop statements>]			
LIST[<line number> [-<line number>]]]		44	41 101	* WEND	83	65	206
LLIST[<line number>]	(46)	(42)	104	<b>PROGRAM INITIALIZATION</b>			
LLIST[<line number> [-<line number>]]]		46	42 104	CLEAR[, [<expression1>] [,<expression2>]]	6	18	22
MERGE<filename>		50	44 112	COMMON<list of variables>	9	20	30
NEW		53	46 117	DEF FN<name>[(<parameter list>)]			
REM or ' <remark>		73	58 163	=<function definition>	14	23	39
RENUM[[<new number> [,<old number>] [,<increment>]]]		74	59 164	DEFDBL<range(s) of letters>	16	24	42
				DEFINT " " "	16	24	42
				DEFSNG " " "	16	24	42
				DEFSTR " " "	16	24	42
				DIM<list of subscripted variables>	19	26	43
				OPTION BASE n			
				where n is 1 or 0	58	48	134



TABLE 2. (Continued) MS-BASIC COMMANDS AND STATEMENTS BY USAGE CATEGORY

	Page Ref.			Page Ref.		
	'82	'83	'85	'82	'83	'85
<b>DATA I/O</b>						
DATA<list of constants>	13	22	36			
INPUT[;][<"prompt string">;] <list of variables>	37	37	82			
INPUT#<file number>, <variable list>	39	38	84			
[LET]<variable>=<expression>	41	40	96			
LINE INPUT[;][<"prompt string">;] <string variable>	42	40	99			
LINE INPUT#<file number>, <string variable>	43	41	100			
LPRINT[<list of expressions>]	48	43	110			
LPRINT USING<string exp>; <list of expressions>	48	43	110			
NULL<integer expression>	54	46	**			
OUT I,J where I and J are integer expressions, range 0-255	59	49	135			
also UG 2-4						
PRINT[<list of expressions>]	61	49	149			
PRINT USING<string exp>; <list of expressions>	63	51	151			
PRINT#<filename>,[USING <string exp>;]<list of exps>	67	54	156			
READ<list of variables>	71	57	161			
RESTORE[<line number>]	75	60	167			
WAIT<port number>,[I,J] where I and J are integer expressions	82	65	205			
WIDTH[LPRINT]<integer exp>	84	66	207			
WRITE[<list of expressions>]	85	67	212			
WRITE#<file number>,<list of expressions>	86	67	213			
<b>DATA I/O (Random File)</b>	UG4-6	93	F-7			
FIELD[#]<file number>,<field width> AS <string var>	29	32	61			
GET[#]<file number> [,<record number>]	32	34	68			
LSET<string variable>= <string expression>	49	43	111			
PUT[#]<file number> [,<record number>]	69	56	F-9			
RSET<string variable>= <string expression>	49	43	111			
<b>PROGRAM LINKAGE</b>						
CALL<variable name> [( <argument list> )]				3	16	9
also UG 2-3				104		
CHAIN[MERGE]<filename> [,<line number exp>] [,ALL][,DELETE<range>]]				4	17	15
also see note p5						
DEF SEG[=<address>]	UG 2-1	25	41			
DEF USR[<digit>]=<integer exp>	17	25	48			
PEEK(I) (function)	3-16	80	146			
POKE I,J where I and J are integer expressions		60	49	146		
USR[<digit>][( <argument> )]	UG 2-5	108	195			
UG 5-7						
RG 3-23		25				
<b>DATA MANIPULATION</b>						
DEF SEG[=<address>]	UG 2-1	25	41			
MID\$( <string exp1> , n [ , m ] ) = <string exp2>		51	45	113		
PEEK(I) (function)	3-16	80	146			
POKE I,J where I and J are integer expressions		60	49	146		
RANDOMIZE[ <expression> ]	70	57	159			
SWAP<variable>,<variable>	80	63	188			
<b>ERROR HANDLING</b>						
ERROR<integer expression>	27	31	58			
IF ERR=<error code> THEN...	26	31	57			
IF ERL=<line number> THEN...	26	31	57			
ON ERROR GOTO<line number>	55	46	120			
RESUME	76	61	168			
RESUME [0]	76	61	168			
RESUME NEXT	76	61	168			
RESUME<line number>	76	61	168			
<b>Notation used in above formats:</b>						
CAPS Capital letters - portions of commands or statements that must be entered, exactly as shown.						
[ ] Square brackets - enclosed entry is optional.						
< > Angle brackets - user entered data.						
Vertical bar - separates two alternate items.						
... Ellipses - entry may be repeated as many times as needed.						
All other punctuation (commas, semi-commas, colons, slash marks, and equal signs) must be entered exactly as shown.						

NOTE: \*\* NULL is not available in VBASICA

TABLE 3. MS-BASIC FUNCTIONS BY USAGE CATEGORY  
 (and those of VBASICA common to MS-BASIC)  
 (See notation list and notes at end of table)

	Page Ref.			
	'82	'83	'85	
<b>MATHEMATICAL FUNCTIONS</b>				
ABS(X)	2	69	2	Absolute value of X
ATN(X)	3	70	3	Arctangent of X in radians
CDBL(X)	3	70	15	Convert X to double-precision number
CINT(X)	4	71	19	Convert X to integer by rounding
COS(X)	5	71	32	Cosine of X in radians
CSNG(X)	3	72	33	Convert X to single-precision number
EXP(X)	7	73	60	e to the power X
FIX(X)	7	74	64	Truncated integer part of X
INT(X)	12	77	87	Largest integer <=X
LOG(X)	14	78	109	Natural logarithm of X
RND[(X)]	18	81	171	Random number between 0 and 1
SGN(X)	18	81	177	1 for X>0; 0 for X=0; -1 for X<0
SIN(X)	19	82	180	Sine of X in radians
SQR(X)	20	83	184	Square root of X; X>=0
TAN(X)	22	84	190	Tangent of X in radians
<b>STRING FUNCTIONS</b>				
ASC(X\$)	2	69	3	Numerical value of ASCII code of 1st character of X\$
CHR\$(I)	4	70	18	ASCII code of I
CVI(<2-byte string>)	6	72	35	Convert string to integer
CVS(<4-byte string>)	6	72	35	Convert string to single-precision number
CVD(<8-byte string>)	6	72	35	Convert string to double-precision number
DATE\$	-	72	38	Date as string: mm-dd-yyyy
FRE(X\$)	8	74	67	Number of free bytes in string space
HEX\$(X)	8	75	77	Hexadecimal value of rounded value of decimal X
INKEY\$	9	75	80	One-character string read from keyboard
INPUT\$(X[, [#]Y)	10	76	85	String of X characters from keyboard or file number Y
INSTR([I,]X\$,Y\$)	11	76	86	Position of 1st match of Y\$ in string X\$, starting at Ith character
LEFT\$(X\$,I)	12	77	95	Leftmost I characters of X\$
LEN(X\$)	13	78	96	Length (number of characters) of X\$
MID\$(X\$,I[,J])	15	79	113	Middle J characters of X\$ starting with Ith character
MKI(<integer expression>)	15	79	115	Make string of integer
MKS(<single-precision expression>)	15	79	115	Make string of single-precision number
MKD(<double-precision expression>)	15	79	115	Make string of double-precision number
OCT\$(X)	16	80	117	Octal value of rounded value of decimal X
RIGHT\$(X\$,I)	17	80	*	Rightmost I characters of X\$
SPACE\$(X)	19	82	182	String of X spaces
STR\$(X)	21	83	186	String representation of X
STRING\$(I,J)	21	83	187	String of length I, all characters have ASCII code J
STRING\$(I,X\$)	21	83	187	String of length I, all characters are 1st character of X\$
TIME\$	-	85	191	Time as string: hh:mm:ss
VAL(X\$)	23	85	196	Numerical value of X\$

TABLE 3. (Continued) MS-BASIC FUNCTIONS BY USAGE CATEGORY  
 (and those of VBASICA common to MS-BASIC)  
 (See notation list and notes at end of table)

	Page Ref.				
	'82	'83	'85		
<b>MISCELLANEOUS FUNCTIONS</b>					
DATE	-	-	**	Numerical day of year	
EOF(<file number>)	6	73	54	Returns -1 (true) at end of sequential file	
FRE(0)	8	74	67	Number of bytes in memory not used by MSBASIC	
FRE(" ")	8	74	67	Forces garbage collection, then returns number of free bytes	
INP(I)	UG 2-4	9	75	81	Byte read from port I
LOC(<file number>)	13	78	105	Record number just read or written in random file	
LPOS(X)	14	78	109	Position of line printer print head in printer buffer	
PEEK(I)	16	80	146	Byte read from memory location I	
POS(I)	17	80	147	Cursor position (I=dummy)	
SPC(I)	20	82	183	Prints I spaces (only with PRINT and LPRINT)	
TAB(I)	22	84	189	Spaces to position I (only with PRINT and LPRINT)	
TIME	-	-	**	Numerical seconds after midnight	
USR[<digit>](X)	UG 2-5	23	85	195	Calls assembly language routine with argument X
VARPTR(<variable name>)	24	86	197	Address of first byte of <variable name>	
VARPTR(#<file number>)	24	86	197	Address of buffer assigned to <file number>	

Notation used in formats:

- CAPS Capital letters - portion of functions that must be entered exactly as shown.
- ( ) Parentheses - enclose arguments to functions.
- [ ] Square brackets - enclosed entry is optional.
- < > Angle brackets - user entered data.

All other punctuation must be entered exactly as shown.

X and Y Represent any numeric expressions.

I and J Represent integer expressions.

X\$ and Y\$ Represent string expressions.

Page references are to following Victor documentation:

'82 = "User's Guide for MS-BASIC", first Victor printing December, 1982 and "Reference Manual for MS-BASIC". These two parts were issued together in a looseleaf binder identified as "BASIC 86 User's Guide". Most page number references are to Chapter 3 of the Reference Manual. For brevity the 3- portion of these page numbers is omitted; other pages are identified as in the guide; UG stands for User's Guide.

'83 = "MS-BASIC", first Victor printing April, 1983. This manual, issued in a looseleaf binder also identified as "MS-BASIC", contains essentially the same material as the earlier manual but has a single set of consecutively numbered pages, I-VII and 1-122.

'85 = "VBASICA", first Victor printing August, 1985. This manual, issued in a looseleaf binder, contains essentially the same material as the above earlier manuals plus the addition of much material, not covered here, relating primarily to graphics and sound. Page numbering has reverted to separate series for each chapter and appendix. Unless otherwise indicated, page references are to Chapter 3; for brevity the 3- portion of these page numbers is omitted. (Only Rev. B and C have been reviewed.)

Notes: Double-precision functions are supported only by the MS-BASIC compiler.

\* RIGHT\$ is omitted in '85, but the function is available in VBASICA.

\*\* DATE and TIME are undocumented in '82 and '83; they are not available in VBASICA.

TABLE 4. VBASICA MANUAL ERRATA

Page	Item	Comment
3-103	LLIST Command	Format includes braces for no apparent reason. (undefined) (Other symbols are defined on page 3-1.)
3-117	NBW Command	Should also say: Closes all disk files.
—	PEEK Statement	Missing from Chapter 3, except for incidental mention on page 3-146 under POKE Statement. Missing from Contents and Index.
—	PUT Statement	Missing from Chapter 3, but appears on pages F-8 and F-9. Missing from Contents. Index reference to page 1-13 is only as one item of a list!
—	RIGHT\$ Function	Missing from Chapter 3 but mentioned or used on pages E-1 and F-5. Missing from Contents and Index.
Index	DEFDEL Statement	Should have 3-42 added.
Index	DEFINT Statement	" " " " .
Index	DEFSNG Statement	" " " " .
Index	DEFSTR Statement	" " " " .
Index	END Statement	Should read 3-50 instead of 3-53.
Index	EOF Function	" " 3-54 " " 3-55.
Index	ERASE Statement	" " 3-55 " " 3-54.
Index	file	Should read: protected, F-2 random, F-7 sequential, F-3

(Text continued from page 7)

### MULTIPLE BRANCHING DECISIONS

The next three programs show three different ways of handling a multiple branching decision problem: with multiple IF... THEN statements, with ON... GOTO, and with ON...

GOSUB switching to the multiple branches. As with the last set of programs, these three programs produce identical results (and for that reason the results of the runs are given only for the first program).

```

5 REM Multiple decision example with IF . . . THEN
statements
10 PRINT
20 INPUT "X ="; X
30 IF X>=1000 THEN PRINT "4 OR MORE DIGITS": GOTO 10
40 IF X<1 THEN PRINT "NO DIGITS": GOTO 10
50 IF X<10 THEN PRINT "1 DIGIT": GOTO 10
60 IF X<100 THEN PRINT "2 DIGITS": GOTO 10
70 PRINT "3 DIGITS": GOTO 10
Ok
RUN

```

```

X =? 1000
4 OR MORE DIGITS
X =? .9
NO DIGITS

```

```

X =? 8
1 DIGIT
X =? 77
2 DIGITS

```

```

X =? 666
3 DIGITS
X =? ^C
Break in 20
Ok

```

```

5 REM Multiple decision example with ON . . . GOTO
10 PRINT
20 INPUT "X ="; X
30 IF X>=1000 THEN PRINT "4 OR MORE DIGITS": GOTO 10
40 IF X<1 THEN PRINT "NO DIGITS": GOTO 10
50 ON .5+LOG(X)/LOG(10) GOTO 60,70,80
60 PRINT "1 DIGIT": GOTO 10
70 PRINT "2 DIGITS": GOTO 10
80 PRINT "3 DIGITS": GOTO 10

```

In line 50 the rounded (integer) value of the expression following ON determines the line number in the list after GOTO to which control is switched. Since MSBASIC does not include a function for logarithms to the base 10, the function LOG (to the base e) must be used. Hence the expression LOG(X)/LOG(10) is used. The constant .5 must be added, because of the rounding, to obtain the correct answer for all values of X. You can verify this result by the following runs:

```
X =? 9.999999
1 DIGIT
```

```
X =? 10.000000
2 DIGITS
```

```
X =? 9.9999999
2 DIGITS
```

(The incorrect answer for the last result is caused by the computational accuracy being limited to only about 7 digits, not to any error in the program.)

Making changes to lines 50 thru 80, the ON...GOSUB command is used. As a subroutine returns control to the line following that calling it, line 55 was added to cause control to switch back to line 10.

```
5 REM Multiple decision example with ON GOSUB
10 PRINT
20 INPUT "X ="; X
30 IF X>=1000 THEN PRINT "4 OR MORE DIGITS": GOTO 10
40 IF X<1 THEN PRINT "NO DIGITS": GOTO 10
50 ON .5+LOG(X)/LOG(10) GOSUB 60,70,80
55 GOTO 10
60 PRINT "1 DIGIT": RETURN
70 PRINT "2 DIGITS": RETURN
80 PRINT "3 DIGITS": RETURN
```

## LOOPS

The FOR...NEXT statement for executing loops was previously introduced. A fuller treatment of the subject is given here, adding forms using the IF...THEN, ON...GOTO, and WHILE...WEND statements. (Similarly, other trivially different forms, such as IF...THEN with statement rather than line number and ON...GOSUB, could be added.)

A loop is a means of repeating a set of commands a requisite number of times to obtain a desired result. It consists of five steps: initialization, operation (body of the loop), test for finish, incrementation, and return to start. In the program that follows the steps are shown in this order; however, the middle three steps may be performed in any order.

```
5 REM Summing numbers from 5 to 15
10 S=0: X=5 'Initialization
20 S=S+X 'Body of loop
30 IF X=15 THEN 60 'Test for finish
40 X=X+1 'Incrementation
50 GOTO 20 'Return to start
60 PRINT "S ="; S
70 END
Ok
```

```
RUN
S = 110
Ok
```

The four program switching commands, FOR, IF, ON, and WHILE can all be used to form loops. The use of IF has just been shown above. The following three programs illustrate the use of the remaining commands in loops.

```
5 REM Summing numbers from 5 to 15
10 S=0 'Initialization
15 FOR X=5 TO 15 'Loop initialization and
test for finish
20 S=S+X 'Body of loop
25 NEXT X 'Return to start
60 PRINT "S ="; S
70 END
```

```
5 REM Summing numbers from 5 to 15
10 S=0: X=5 'Initialization
15 ON 17-X GOTO 60 'Test for finish
20 S=S+X 'Body of loop
40 X=X+1 'Incrementation
50 GOTO 15 'Return to start
60 PRINT "S ="; S
70 END
```

This is a poor use of the ON...GOTO command (which is really intended for multiple branching) in that an awkward expression is required for the test for finish. (The expression can not be negative and must equal or round to a value of 1 for the desired switch to occur.)

```
5 REM Summing numbers from 5 to 15
10 S=0: X=5 'Initialization
15 WHILE X<16 'Test for finish
20 S=S+X 'Body of loop
40 X=X+1 'Incrementation
50 WEND 'Return to start
60 PRINT "S ="; S
70 END
```

The WHILE...WEND form differs from the IF...THEN form only in the test for finish expression and the return to start does not require that the start line be specifically identified by number.

Note that the FOR...NEXT form is the simplest to use in that it provides for initialization of the loop index, incrementation, and test for finish, thereby shortening the program. Also note that the body of the loop is indented. While indention is not necessary, it is a good practice to follow in that it makes loops easy to find.

The FOR...NEXT command can accommodate steps other than the implied 1 in the above example. Steps of 2, .5, and -1 are illustrated in the following three examples.

```
5 REM          Summing numbers from 5 to 15
10 S=0          'Initialization
15 FOR X=5 TO 15 STEP 2  'Loop initialization and
                        test for finish
20  S=S+X      'Body of loop
25 NEXT X      'Return to start
60 PRINT "S ="; S
70 END
Ok
RUN
S = 60
Ok
```

```
5 REM          Summing numbers from 5 to 15
10 S=0          'Initialization
15 FOR X=5 TO 15 STEP .5  'Loop initialization and
                        test for finish
20  S=S+X      'Body of loop
25 NEXT X      'Return to start
60 PRINT "S ="; S
70 END
Ok
RUN
S = 210
Ok
```

```
5 REM          Summing numbers from 5 to 15
10 S=0          'Initialization
15 FOR X=15 TO 5 STEP -1  'Loop initialization and
                        test for finish
20  S=S+X      'Body of loop
25 NEXT X      'Return to start
60 PRINT "S ="; S
70 END
Ok
RUN
S = 110
Ok
```

We have now completed the introduction by example of all the commands and statements in the program switching category. In the next article, example programs will be given showing the use of arrays and the use of sequential data files. This will complete the introduction of all commands and statements in the file operation category and most in the data I/O category.

\* \* \*

## RE: UPGRADING THE VICTOR 9000

A postcard from Marty Moleski says the following:

Enjoyed the latest VTalk (Vol. 5, No. 2, Mar-Apr 1987) very much; keep up the good work!

One technical correction in Habeck's article: putting an 8MHz NEC V20 into the V9 does NOT make the Victor run at 8MHz! The CPU swap leaves the built-in system clock unchanged, so the 8MHz chip gets pushed along at the normal V9 rate of 5MHz (like driving a Lamborghini at 55 mph instead of its top-rated speed). It is very highly unlikely that anyone will come up with any accelerating hardware for the V9. The clock chip is tied to a lot of other things besides the CPU, so drastic revisions would have to be made on the motherboard. The people who are smart enough to do this have found other things to do for more \$\$\$.

[Further technical details on a practical speed-up would be a worthwhile article for a future VictorTalk from Roger Habeck (or anyone else). - Ed.]

\* \* \*

## VALID MEMORY BOARD TROUBLES IN A PLUS PC / Wayne Crouch

I have still been totally unsuccessful in all attempts at getting my PLUS PC (256K) to work reliably with my VALID (+640K) memory board. I have tried all possible variants of the instructions provided by VALID - to no avail. The VALID board works fine with my system set up as a normal V9000 HD, but fails when I install the PLUS PC option. I have determined that what seems to be happening is that the DRAM refresh is being either intermittently blocked or delayed. The result is random "trashing" of memory. I suspect that the cause of this is that memory access timing between the VALID board and the PLUS PC is marginal. That would explain why it seems to work for some people, but not for others. I have paid VALID \$60 to "fix" the problem. They had no trouble cashing my check, but totally failed to fix the board. What's more, once they had my money, they won't answer my letters on the subject. . . . I don't really know what else to do now except ask for help from other readers. Has anyone out there got any suggestions?

Wayne Crouch  
6603 Foxhall Drive NW  
Huntsville, AL 35805  
205-837-9577 (Home)

\* \* \*

A means for accessing system time in dBASE II is described in Section 5.10 of the Supplementary Technical Reference Manual (for the Sirius 1 computer). Version 1.1 of this manual, published by Barson Computers Pty, Ltd, of Australia, is available on Victor\*Group Public Domain disk #47. The author, in attempting to run the program, found several errors requiring fixes to run properly. This article describes these fixes. (Ed.)

The author has prepared three files, which will be included in a forthcoming Public Domain disk:

- DTIME.DOC - Documentation comments, selected from the reference manual
- DTIME.COM - An executable file
- DTIME.ASM - Source code, permitting user modification if desired. This file, except for fixes described below, is that given in the reference manual.

The full description of the program, as given in the reference manual, follows:

ENTRY:BX - Pointer to length byte at start of string  
 - 11 byte (space) string passed from dBASE

EXIT: String is passed back to dBASE with time  
 CHANGE: All registers, destroyed, but dBASE will return machine's state when it regains control.

This program takes an 11 byte long string from dBASE and puts the MS-DOS time into it. The string must be of "character" type. If the string is not 11 bytes long, the routine is exited with no change.

By typing DTIME at the system prompt before entering dBASE, the first part of the program loads the second half at address 65024 decimal in the current program segment. (Actually, any location above A400H is okay. Further, the dBASE command load [filename] could load an assembly language routine within dBASE.) This address is then used inside dBASE as the argument for the SET CALL TO command. Because the 1/100th seconds clock in MS-DOS returns either 00 or 50 it was felt to be of limited use and was not included in this program. A 12 hour clock with an AM or PM tag at the end was employed for ease of use. If a 24 hour clock is desired the conversion code can easily be eliminated. A SORT routine might overwrite the time code if it is large enough. If this is a problem, the code can be modified to use INT 27H to create a protected

area in MS-DOS.

A common sequence of commands to get the time would be:

```
A>DTIME
A>DBASE
Copyright (C) 1982 RSP Inc.
*** dBASE 11/86 Ver 2.4 1 July 1983
. STORE "12345678901" TO TIME
12345678901
. SET CALL TO 65024
. CALL TIME
? TIME
5:23:00 PM
```

The code in Section 5.10 was fixed by these changes so that it would assemble:

```
Changed CODE SEGMENT PUBLIC 'CODE'
to CODE SEGMENT PUBLIC 'CODE'

Changed MOV CX,0A6H
to BEGIN: MOV CX,0A6H

Changed END
to END BEGIN
```

After the program assembled with linking errors, it would not run. It was fixed by the following procedure (adapted from Section 3.13 of the reference manual):

```
LINK will produce an error 'Warning: No STACK
segment'; ignor it. The file DTIME.EXE produced by
LINK will not run. You must use DEBUG to generate
DTIME.COM as follows:
DEBUG DTIME.EXE
N DTIME.COM
W
Q
```

While the resulting program DTIME.COM ran, the answer was wrong. The problem was corrected by the following change:

```
Changed MOV AL,C ;put minutes in AL
to MOV AL,CH ;put minutes in AL
```

All these fixes have been incorporated into the DTIME files referred to above. The above comment about possibly overwriting the time code is important; it happened to me for a file of 1400 records!

\* \* \*

# PLUS PC AND VI UPGRADES

Al Bullock, of Bullock and Associates [39522 Benavente Place, Fremont, CA 94539 (415) 651-6183], is offering upgrades for the Victor PLUS PC board and for the Victor VI computer. These upgrades are described as follows:

## Plus PC Upgrade

1. The following versions of the +PC Operating System are all down level and should not be utilized: V 3.00, V 3.01, V 3.02 and V 3.03. The new release is V 3.04.

2. New EPROMs are installed in the +PC DMA board (which occupies an expansion slot).

VER. .026, +PC ROM ASSY, ROM 0 LOCATION: IC U5C  
VER. .026, +PC ROM ASSY, ROM 1 LOCATION: IC U4C  
(If making the installation yourself, replace only one EPROM at a time to avoid misplacement of the two EPROMs.)

3. The upgrade provides the following product enhancements:

A. Corrects the Mode utility for correct serial port operation.

B. Provides a batch file which automatically sets up a fixed disk drive.

C. Increases the system's allocated space for the stack. Previously the stack area was not adequate for all jobs and the stack would overflow, causing system lockup.

D. Corrects various sound problems such as no bell in Victor mode and noise in I mode.

E. Corrects the NUM LOCK function. NUM LOCK now only toggles the 10 keypad key functions. The cursor keys will now function correctly either with the NUM LOCK on or off.

F. If an attempt was made by the operator to change the default drive to a drive which was not ready, the system would make a DOS call to get the directory name, causing an error which could not be corrected unless the drive could be forced into a ready state. The DOS call for the directory name has been delayed until invoked by a command from the operator (such as DIR).

G. Corrects an error which would occur when redirection was attempted while invoking COMMAND.COM. This facilitates nesting of batch files.

H. Corrects repeat key characters from keyboard to screen.

Cost of upgrade to user group members is \$50.00, including all hardware, software, and installation.

## VI Upgrade

1. The following versions of the VI Operating System are all down level and should not be utilized: V 3.03. The new release is V 3.04; new ROMs are ver. 13.

2. Current FCO levels for the VI PC boards are as follows:

Board	P/N	FCO Level
FLOPPY CONTROLLER	106740-01	D
HARD DISK CONTROLLER	106750-01	E
256K CPU BOARD	106520-01	E
512K CPU BOARD	106520-02	E

3. The upgrade provides the following product enhancements:

A. Updates the FDISK and FDSETUP utilities to handle more than one fixed disk drive, and fixed disk drives with capacities greater than 32 Mb.

B. Updates video stretch circuitry to fix I mode video "snow".

C. Updates video display circuitry to insure correct CPU instruction prefetch during echo cycles and to provide secret characters in I mode.

D. Corrects audio circuitry to prevent speaker noise when the power is first turned on.

E. Reduces hard and soft errors when accessing the floppy disk drive(s).

F. Updates to allow installation of VI 8087 option board.

G. Corrects head select problem when a read sectors from head 0 operation is followed sequentially by a read sectors from head 1 operation on I format diskettes.

H. Enables installation and utilization of popular software packages which utilize the SOFTGUARD SUPERLOK copy protection scheme.

I. New translator enhances the keyboard operation in the I mode.

J. Corrects repeat key characters from keyboard to screen.

Cost of upgrade to user group members is \$250.00, including all hardware, software, and installation.

\* \* \*



# FROM THE BOARDS

[FROM THE BOARDS is assembled from messages appearing on various (generally Victor-specific) bulletin boards, particularly The PUB, by John and Mary Jo Knobelsdorf. Only minor editing is done to correct obvious typographical errors and to eliminate some superfluous material. An effort has been made toward a degree of standardized formatting and the elimination of excess space to the extent practical without degrading legibility. Garbles that cannot be readily rectified are allowed to remain. Believing that considerable worthwhile material is contained in these excerpts, our intention is to include this feature in each issue of VictorTalk, as space will allow. - Ed.]

## #114 SYSOP, Subject: 40MB Atasi Drives

LLOYD LEBLANC asked for my thoughts on the Atasi 40MB hard disk drives. I've purchased six of those drives over the past two years. The first was the unit which The PUB ran on for almost a year and a half until the recent crash. I've rerun the low-level format and HDSETUP routines and now use that drive on a backup unit for non-critical stuff. It seems to "power down" for no good reason on occasion, but switching the power on and off brings it back up with no apparent damage to the files. This unit originally cost \$1200 installed.

I purchased one unit from Disk Drive Repair in Beaverton Oregon for \$500 and installed it in another HD V9000 CPU. I used that one for quite awhile before passing it on to a family member. I was pleased enough with that drive to try and purchase more from Disk Drive Repair. Numerous calls and a standing "back order" failed to secure another drive from that source, however. All units from Disk Drive Repair are used, reconditioned drives. Their response to telephone calls is indifferent at best.

I then purchased two units from Al Bullock and Associates in Fremont, Calif for \$600 each. Al installed one for me with no charge for the installation other than \$60 for new boot ROMs and the UPS charges for schlepping the CPU back and forth. The other unit I installed myself. Both units seem to be running o.k., and one of them contains the "post-crash" PUB files. Al's drives are also used, reconditioned units.

The only "new" drive I've seen is one I purchased from JB Technologies, Inc., 21011 Itasca St, Unit "F", Chatsworth, CA 91311, (818) 709-6400 or 884-8500, for \$475. This unit came with the original factory test print-out sheets and had obviously never been used. I installed this unit for a PUB member, and the low-level format, the HDSETUP, and the loading of the OS and numerous files were all accomplished

without incident. I was so impressed with this drive that I ordered another for my own use the very same day. The second unit, however, was a disaster. It was a (very) used unit with a badly bent frame. I will be calling JB Tech for a replacement unit tomorrow, and will report on their response. Their product return policy sheet is as complicated as a Microsoft copyright/warranty/disclaimer notice.

I like the Atasi 40MB units — notwithstanding the crash of the original PUB unit. Almost a year and a half of continuous 24 hour a day operation with literally thousands of file read and writes seems like fairly good performance to me.

p.s. JB Tech's most recent ad in The Computer Shopper lists the Atasi 3046 for both \$465 and \$475 (in the same ad). They also list Tandon TM 100-4's for \$59.95. Caveat emptor, however; PUB member reports on JB Tech are not all that favorable!

## #672 SYSOP, Subject: Message in a Bottle (More on SS to DS Drives)

Michael, if you wish to post my recent experience w/ drives you may do so. It was...

Pound-foolish Wells wished to convert one single-sided drive to double-sided in each of two recently-acquired floppy-drive V9's. Inspection told me that upgrading a Tandon TM103 to TM104 (ds drive) required no more than replacing the current upper foam bearing surface and it's plastic spring-loaded retainer with an upper head assembly intended for the TM104. The gizmos from each drive look nearly identical except the TM104 gizmo has, in place of the small circular foam pad, a read/write head with attached leads. Undoing two Allen screws (and keeping track of one small spring) would make way for the upper head assembly. The latter looked like an \$8 to \$10 item. Ha! I ended up buying used, as-is drives for parts. On the first try, no problem. Soldered in the requisite jumper on the controller board, installed the new head, and presto!, the machine performed all the usual double-sided functions. Except... it wouldn't read ds disks written elsewhere. No big deal again, though. Loosened the Allen screws, aligned the upper head by guess/trial-and-error, and w/in minutes the computer was working like a champ. Wonderful. I'm on my way. Ha! again. The next two head assemblies (the extent of my supply) were worn beyond use. Initially they dragged on the disk. No amount of adjusting spring tension or aligning would get the drive to format both sides. So, cutting corners didn't pay. Yes, I'd try the same thing again, but only with new (and inexpensive), V9-spec head assemblies. For now,

## SOME THOUGHTS ON MS-DOS 3.1 / Oliver Deex

I came to DOS 3.1 late and directly from DOS 1.25. Dale Cops of VictorCircle offered the new (and necessary) boot ROMs along with the DOS program for a very attractive price so I couldn't turn it down. I am not sure how much DOS 3.1 does for me or the programs I run, but it has been fun – and somewhat of a challenge. The fun comes from the fact that it is a good deal more powerful than DOS 1.25 and the challenge comes from the fact that it is poorly documented (and frequently even mis-documented).

Functionally, DOS 3.1 seems to be much like DOS 2.11 and anyone who is planning to "move up" should read the very excellent articles that appeared in VictorTalk. Reading the manual that comes with DOS 3.1 helps too, but be warned that many of the programs don't work the way the manual says they do – or even the way the Victor supplement implies.

The "Victor 9000 Supplementary Information" section is a good place to start. I don't have a hard disk so I ignored section 1.1 about AutoSet. (Perhaps someone else will cover that.)

### FORMAT

Sections 1.2 and 2.2 both deal with Format – poorly. DOS 3.1's FORMAT is a weak shadow of the DOS 1.25 version. The only switches available for the V 9000 are /m /c and /v. The best way to find out what FORMAT will do is type:

```
format ?<CR>
```

and read the help message. This tells you all the things FORMAT won't do with the Victor. The /c switch clears the allocation table (sort of a super "del \*.\*" if you have subdirectories), /v lets you put on a volume label (easier to do later with "vol/v") and /m puts some "messages" on the screen. These messages tell you what track is being formatted and a few other useless things but there seems to be no way, unless a "fatal" format error is encountered, to find out how good the disk is. When I want to count soft errors on an older disk, I boot up DOS 1.25 and run its format program with the /e switch. The disks work fine later with DOS 3.1.

Much has been said about the propensity of the higher DOS's to format the disk with the format program if one forgets to specify the other drive. DOS 3.1's FORMAT will format the default drive but at least it pauses, gives a message, and allows one to put in a new disk before it starts formatting. Nevertheless, proceed cautiously at first until you get used it.

### DISKCOPY

DISKCOPY, the program to copy diskettes, on the other hand, doesn't hesitate at all and will rapidly copy the wrong disk if called up with the wrong sequence. As one would expect for a program that can get you in trouble fast, DISKCOPY does not work as per the directions in the manual. The command is:

```
diskcopy L[R] to R[L] /sw
```

where L is the left or "A" drive and R is the right or "B" drive and sw is one of the familiar c, e, z, or s switches from DOS 1.25. This form of the command goes to work immediately and if you get your right and left mixed up, it will copy the wrong way.

If you want to take time and think, call it up with the command:

```
diskcopy /sw
```

and it will pause and ask which drive to copy from, left or right. Just don't mix up your right and your left after using A and B for all these years.

### SYS

SYS, the command to copy system tracks, also does not work as described on page 6-129 of the manual. It is called up with the command:

```
sys d:
```

but the drive letter stands for the the source drive and the program then prompts for the destination drive. It also will copy COMMAND.COM and a few other of the system files if it can find them. The way I have things set up, with AUTOEXEC.BAT instead of CONFIG.BAT (so I don't have to enter the time and date), SYS never seems to find anything past COMMAND.COM and flashes all kinds of error messages on the screen. However, it does get the hidden MSDOS.SYS file copied (there is no IO.SYS file, the manual notwithstanding) and the rest can be copied individually.

### CONFIG

The CONFIG.SYS file works just like DOS 2.11 and needs no further discussion. I have found that the assembly language PPORT.SYS and INITSIO.COM files available on PD #60 save memory and load much faster than the longer .EXE files furnished with DOS 3.1. In general, most of the utility programs furnished with DOS 3.1 are .EXE files and are longer than their .COM counterparts of DOS 1.25. One thing I have observed is that the serial ports need to be re-initialized (i.e. INITSIO.COM must be rerun) every time I quit Crosstalk or

the program will not work the next time it is called up. I haven't figured out what is happening but everything works fine if I use the batch file that calls Crosstalk and disables the scrnsave option (SCRNSAVE.COM is on PD #44) to also rerun INITSIO.COM. VICMINI does not require reinitialization.

## PRINT

One thing that is very obvious about DOS 3.1 is its slowness in the printing department. It is not only slow with word processing programs but also with the Ctl-P printer toggle and a DIR command. And slow as it is, it doesn't seem to care if there is a printer on line or not. Whereas 1.25 will hang up if the parallel printer is turned off when a print command is given, 3.1 sends the bytes right out and if the printer buffer can't get them all, they are lost. This can be avoided by using a print spooler program or printing files with PRINT.EXE (which is essentially 3.1's spooler). PRINT.EXE is a pretty good program except that it sends a form feed after every file and thus wastes a piece of paper between each file if you are printing fully formatted files from a word processor. It also sends out a form feed when it is terminated early with the /t switch (see the manual - it is fairly good for this utility). I managed to get rid of the form feed after every file by altering PRINT.EXE with DEBUG but I haven't figured out how to keep it from sending out the form feed upon early termination. (If anyone wants the directions for removing the first form feed, I will be glad to furnish them.)

## COMMAND LINE EDITING AND CONTROL CHARACTER FUNCTIONS

The command line editing key identifications given in the manual are all wrong. Dale Copps published a partial list of the correct assignments on VictorCircle NewsDisk 17. Here is my update (definitions from page 5-4 of the "MS-DOS 3.1 Reference"):

- F1 (or F9) Enters/exits insert mode
- F2 Makes the new line the new template
- F3 Copies one character from the template to the command line
- F4 Skips over (does not copy) a character in the template
- F5 Skips over (does not copy) the characters in the template up to the character specified
- F6 Copies characters up to the character specified in the

template and puts these characters on the command line

- F7 Copies all remaining characters in the template to the command line

Ctl-C/Ctl-X Cancels the current input; leaves template unchanged

The F1/F9 setup is probably for those who don't have the last three function keys. F2 prints an @ sign at the end of the line and moves the cursor to the beginning of the next line. F7 will then bring back what was entered before the @. I have not found any use for this command, but then, except for repeating very lengthy command lines, F7 and a little backspacing is all I usually use. It is normally faster to retype than it is to stop to figure out which function key does what. Ctl-C (Alt-C on the standard Victor keyboard) enters a ^C on the screen and gives a new DOS prompt; Ctl-X enters a \ on the screen and puts the cursor at the beginning of the next line. In both cases DOS is ready for a new command.

This last is covered on page 5-2 of the manual, again with the wrong keys. The printer toggle for the Victor is Ctl-P, although Ctl-N will turn the printer off as it did with DOS 1.25. Screen dump with the Victor requires the external command SCRNM80 (Epson and clones) or SCRNM84 (Okidata 84). These are adequately documented but I find I cannot use the screen dump with the cache memory/ramdisk program that came with my Dataflex clock-memory board.

## PATH

Although the PATH command and the ability to label each disk are handy, the directory/subdirectory capabilities are really not very necessary with single sided floppies. Shuffling directories and files seems to require an inordinate amount of copying and re-copying with its attendant wear and tear on the disk drives and I find it about as easy to use and rename extensions for filing purposes.

## SEARCH

If you do use directories, however, and back your work every day (as you should) then SEARCH.EXE is a really useful program. It is complicated and can get one into a lot of trouble fast but it is very powerful and it is well documented in the manual. I have only three suggestions for anyone who wants to try it: First, read the manual carefully - several times. Second, make liberal use of the /P switch so the program will ask yes or no. Third, run it with a copy of your file disk and a freshly formatted receiving disk until it does what you want. Once you get it going it is very easy to back up all the files you have worked on during the day and to purge

all the .BAK files you no longer want.

## CUSTOM KEYBOARD

The most fun I have had with DOS 3.1, however, was putting in a custom keyboard file. The MSDOS.SYS file for 3.1 is 130C0 (78,016) bytes long. Although a new version of DEBUG is furnished with DOS 3.1, like the old version it will only show the first 10000 (65,536) bytes. The business part of the keyboard file begins at 12A80 (75936). (From now on the byte count will be in hexadecimal only, since this is what DEBUG uses.) Thus, it can't even be found in the normal search which only covers the first 10000 bytes. I don't know if my solution to the problem is the best one, but it worked and I now have a custom keyboard file installed that is reasonably easy to change if required. Credit for the general method must be given to Franz Hirner who outlined it and described the whole operation in the first issue of "MicroVision" (Vol.1, No.1, Jan. 1985, p 21). I see by the issue of VictorTalk that just arrived that Jim Hasteley has done the same for 1.25/2.6 and 2.11/2.9. My directions are for DOS 3.1/3.02 and the steps are as follows:

1) Copy the system tracks (i.e. the files MSDOS.SYS and COMMAND.COM) to a fresh disk. ALWAYS work with a copy on a disk you can afford to mess up.

2) Un-hide the hidden file with ALTER.EXE (on PD #25) with the command:

```
alter b:msdos.sys<CR>
```

ALTER will give the message that the attributes are RHSB as ask for new attributes. Type n<CR>. This changes the attributes to B. (See ALTER.DOC for a full explanation.)

3) Call up DEBUG.EXE with the command:  
debug<CR>

(Note that the new version of DEBUG, which is on the first DOS 3.1 disk, uses a hyphen prompt instead of the old right angle bracket.)

4) Next we are going to "name" the system file, load it, and write it back out as two shorter files that will fit into DEBUG. The sequence is as follows (<CR>'s are left out for clarity):

```
-rmsdos.sys      ;"names" the file
-lcs:100         ;loads it into debug
-rbx            ;look at the BX register
BX 0001         ;debug gives current value
:0000           ;enter desired value after
                ; the :
-rcx            ;look at the CX register
CX 30C0         ;current value
:C000           ;enter the bytes for the
                ; first part
```

```
-norig1         ;enter a name for the first
                ; part
-wcs:100        ;writes C000 bytes starting
                ; at 0100
Writing C000 bytes ;DEBUG gives this message
-rcx           ;change the CX register
CX 30C0        ; again to write the
                ; remaining 70C0 bytes
:70C0         ; (130C0-C000=70C0)
-norig2        ;enter a name for the second
                ; part
-wcs:c100      ;write starting from
                ; C000+100=C100 (remember
                ; the first C000 bytes were
                ; loaded starting at 0100)
Writing 70C0 bytes ;DEBUG message
-q            ;quit (let's take a look)
```

5) Note that drive designations have not been used. If the disk with the MSDOS.SYS file is in the B drive and it is the current drive then one would call ALTER and DEBUG with the a: designation (or use the path=a: internal command). It is not really necessary to leave debug at this point but on the first time through this is a good point to check on things. At this point a DIR command should show MSDOS.SYS at 78016 bytes, ORIG1 at 49152 bytes and ORIG2 at 28864 bytes (now we are back in decimal because that's what DIR shows).

6) If you haven't looked at a keyboard file in debug, this should be done using the method of step 4. Entering d<CR> twice at the first prompt after loading the file should display the first 100 (hex) bytes. Beginning at 180 (80 hex bytes in-remember the 100 offset) is the sequence AA 04 00, which is the real beginning of the file (the rest is just header). This is the sequence we must find in MSDOS.SYS to locate the original keyboard file we want to replace.

7) If you do a hex dump of the entire MSDOS.SYS file you will find the sequence AA 04 00 occurs at 128A0. In the ORIG2 file this sequence will be at 68A0 (128A0-C000=68A0). This should be checked, however, so call up debug again and look.

```
-norig2         ;"names" the file ORIG2
-lcs:100        ;loads it into debug
-scs:10010 aa 04 00 ;search all memory for the
nmm:68A0        ; location of the sequence
-d6800         ;look at this area, should
                ; be 00's until 68A0
-d             ;this should show the
                ; AA 04 00 seq
-nnew.kb       ;name the kb file
-lcs:6820      ;load (overlay) it (remember
                ; the 80-byte header will
```

```

; go in too)
-d6800 ;look again, now the header
; will be there
-d ;this should show the
; AA 04 00 seq
-fcs:6820180 00 ;replace the header with
; 00's

```

8) At this point we have the new keyboard file in but the banner (when the computer boots up) will still say "Standard 01" or something like that. DEBUG will show this message at 6EC5 in the ORIG2 file. It can be located with debug's search command -scs:10010 "board", which will find the word keyboard at 6EB2. (In case the edition of DOS 3.1 you are working with doesn't say Standard but has some other keyboard.) To change the keyboard name, look at it and then use the edit command as follows:

```

-d6e80 ;to display the area to
change
-e6ec5 ;to change "Standard 01" to
53.43 74.75 ; "Custom NEW"
61.73 6E.74 64.6f 61.6d 72.20 64.20 20.4e 30.45
31.57

```

9) Now everything is changed and you are ready to write the changed second part out to disk.

```

-rcx ;to reset the cx register
CX 052C ;(when kb file is loaded, cx
:70c0 ;is set to length of new.kb)
-nnew2 ;name for write out
-wcs:100 ;write out from the
beginning
Writing 70CD bytes ;message for the write
-q ;exit debug

```

10) Now a dir command should show a new 28864 byte file called NEW2. All that remains is to combine ORIG1 and NEW2 and replace MSDOS.SYS with the result. This is done with the COPY command:

```
copy /b orig1 + new2 newdos<CR>
```

Now we have a 78016 byte file that will boot the system with NEW.KB and say "Custom NEW" on the banner.

11) At this point it is best to put the new file on a boot disk that also has the config.sys, autoexec.bat pport.sys, etc. files on it. So, copy an old boot disk, change the "modcon new.kb" line in the autoexec.bat file since it is no longer needed, and then unhide MSDOS.SYS with ALTER, just as before. Next the command:

```
copy newdos a:msdos.sys<CR>
```

is given and the new boot disk is ready to go. It is not

necessary to hide MSDOS.SYS. The disk will work fine with it unhidden. It could be erased accidentally, but you have the work disk from which you copied it and you will back the new boot disk so the danger is not great.

12) By saving the work disk with ORIG2 on it, it is very easy to change the keyboard for other programs or to revise it (if you have KEYGEN but didn't want to spring for SYSELECT).

13) I have not tried to change the character file in DOS 3.1. The "Int'l 01" file that is installed can be retrieved as DOS.CHR by entering the command:

```
modcon * dos.chr<CR>
```

It has the sequence 30 30 36 34 (..0064..) at nnnn:0157 and the sequence FC 00 02 01 4A at nnnn:01A4. The latter sequence is found at nnnn:1C28 in ORIG1 (and, of course, MSDOS.SYS). I don't know if the 4-byte shift is significant or not.

\* \* \*

#### FROM THE BOARDS (Continued from page 17)

daughter #2 will work with a ss V9 — no real hardship.

- - -

Another tidbit, likely not news to you. I found an old Metasoft 800 number and gave it a try, to find out about updating a 3.0, rev B version of Benchmark word processor (came as part of the deal on my Mich. V9's). Bill Kroll at Metasoft referred me to Mike Penna at "Technically Yours" in CA (408- 688-5201). Technically Yours turns out to be an outlet for a lot of stuff, including a spanking new incarnation of Benchmark (\$79), Report Manager (\$125), Cross Talk (\$125), etc. They/he/it also distribute Microvision as part of Technically Yours membership (I) benefits. The hitch, then, is that \$29.95 "memberships" are required to cash in on advertised prices (add 10% otherwise; Bay area V9 folks just worked a group deal of some sort). But Penna... familiar-sounding name. Nice guy. Was at Victor, he says, for ages, even before the sale to Kidde. I checked PSU and found his name on p. 315, at the end of the infamous "rumor monger" letter. Tired of being caught between manufacturer and user, Penna says he's now out to help owners of orphaned computers. Interesting.

Two phone calls later, still no V2 board from Affiliated. My gad, they're slow. They claim the problem is with Western Automation, which supplied a bunch of bad boot prompts. Still I'd think twice before relying on AFG again (even if they don't sell memberships).

\* \* \*

# BAVUG RBBS / Franz Hirner, Sysop

March was a busy month on BAVUG. There were 499 calls during the month. Interesting, on average you are moving up to more sophisticated equipment. There are few 300 baud callers any more and there are several 2400 baud callers. That shows what happens when volume brings the prices down. All callers are welcome on BAVUG. The 300 baud callers will soon move up to at least 1200 baud modems now that prices have fallen below \$100.00. If you're not yet a regular caller, think about getting on line.

The program for DSKBENCH was left out of last month's article. It is posted on BAVUG and may be downloaded by callers. The DSKBENCH listing was too long to fit into the small space available. I imagine that it will be picked up and placed on one of the PD disks also.

Microsoft Multiplan 3.0 has been ported to the Victor 9000 by several users. If you are using the older Multiplan Version 1.x you should look into this new release. Among other features, Multiplan will now read and write Lotus 1.A and Lotus 2.0 files, in addition to the normal Multiplan format. Calculation speed is much faster than before and the memory restrictions found in earlier versions have been eliminated. If you're a Multiplan user you should consider an upgrade. One of our callers is working on a Graphics package to support Multiplan 3.0 or any other program that will produce SLYK files.

Several V9 users also are subscribers to one of the national communication services such as EasyLink, MCI or the Source. Until now, these systems only allowed you to transmit pure text messages. There was no way to compress data to save transmission time nor to encode sensitive data. John Navas has come to the rescue with his program ECHOARC. ECHOARC allows the user to use either ARC or PKARC to either compress or encrypt portions of messages transmitted via public mediums. The password feature of PKARC or ARC provides security for your sensitive information being transmitted to the home office.

ECHOARC is a C filter program which converts ARC files into ASCII text which can be embedded into an echomail message — so pack whatever you want into an ARChive and you can send it by echomail — or any other form of ASCII text transmission!

Here's how it works — say you want to send a small program and some documentation. First you ARC them:  
ARC A PACKET.ARC PROGRAM.COM PROGRAM.DOC  
Then you use ECHOARC with I/O redirection to convert the

ARChive into transmittable text:

```
ECHOARC <PACKET.ARC >PACKET.TXT  
Now you can transmit PACKET.TXT. At the other end, the receiver uses ECHOARC to convert the text back into an ARChive:
```

```
ECHOARC <PACKET.TXT >PACKET.ARC  
And, finally, the receiver uses ARC to recover your files:  
ARC E PACKET.ARC
```

ECHOARC automatically knows the type of conversion to be performed by examining the input data. ARC provides both data compression and CRC checking. The C source code as well as a BASIC version of the program follows.

## ECHOARC C LISTING (MSC 4.0)

```
/ FILTER TO CONVERT ARC FILES TO/FROM TEXT FORMAT /  
#define BANNER "ECHOARC filter v1.1 Copyright 1987  
John Navas 11\n"  
  
#include <fcntl.h>  
#include <io.h>  
#include <stdio.h>  
main()  
{ register unsigned i, j, k;  
  fputs( BANNER, stderr );  
  setmode( fileno( stdin ), (int) O_BINARY );  
  setmode( fileno( stdout ), (int) O_BINARY );  
  i = ungetc( fgetchar(), stdin );  
  if ( i == 0x1A ) {  
    puts( "echoarc{\r" );  
    for ( j = k = 0; ! feof( stdin ); )  
      do { i != fgetchar() << j; j += 8;  
        for ( ; j >= 6; i >>= 6, j -= 6 ) {  
          for ( ; k++ >= 72; k = 0 ) puts  
            ( "\r" );  
          fputc( (char) ( i & 0x3F ) +  
            '!' );  
        }  
      } while ( j );  
    puts( "\r\n}echoarc\r" );  
  }  
  else for ( i = j = 0; ! feof( stdin ); )  
    if ( ( k = fgetchar() ) >= '!' && k < '!' +  
      64 )  
      for ( i != k - '!' << j, j += 6; j >= 8;  
        i >>= 8, j -= 8 )  
        fputc( (char) i );  
  return 0;  
}
```

(Continued on page 24)

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
P.O. Box 50893  
Palo Alto, CA 94303

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
City _____ State _____ & Zip _____	_____
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. Alternatively, see VictorTalk, Vol. 4, No. 2 for a brief description of disks 61-64; Vol. 4, No. 3 for disks 65-71; Vol. 4, No. 6 for disks 72-77; Vol. 5, No. 1 for disks 78-81; and Vol. 5, No. 2 for disks 82-84.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40	
														*(34 & 35 are a two-disk set)						
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56*	57*	--	59	60	
																	*(56 & 57 are a two-disk set)			
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	
81	82	83	84	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

1987 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, set of 6, \$12.00 ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_

1987 membership fee \$35 (receive all 1987 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.)..... \$ \_\_\_\_\_

BAVUG RBBS (Continued from page 22)

ECHOARC BASIC LISTING

```
100 PRINT "CREATING ECHOARC.ARC..."
102 OPEN "ECHOARC.ARC" AS #1 LEN = 1: FIELD #1,1
AS A$: R% = 0: S% = 1
104 READ B$: PRINT B$: L%=LEN(B$): FOR I%=1 TO L%
108 R%=R%+(S%*(ASC(MID$(B$,I%,1))-48)): S%=S%*64
110 WHILE(S%>=256): LSET A$=CHR$(R% MOD 256): PUT
#1
112 R%=FIX(R%/256): S%=S%/256: WEND: NEXT I%
114 IF(RIGHT$(B$,1)<>"!") THEN GOTO 104: ELSE
SYSTEM
116 DATA "JP@A3QdC19e@^D4F5109381000W3m=
EghfH50003=A;=:0;006440@H0hOok9"
118 DATA "10@640@00T80N8R100@jOJTNEIA\K@\
c0Ga:h2B64:hb600:X3M1H@450h200"
120 DATA "2:Pj@00C^TB2@VddVhIAkdUSjWc4G@d
U^90^LK8U0;2N3bOJX>]1GXYUXcOaT"
122 DATA "H2H50;A6AO;2O[k8XC91HA?KJPK1DmH
8HM^0GbQ[Mk[2hHUEP;Z\o[\m090H>"
.
. (About 100 DATA lines omitted - Ed.)
.
318 DATA "g\L@M0L0e400h@UM0Pe0L6TP7P60loo!"
```

As usual, the listings can be downloaded from BAVUG.

Our "C" Graphics package still is "about done". Documentation needs to be written and a few routines cleaned up a bit.

As always, your comments and suggestions are welcome.

\* \* \*

VictorTalk

Address inquiries to: P. O. Box 50893  
Palo Alto, CA 94303

Subscriptions: Nancy Weaver

Editor: Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

Contributors: Al Bullock, Wayne Crouch, Oliver Deex, Franz Hirner, John Knobelsdorf, Mary Jo Knobelsdorf, Martin Moleski, Jack Prichard, Steve Venitsky.

VictorTalk is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in VictorTalk. Copyright 1987 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

VictorTalk is an independent publication not affiliated in any way with Victor Technologies, Inc. Victor is a registered trademark of Victor Technologies, Inc.

Submit Articles on MSDOS disks in ASCII or Spellbinder format, with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

Yes, I want to join.

Enclosed is a check/money order for \$35.00 (foreign membership, \$40 in U.S. dollars) for my 1987 Victor\*Group membership.

Name \_\_\_\_\_  
Title \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_

Make checks payable to:

VictorTalk

P. O. Box 50893  
Palo Alto, CA 94303

City \_\_\_\_\_  
State \_\_\_\_\_ ZIP \_\_\_\_\_  
Country \_\_\_\_\_  
Phone \_\_\_\_\_



# VICTOR® TALK

VictorTalk Magazine for Victor 8000 users

VictorTalk, Vol. 3, No. 4, July-Aug 1987

## CONFIGSYS AND AUTOEXEC.BAT, by Franz Hirner, Sysop

An article on writing CONFIGSYS and AUTOEXEC.BAT files for the Victor 8000. The article explains the purpose of these files and provides instructions on how to create them. It also discusses the importance of backing up these files before making any changes.

The article continues with a detailed explanation of the CONFIGSYS file, including the various options available and how they affect the system's operation. It also covers the AUTOEXEC.BAT file, which is used to execute commands when the system boots up.

The author provides a step-by-step guide to creating and editing these files, including examples of code and commands. The article is intended for users who want to customize their Victor 8000 system and improve its performance.

**VICTOR\*GROUP**  
**P.O. BOX 50893**  
**Palo Alto, CA. 94303**

The article also discusses the importance of testing the files before using them. It provides instructions on how to boot the system in safe mode and how to load the files. The author also provides a list of resources for further information.

The article concludes with a summary of the key points and a final note from the author. The author expresses his hope that the article will be helpful to readers and encourages them to share their own experiences.

The article is a comprehensive guide to writing CONFIGSYS and AUTOEXEC.BAT files for the Victor 8000. It covers all the important details and provides clear instructions and examples. The article is a valuable resource for any Victor 8000 user who wants to customize their system.

The article is a well-written and informative piece that provides a clear and concise explanation of the CONFIGSYS and AUTOEXEC.BAT files. The author's use of examples and step-by-step instructions makes the article easy to follow and understand. The article is a must-read for any Victor 8000 user who wants to take their system to the next level.

**FIRST CLASS MAIL**

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
P.O. Box 50893  
Palo Alto, CA 94303

Date \_\_\_\_\_ Office use only:  
Name \_\_\_\_\_ Order rec'd \_\_\_\_\_  
Company \_\_\_\_\_ Member check \_\_\_\_\_  
Address \_\_\_\_\_ Comments \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_  
& Zip \_\_\_\_\_  
Country \_\_\_\_\_ Phone \_\_\_\_\_ Order shipped \_\_\_\_\_

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. Alternatively, see VictorTalk, Vol. 4, No. 2 for a brief description of disks 61-64; Vol. 4, No. 3 for disks 65-71; Vol. 4, No. 6 for disks 72-77; Vol. 5, No. 1 for disks 78-81; Vol. 5, No. 2 for disks 82-84; and Vol. 5, No. 4 for disks 85-86.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
21 22 23 24 25 26 27 28 29 30 31 32 33 34\* 35\* 36 37 38 39 40  
\*(34 & 35 are a two-disk set)  
41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56\* 57\* -- 59 60  
\*(56 & 57 are a two-disk set)  
61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80  
81 82 83 84 85 86

1987 Victor\*Group member: No \_\_\_\_\_ Yes \_\_\_\_\_ Date joined/renewed \_\_\_\_\_  
Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_  
Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_  
Postage/handling (applies to disk orders only)..... \$ 4.00  
Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_  
VictorTalk back issues: 1986, set of 6, \$12.00 ..... \$ \_\_\_\_\_  
1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_  
1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_  
1987 membership fee \$35 (receive all 1987 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)  
TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

## CONFIG.SYS AND AUTOEXEC.BAT / Franz Hirner, Sysop

An excellent file explaining CONFIG.SYS and AUTOEXEC.BAT was uploaded to BAVUG. The original was prepared by Barry Simon for the IBM. I have rewritten modified portions to properly describe the Victor 9000 CONFIG and AUTOEXEC files and present the revised article for your information.

During the startup of a MS-DOS or PC-DOS compatible computer, the operating system looks for two files of user supplied commands which allow you to customize your system in various ways. This article will attempt to explain some of the options available to you when you make these basic files. While I have written it for the relative novice, I hope it may provide some useful new tricks even to the more experienced user.

These two files, whose names must be CONFIG.SYS and AUTOEXEC.BAT, should reside in the root directory of your boot diskette or of your hard disk if you have a bootable hard disk (i.e. if you can start your system from your hard disk without placing a diskette in drive A). Actually, there is a method for placing the AUTOEXEC.BAT file in a subdirectory but despite the fact that I tend to be fanatical about keeping my root directory lean, I don't recommend using this option.

When you turn on your computer, the CPU is setup to begin a program in the ROM (read only memory) that all computers have. This ROM is distinct from and in addition to the working RAM (random access memory). When you are sold a machine with 256K of memory, that figure refers to the RAM. There is an additional ROM located in high memory. RAM is cleared whenever you turn off your machine or reboot while the ROM is permanently burned in and should not change.

### Booting Up

The program that is automatically run from ROM begins with a brief test of various components of your computer. During this operation you will see the "M2000" or similar text on the screen which indicates how much memory is in the machine.

After this test, the machine searches for various "ROM extensions" that is additional ROM that can come with a hard disk or clock card for example. The program then searches first drive A and then a hard disk if you have a bootable hard disk for a diskette or disk program to transfer control to. It transfers control to the very first sector on the disk which is called the boot sector. When you format a diskette, a little program is placed in the boot sector which will display the message "non system disk, replace and hit any key". When

you transfer the operating system to the disk with the SYS command or via FORMAT/S, this boot sector program is changed to transfer control to a program which must be in the position immediately following the boot sector. The exact layout of the boot sector is contained in the Victor Technical Reference Manual or may be downloaded from the Victor Technical File Area on BAVUG.

If the disk has the system on it, control is transferred successively to a hidden file MSDOS.SYS. MSDOS.SYS is the equivalent of the two IBM files IBMBIOS.COM and IBMDOS.COM, which may be familiar to you. When most users think of DOS they think of the familiar prompt and copy,.... commands. These parts of the DOS are only loaded later; the part in the hidden file involves services provided by DOS to programmers rather than directly to users.

Parenthetically, note that there isn't really much hidden about hidden files. As you may know, the DIRectory command gets its information from a special file, also called the directory. This file is essentially a little database with information about each file, including the filename, extension, date and time of creation. One byte in the record for each file is called the attribute byte and contains eight "switches" to keep track of things like whether the file is a volume label, read only, etc. One of these switches is whether the file is "hidden". To anyone with any programming experience or with any of a large number of public domain or commercial programs, these files are not in any sense hidden. ALTER or DISKTOOLS will allow you to change the file attribute of MSDOS.SYS so that you can examine it. Make a copy first for safety. The basic DOS services like DIR and COPY are specially set up to ignore hidden files and that is the only sense in which these files are hidden. The system files are hidden because their location is critical for a successful boot-up and they are less likely to be moved by accident if they are hidden.

After MSDOS is loaded, it looks for a special file called "CONFIG.SYS" and processes the commands in it. Then control is passed to the second file in the operating system, COMMAND.COM. As the final step in booting up, COMMAND.COM looks for a file named AUTOEXEC.BAT and, if found, loads and runs it. If not found, COMMAND.COM exits with the DATE and TIME commands.

Except for its special status as a bootup file, the AUTOEXEC.BAT file is an ordinary BATch file with the usual rules of syntax. The CONFIG.SYS file has a special syntax with a limited number of allowed commands. Both must be pure ASCII files, that is, without any special formatting codes

that some word processors add. Many word processors which have special codes have a "non-document" mode for preparing ASCII files. These files have separate lines which must be ended with carriage return/line feed pairs. If you have any doubts about whether the file is "pure ASCII" you can use the TYPE command to display it on the screen and see if it just has ordinary letters and numbers.

### What Goes In Your Root Directory

When a subdirectory fills up, it adds another cluster of disk space to increase its size but the size of the root directory is fixed at the time the diskette or disk is formatted. It is not merely because of the size restriction that I recommend that you keep your root directory slight. Since the files in the root are likely to be of diverse type, it will be difficult to keep track of things if you put too much there. I mainly put subdirectories there and mainly subdirectories which have no files but only subsubdirectories. For example, I have a words subdirectory with my word processor, outliner, thesaurus, etc in subsubdirectories. Generally, there are only three files that I recommend go into the root: COMMAND.COM, CONFIG.SYS and AUTOEXEC.BAT. As I mentioned, one can put AUTOEXEC.BAT elsewhere and even COMMAND.COM but I feel that is carrying things too far. In fact, I even have a startup.bat file of the type I'll describe there but the point is to keep that directory thin and to complain bitterly about software so inconsiderate that it forces you to place it in the root directory. My point in mentioning this here is that I'm about to discuss device drivers which many people put in the root directory. If you like to be organized, I recommend you make a directory for device drivers (mine is called \bin\devices). Another option is to put the drivers in different directories with each driver in with related files. For example, the drivers that come with DOS are kept in the same directory as the other DOS programs or the mouse driver is kept with the other mouse software.

### Device Drivers

There is a group of programs which is made permanently resident and which is loaded as part of the CONFIG.SYS. Virtually any resident program can be produced in this format but certain ones must be of this form. Typically, console drivers and any program which controls "a device" must be loaded now. Most virtual disks and print spoolers also are loaded as device drivers. While device drivers are programs, they need not have the extension ".com" or ".exe". In fact, so far as I can tell, their extension can be anything that you wish. Nevertheless, virtually all commercially available device drivers have the extension ".sys". Some drivers are available with the extension ".dev". The syntax for loading a device driver in your CONFIG.SYS is

```
device=<path><name><parameters>
```

so if you have a device foo.sys in the directory \bin\devices of drive C: and it will take a numeric parameter to set the size of some buffer, you might load it with

```
device=C:\bin\devices\foo.sys 128
```

Note that it is essential to include the extension ".sys" or else you will get an error message "bad or missing foo". The drive letter C: is not required but it can't hurt and I know of one person who claimed the device driver on her machine couldn't be found without it. The question of which parameters a given device driver allows or whether it allows any at all depends on the driver and should be dealt with in the documentation for the program in question. For the drivers ANSI.SYS and VDISK.SYS which come with DOS, I note that the former takes no parameters while the later takes parameters explained in the DOS manual.

IBM DOS 3.2 comes with a third driver called DRIVER.SYS while some versions of MSDOS 3.2 come with an alternate ram disk called RAMDRIVE.SYS. Both take parameters.

### Examples of Device Drivers: The Default Drivers

I will not attempt to describe all available device drivers since there are so many. For example, Chris Dunford, one of the sysops of Comuserve has written public domain programs which installs "devices" to control screen blanking (BURNDEV) and another allowing you to send control sequences easily to your speaker (SPKR). These represent examples where a real "device" is installed. A device is a virtual file which can typically be written to and read from. The most common example is "con" which you typically read from when you issue the command "copy con filename". Devices can only be installed via the CONFIG.SYS. Despite the name, the device command can load other programs which do not control devices and physical "devices" may not be devices in the sense of setting up a virtual file. A mouse is a good example of something which is not a device in this technical sense.

The hidden file MSDOS sets up several devices, even if you have no CONFIG.SYS: con, prn, aux. Con, short for console, is a combined keyboard/monitor device, prn for printer is by default a name for lpt1 and aux a name for com1. The VI or Plus PC DOS mode command allows reassignment of these devices. LPTn and COMn are names for the parallel and serial ports on the computer. These device names are assigned even if you don't have the full complement of ports. The V-9000 must have additional drivers loaded and defined via SERIALA, SERIALB and PPRORT. Check your DOS manual for details on the switch settings. You do not have to use all the drivers and can save a little memory by omitting them if they are not needed.

## Examples of Device Drivers: Console Drivers

The most common device driver to install is a console driver which replaces the default console driver. Some of these replacements attempt to address the notoriously slow display speed of the monitors and/or the annoying flicker on the color graphics display. In addition, some of the escape sequences of the 1977 console standard of the American National Standards Institute (ANSI) are implemented. These sequences include ways of controlling colors, cursor position and some DOS level keyboard macros. (They are described in my article ANSI.ART). One console driver of this type called ANSI.SYS is supplied with DOS and takes about 2K of resident memory. (ANSI.SYS for DOS 2 is available on BAVUG or from the Public Domain Library). It does not address the speed of display issue but it does implement several ANSI escape sequences. There are numerous programs which assume the ANSI.SYS is installed to operate properly (as well as a few that don't work properly if ANSI.SYS is installed!) so it is wise to install ANSI or an equivalent driver even if you do not want to use its features yourself. Actually, it is not hard to use the driver at the DOS level to set up a fancy prompt or redefine keys.

## Examples of Device Drivers: Other Drivers

Some older hard disks are not self-booting and require a device driver loaded in your CONFIG.SYS but that is not so common any more. IBM DOS 3.2 has a program called DRIVER.SYS which is a device driver to initialize external 3.5-inch drives. By far the most common drive device driver is to operate a RAM disk, that is a segment of RAM set aside as a fast virtual disk. There are com files loaded after the CONFIG.SYS which set up such drives but generally it is more sensible to use a device driver for this. Guy Gordon's Ramdrive comes with a file RAMDRIVE.SYS to set up a RAM disk. In addition, Microsoft WINDOWS, if you have a VI or Plus PC, comes with a RAM disk device driver (which can be run independently of WINDOWS) and which can be set up. Given Microsoft's experience and the care they have lavished on WINDOWS, I'd recommend trying the WINDOWS RAM disk driver if you have it in preference to alternatives. On IBM's, if you want to set up more than one RAM disk, you can include more than one line loading a RAM disk driver in your CONFIG.SYS file. You can normally load the same driver twice or use different drivers if you prefer. Be warned that there is typically a few K overhead in conventional memory to load a RAM disk and you will pay this overhead more than once if you load more than one RAM disk.

Print spoolers set aside some memory to receive printer output and then send that output to your printer as a background process. I regard them as a tremendous productivity tool. While there exist print spoolers loading as com files, many are

loaded as device drivers.

As you may know, you can place remarks in your BATch files and in particular in your AUTOEXEC.BAT. This is useful if you want to temporarily run your system without some resident program that is usually loaded in your AUTOEXEC.BAT file. You need only "remark it out"; i.e., add the phrase "REM " at the beginning of the line including it. Technically, remarks are not allowed in CONFIG.SYS files. If you insert the word "REM" at the start of a line in your CONFIG.SYS file you will get the message:

Unrecognized command in CONFIG.SYS

However, since the rest of the line is not acted on, this procedure will have the desired effect of "commenting out" the line in question, so you should not hesitate to use it.

ECHO also doesn't work in CONFIG.SYS, so there is no direct way of placing messages on the screen during the loading of the CONFIG.SYS.

## The FILES Command

DOS is a prisoner of its past. Original IBM PC's came with only 16K of memory (!) so when DOS boots up it sets aside memory for various purposes in an incredibly frugal manner. The defaults for three regions of memory set aside for file handles, disk buffers and environment are woefully inadequate. If you know what you are doing, it is easy to change these defaults, but it's unfortunate that the novice gets stuck with these small values. In any event, FILES and BUFFER commands are among the most important for you to include in your CONFIG.SYS. When DOS opens a file, it keeps certain information in memory to be able to quickly access the file. This information is called a file handle. During bootup, memory is put aside for these file handles, so a limit is placed on the number of files that can be open at one time. The default is eight, which may seem adequate, since programs normally close files when they are done, allowing the file handles to be reused. However, eight is often not adequate. DOS uses four of the handles itself for "files" like con and prn. Thus, there are four available for your programs. Some resident programs leave files open and even the ones that don't, may need to open a file for an initial access at the same time that an application program has several files open. Database programs often have separate index and data files and, typically, may want to have more than four open files. If DOS is asked to open a file and a handle is not available, DOS issues an error message and the running program may even abort. I strongly recommend that you place the line

FILES=20

in your CONFIG.SYS file. Indeed, since the cost of increasing files is less than 40 bytes per handle, you could even use a number larger than 20. For most purposes 20 should suffice; but, ever since it wasn't enough for me in a rather specialized situation, I've taken files=30.

## BUFFERS

You may have heard of disk caching. As you've noticed, diskette access is very slow and even a hard disk has access times 100 fold greater than RAM access times. Disk caching sets aside some RAM to keep a copy of the most recently accessed disk information so, for example, if a database is continually accessing a disk, the first time the disk is really read but the next time the copy in cache memory will be read instead. This is not the place to discuss the pros and cons of commercial disk caching software but you should know that DOS comes with some free rudimentary disk caching included. It keeps N buffers of 512 bytes each with copies of the last N disk sectors accessed. By default N, is only two (three on the AT). You should certainly make this number larger by including the line

```
BUFFERS=N
```

in your CONFIG.SYS where recommended values of N are between 10 and 25.

Let me tell you an anecdote to show how dramatic a difference this number can make. The first time that I ran my tape backup drive to backup my 30 meg hard disk, I was bitterly disappointed. Despite what I'd been told by the salesman, it took over 45 minutes! The next day, when I thought about it and tried again, it took only 8 minutes! What had happened? The first time I had been nervous about the effect my many resident programs might have so I put an original write protected DOS disk in drive A and rebooted before running the backup software. This disk had no CONFIG.SYS so I was running with the default three buffers. The next day, I used my regular hard-disk boot with buffers=20 and that made the difference. I have done some time tests comparing something as simple as copying a directory from a hard disk to a floppy and I've found that using extra buffers can decrease times by 30 or 40 percent. So USE YOUR FREE DISK CACHING.

The issue of precisely how many buffers to take is not an easy one. Increasing the number of files handles has little effect on memory or efficiency so you can freely take files=99 if the mood strikes you. This is not so with buffers. Each buffer takes .5K of RAM, so buffers add up. Moreover, at some point it will take DOS longer to check through all its buffers looking to see if a file is there than it would take it to access it directly. I've seen the number 25 given as a dividing line but I would like to do some tests to check this. I can only say that I've settled on buffers=20 myself and that, with a floppy based system, you should take a higher figure than you might with a hard disk.

### Increasing Your Environment

DOS sets up a special section of memory, called the environment, which has a default size of 160 bytes. This area

must hold your path, your prompt, the place that COMMAND.COM can be found and various other strings. Programs can communicate with you by asking you to place information in the environment with the SET command. In addition, you can keep global variables in the environment to pass between BATch files. If you attempt to place more there than it has room for, you'll get a message "Out of environment space". With DOS 3.1 and later there is a CONFIG.SYS command allowing you to increase the amount of space reserved for your environment. There are patches for earlier versions of DOS which are listed, for example, in my article on ANSI.SYS. The procedure is documented in DOS 3.2 and so, likely, it will be a permanent feature of DOS. It is undocumented in DOS 3.1. The syntax is

```
shell=C:\command.com /P /E:nnn
```

where n is the number of bytes you want to set aside for the environment. For DOS 3.1, nnn represents the number of 16-byte paragraphs you want to set aside. So, for a 512-byte environment, take nnn=32 in DOS 3.1 and 512 in DOS 3.2. Obviously with a floppy based system, replace C: by A:

How much space do you need for your environment? That depends on your path, applications and how fancy a prompt you make. My advice is to do nothing until you have a problem, at which point you should remember that there is something that you can do.

For more advanced users, I note that the environment is not as benign as you might think. I know of several programs that crashed if there was too much in the environment (most of the ones I know about have been fixed) and one that crashed if the PATH was the last thing set in the environment. I have occasionally been baffled at what could be causing a conflict only to discover the culprit was the environment.

### Miscellaneous CONFIG.SYS Commands

There are some other commands that can be used in your CONFIG.SYS:

-You can turn BREAK ON; that is, have the operating system check for control C more often than just during disk I/O. This slows down certain processing but gives greater safety from certain kinds of dead ends. The syntax is a line saying

```
BREAK=ON
```

Unlike any other CONFIG.SYS command, this one can also be issued from the DOS command line or in your AUTOEXEC.BAT file.

-In addition to file handles, DOS has something called file control blocks which in DOS 3.x can be changed by an FCBS command. These are needed only if you have a LAN (local area network) and the parameters to take should be discussed

by your LAN software.

-DOS 3.2 has a STACK command. From what I've read, this is a real cludge and the manual seems to suggest that it was added at the last minute to solve a problem connected with a new way that DOS 3.2 treats the stack. In any event, if you use DOS 3.2 and seem to have unexplained crashes, try adding  
STACK=20  
to your CONFIG.SYS.

-DOS 3.1 and later allows you to use the SUBST command to assign drive letters to directories. In addition, with several RAM disks, you may want to assign a letter beyond the default last drive of E. DOS 3.x allows you to add a command  
LAST DRIVE = ? (where ? is a letter)  
Then you can assign any drive up to and including that letter. Even a last drive=z takes only about 1K of RAM.

-There is a COUNTRY command to control things like the time format. The default is USA.

One final remark about your CONFIG.SYS. The order of the commands is irrelevant except to the extent that certain device drivers like to be loaded before others (and, if you are loading two RAM disks of different sizes, you may care which is assigned which letter). As with most DOS commands, the syntax is not case sensitive.

As a review of what CONFIG.SYS can contain, let me list the CONFIG.SYS from one of my machines that is running DOS 3.2:

```
break=on
buffers=20
device=C:\bin\intel\emm.sys M3 I5 D
device=C:\bin\devices\fconbeta.dev
        /C=1/S=2000/H=0/V=0/R=200/L=1/W=1
device=C:\bin\devices\ramdrive.sys 1024 512 128 /A
device=C:\bin\devices\ramdrive.sys 1300 512 64 /E
device=C:\bin\devices\atqlpt1.sys 1644,1,3
device=C:\bin\devices\mouse.sys
files=30
lastdrive=z
shell=C:\command.com /P /E:512
```

## AUTOEXEC.BAT

What should your AUTOEXEC.BAT contain? Most of my AUTOEXEC.BAT file loads my own particular blend of resident programs. This is not the place for me to advise you on what resident programs you might want to put into your system. However, I will make some comments about DOS and general aspects of what goes into your AUTOEXEC.BAT file.

First, if you have very many resident programs, they may have

conflicts and you must be prepared to permute the order of loading. This often cures some or all of the conflicts. For technical reasons (that I won't go into here) it really does pay to rearrange the loading sequence of resident programs if there is a conflict.

In addition to loading a stable of resident programs, your AUTOEXEC.BAT can contain some of the following:

-VERIFY ON - This slows down copying because DOS checks that the copy at least has consistent CRCs; this is not the same as comparing after copying, but it is a fairly good check. Only several compensating errors could pass this test after an incorrect copy.

-PROMPT - At a minimum use  
prompt=\$p\$g  
Mine uses ANSI.SYS to set colors and place the path and date on the bottom line of my screen

-PATH - If possible, keep your path short; every time you type in a bad command, DOS will have to read every directory in the path before responding "Bad command or filename". Also, list the path in the order of how many times you expect to access a given directory. That is, place the directories you call most often early in your path. If you have a RAM disk, place its directories first in the path. If you have a relatively large RAM disk, consider copying your BATCH file directory and the programs you call often to that RAM disk and place that RAM disk first in your path.

-If you have a large RAM disk, consider copying COMMAND.COM to it and placing the command  
SET comspec=D:\command.com  
in your AUTOEXEC.BAT (assuming D: is your RAM disk). Even without a large RAM disk, it is worthwhile to do this on a floppy-based system. What the command does is tell DOS to look there when it needs to reload COMMAND.COM. (Large programs will overwrite a part of COMMAND.COM and when they exit, DOS will try to reload COMMAND.COM. With the above command, you'll no longer get "Place a disk with command.com in drive A: and hit any key to continue".)

-It really is important to put the proper date and time in your system. Be sure to include the DATE and TIME commands. Alternatively, install a clock and place the appropriate commands into your AUTOEXEC.BAT file to set the system time from the clock.

-If you want to keep track of how often you boot, keep a record in a convenient directory. Make a file called junk, consisting only of a carriage- return line-feed and include the lines

```
date >>directory\logon <junk
```

```
time >>directory\logon <junk
You will then get the lines
Current date is Wed 7-23-1986
Enter new date (mm-dd-yy):
Current time is 16:29:22.70
Enter new time:
for each time you bootup.
```

### Speed and Memory Tips

Some final remarks about tricks to minimize memory usage and speed your bootup procedure. When DOS loads any program it saves a copy of the current environment in memory, one copy for each program. It doesn't force the copy to be as large as the empty space that you've set aside via a shell command but only to keep, in full, the present value of all environmental variables. Thus, you can save memory by keeping the environment small while your AUTOEXEC.BAT file is loading your resident programs. Two variables are always present: path and comspec. I start my AUTOEXEC.BAT file with a line

```
Path=A
```

This is incorrect syntax and is ignored when the path is needed. Thus, I must use full path names of all the programs that I load (but that speeds processing anyway). I reset the path and set the prompt at the end of my AUTOEXEC.BAT after I've loaded my resident programs. Given my fancy prompt, I save almost 200 bytes per resident program from what would be required if I set my path and prompt at the beginning of my AUTOEXEC.BAT. In total, I save several K of RAM; not a lot, but every little byte helps.

BATch files are read by DOS a line at a time, so BATch files really do get processed much faster from a RAM disk than from a floppy. There is a smaller difference between a hard disk and a RAM disk. If you have a RAM disk and a floppy-based system, it is well worth your while to place what would have been your AUTOEXEC.BAT in a file called startup.bat and have your AUTOEXEC.BAT read:

```
copy startup.bat C:
```

```
C:startup.bat
```

assuming your RAM disk is C:. To conserve space, you can have the last line in startup.bat say

```
erase C:startup.bat
```

You'll get a "batch file missing" error message but, other than that, the method will work perfectly. This procedure can also be used on a hard disk. The savings when I did it on my hard disk was two seconds out of about 65, so you may not feel it is worth-while.

You can slightly speed processing of BATch files, especially from floppies, by using the FOR...IN...DO command to combine several commands in one line. For example, if you want to copy \bin\batfiles, \bin\dump and \bin\opsys to your RAM

disk you might try

```
for %%a in (\bin\batfiles \bin\dump \bin\opsys)
```

```
do copy %a C:\>nul
```

if C: is your RAM disk. This can actually cut about 10% off a long AUTOEXEC.BAT file. Several warnings are in order. First, FOR...IN...DO parse the list at spaces so you can't combine commands which have parameters in this way. Secondly, I strongly recommend against using this device to load resident programs, particularly if you plan to use Kokkenen's MARK/RELEASE package.

### Summary

By using your CONFIG.SYS and AUTOEXEC.BAT files you can personalize many aspects of your PC.

While this original article was written for a IBM and has been modified to a large extent for the Victor, all of the techniques mentioned should work if you have the appropriate level of DOS. If you don't have at least DOS 2.x, you should seriously consider upgrading to DOS 3.1. Most new applications programs and several utilities require DOS 2.x or higher to run.

---

### V9000 COMPUTERS FOR SALE

Several Victor 9000 computers, all with double-sided double floppies and 256K RAM, are available at \$400 each:

Byron Hutchison, (408) 427-3276

1127 1/2 Soquel Ave.,

Santa Cruz, CA 95062

Two Victor 9000 computers, with double-sided floppies, 640K RAM, Easy Writer, Lotus 1-2-3 and tutorial, and MS-DOS 1.25, are available at \$750 each. Call:

Joan Madden

(408) 438-6650 X313 days

(408) 438-4512 evenings

---

### FORMAT CHANGE

In this issue we are introducing a format change that we hope will be beneficial to users. Rather than using mailing envelopes, as in the past, we are adding an extra (temporary) cover sheet. This will provide protection in the mail in place of an envelope and, at the same time, provide tear-out order forms (for Public Domain disks, back issues of VictorTalk, or Victor\*Group membership). This will eliminate the need to either copy the order form or tear up the main body of VictorTalk. It also expands, by one page, the space available for articles. We hope you like the change.



## FINAL SALE ON SOFTWARE FOR VICTOR 9000 SYSTEMS

The following Victor price list and product information has just been received from Rosemary Abell of Victor Technologies, Inc.:

FINAL SALE ends SEPT. 4th

850101	GRAPHICS TOOLKIT	3.0	\$50.
850202	APPLICATION PTK TOOLKIT	3.0	60.
850620	HARDWARE REFERENCE MANUAL	N/A	30.
850900	VBASICA INTERPRETER	1.0	30.
851000	MS-DOS 2.1 HARD DISK	2.1	50.
851100	MS-DOS 2.1 FLOPPY	2.1	50.
855500	MULTIPLAN	1.2	30.
856501	ALDIO TOOLKIT (INCLUDES HARDWARE)	2.1	100.
857100	EASEL GRAFIX	1.2	30.
857400	ASYNCR	2.53a	30.
857500	3276 SNA/SDLC	1.5a	30.
857700	TD830	1.08	30.
857801	HARD DISK TOOLKIT	3.0	50.
861200	MS-COBOL	2.1	75.
861205	MS-COBOL (PC FORMAT)	2.1	75.
862200	MS-FORTRAN	3.31	75.
862205	MS-FORTRAN (PC FORMAT)	3.31	75.
862500	MS-PASCAL	3.30	75.
862505	MS-PASCAL (PC FORMAT)	3.30	75.
863100	MS-BASIC INTERPRETER	5.28	30.
863105	MS-BASIC INTERPRETER (PC FORMAT)	5.28	30.
865505	MULTIPLAN (PC FORMAT)	1.2	30.
869200	MS-BASIC COMPILER	5.36	65.
869205	MS-BASIC COMPILER (PC FORMAT)	5.36	65.
869700	BIS-86/3780	5.15	65.
869800	IEEE-488 (INCLUDES CABLE)	1.2	50.
VIC-6580	CP/M-80		30.

FINAL SALE OF VICTOR 9000 S/W AVAILABLE TILL SEPTEMBER 4TH. REPAIR AT OUR DEPOTS AND SPARE PARTS ARE ON-GOING. THE ONLY S/W TO CONTINUE IS:

861000	MS-DOS 3.1	3.1	95.
861100	MS-DOS 3.1 PORTABLE	3.1	95.

Send orders to:  
VICTOR Technologies, Inc.  
Software Updates  
380 El Pueblo Road  
Scotts Valley, CA 95066  
(408) 438-6680

California residents include 6.5% sales tax. VISA or MASTER CARD accepted; please include number, expiration date and phone number.

Victor's Regional Repair Depot setup has been changed to the following:

### Western

Victor Technologies	Serves all states
380 El Pueblo Rd.	west of the
Scotts Valley, CA 95066-4269	Mississippi River
(408) 438-6680, Ext. 283	plus Wisconsin.
FAX (408) 438-6262	

### Eastern

Victor Technologies	Serves the
2166 W. Park Pl., Suite C	remaining states.
Stone Mountain, GA 30087	
(404) 498-6175	
FAX (404) 498-6686	

For spare parts, contact:

Victor Technologies  
380 El Pueblo Rd.  
Scotts Valley, CA 95066-4269  
(408) 438-6680 Ext. 240  
FAX (408) 438-6262  
TELEX 176110  
TWX 910-598-4511

Contacts for following old V9000 products:

Lotus 1-2-3	Ver. 1.0A	Affiliated Computer Group
		Joel Schardt
		4605 Cedar Lake Road
		Minneapolis, MN 55419
		(612) 377-7880

Plus PC	Ver. 3.04	A.S.I.
		Ed Costa
		200 Westland Ave.
		Manchester, NH 03103
		(603) 783-9401

Network S/W	Ed is writing software
	for the 2.1 and 3.1
	operating systems.

Used VICTOR 9000's	William Bailey
	(806) 541-8262

\* \* \*

# MS-BASIC, Part 4 / Jack Prichard

Part 4 of this series will complete the introduction of the statements and commands in the file operations category. [See Table 2 in Part 3 (VictorTalk Vol. 5, No. 3, May-Jun 1987).] It will also introduce the statements and commands used with both sequential and random access data files.

## PROGRAM FILES

The files we have dealt with thus far are program files. The statements and commands we have used in dealing with them are: LIST, LLIST, LOAD, MERGE, NEW, RUN, and SAVE.

There is one additional command for use only with program files, RENUM (renumber). It is used to renumber some or all of the lines in a program. Like the command AUTO, its format includes an optional (old) starting line number (default is first line of the program) and an optional line number increment (default is 10). In addition, its first optional parameter is the new starting line number (default is 10). Thus, to renumber lines starting at 50 in an existing program to 100, in increments of 5:

```
RENUM 100, 50, 5
```

Additional commands used with program files are: FILES, KILL, and NAME. Unlike the commands already dealt with, these commands are not limited to use with program files, but can be used with files of any type (no file extension or any file extension).

FILES is equivalent to the MS-DOS command DIR /W and is used to display the directory of files of the specified (or default) drive. Like DIR, FILES displays the files in an unsorted order (with no provision for sorting); it also allows the use of the wild-card characters ? and \*. Unlike DIR, it has no switches to vary the display format, but provides only a five-across listing with only the file names (including extension).

KILL is equivalent to the MS-DOS command DEL and is used to delete an entire file from within MSBASIC.

NAME is equivalent to the MS-DOS command REN and is used to rename a disk file from within MSBASIC.

The above statements can be used in either the direct mode (keyboard input) or the indirect mode (program control). As with all file-related statements, the filename must be enclosed in quotes; the trailing quote may be omitted when the <extension> part of the file name is not used. That is:

```
KILL "B:TEST1
```

Program files may be stored or SAVed in three ways. When no format is specified, the default mode stores in compressed binary format and is, therefore, most efficient in the use of file space. Using ,A at the end of the SAVE command, stores the program file in ASCII format. This requires more storage space, but the file is then directly readable with TYPE and usable with MERGE, etc. Using ,P at the end of the SAVE command, stores the program file in an encoded binary format and protects it from being listed or edited.

## DATA FILES

In addition to program files, there are also data files. These can be in either of two forms: sequential or random. The OPEN statement is used, amongst other things, to declare whether the file is sequential or random. The use of both the OPEN and CLOSE statements will be detailed in the section that follows.

It is recommended that, in conjunction with reading the next two sections (and doing related experimenting on your computer), you also carefully read the section "Disk Data Files" in your Victor documentation. (Page references are: '82 UG 4-2; '83, 90; '85, F-3; for description of '82, '83, and '85 see Table 2 in Part 3 of this series, as previously referenced.)

## SEQUENTIAL DATA FILES

In a sequential data file, the data is written sequentially as a series of ASCII characters; the data is read in the same manner. It has the advantage over a random data file of being simpler to use, with fewer commands or statements to learn. However, it has the disadvantage of being less flexible. If data in a sequential file is to be changed in any way, the entire file must be read, the change made, and the entire file rewritten to disk. Thus, if long files must be changed, it is desirable that they be random, rather than sequential files. (In a random file, one or more records of the file can be read, changed, and rewritten at a time, without touching any of the remainder of the file).

To either write to a file or to read from it, the file must be opened with the OPEN statement. The format used is:

```
OPEN <mode>,[#]<file number>,<filename>
```

Mode O specifies sequential output mode from the I/O

buffer to the file; mode I specifies sequential input mode from the file to the I/O buffer. Only one file may be open at a time in O mode; however, more than one file may be open at a time in I mode.

<file number> is an integer from 1 to 15 used to associate other I/O statements with the file as long as it is open.

<filename> is the disk filename in the usual format (covered in Parts 1 and 2 of this series).

When done writing to a file or reading from a file, the file must be closed. This is normally done with the CLOSE statement:

```
CLOSE[[#]<file number>[,[#]<file number>]]
```

Thus, one or more files may be closed selectively, or, if no file number is specified, all files are closed. (Both the END statement and the NEW command also close all disk files.)

RESET is the last command in the file operations category to be introduced. Its stated purpose is to close all open files and write all blocks in memory to disk. Earlier Victor documentation ('82 and '83) adds a caution: "Always execute a RESET command before removing a diskette from a disk drive. If you forget, the diskette will not have the current directory information written on the directory track when it is used again." Later documentation ('85) softens the caution by stating: "PURPOSE: Closes all files. REMARKS: RESET closes all open files and writes all blocks in memory to disk. You must close all files before you remove a disk from its drive." All documentation states that with sequential output files, CLOSE writes the final buffer of output. However, END and SYSTEM also close all files. Just what the subtleties are between RESET and CLOSE (and END, and SYSTEM, with respect to file closure, etc.) is not clear from the documentation. At any rate, I have never used RESET and I am unaware of any trouble traceable to its omission. Therefore, I suggest you forget it (at least until a need arises).

With this little song and dance, we conclude the introduction of the file operations category of commands and statements (see Tables in Parts 1 and 3 of this series). However, there is one remaining entry (actually a function): LOF. Its format is:

```
LOF(<file number>)
```

<file number> must be a currently open file. The function returns the number of bytes allocated to a file (in multiples of 128 bytes). LOF was included in this category because of its use with files and of its omission from the list of functions in the early ('82) Victor documentation. (Probably it would have been better to have listed it under the "miscellaneous functions" category in the table of functions by usage category.)

## OUTPUT TO SEQUENTIAL DATA FILES

Three different statements may be used to output data to a sequential data file:

```
PRINT#  
PRINT# USING  
WRITE#
```

Their formats and usage are given in the Victor documentation (which see, via the Tables of Part 1 or 3). The distinctions between the three statements are:

PRINT# and PRINT# USING both require the use of explicit delimiters to separate the numeric and/or string expressions.

WRITE# does not require the use of explicit delimiters but automatically inserts commas between items as they are written to disk and delimits strings with quotation marks.

PRINT# USING differs from PRINT# in that it controls the format of the disk file by a string expression. (It is analogous to the PRINT USING statement introduced in Part 2 of this series.)

Note carefully the syntax for use of these statements; it is very tricky! While I have given additional hints on their use and example programs using them, it is likely that you will require a lot of hands-on experimentation to get it right for your own applications.

## INPUT FROM SEQUENTIAL DATA FILES

Two different statements may be used to input data from a sequential data file:

```
INPUT#  
LINE INPUT#
```

Their formats and usage are covered in the Victor documentation. (In addition to reading the descriptions of the statements in the referenced pages in the Tables of Parts 1 or 3, also see the section on sequential files previously referenced.) The distinction between the two statements is that INPUT# is the form used with data files in which the numeric data or string items are delimited by commas. Where one or more commas are included as a part of the data string (not as delimiters), the LINE INPUT# form is used.

The number of data items in a file should match the number of variables in the variable list of the INPUT# statement (or should be an integral multiple of it when the INPUT# statement is used in a loop). If the number of data items is too small, on execution, an error message will be displayed: "Input past end in \_\_\_\_". If the number of data items is too large, not all the data items will be inputted.

## EXAMPLES OF SEQUENTIAL DATA FILES

Three brief programs follow to show how these statements could be used to create a file in different formats (using the date and time of each file's creation as its data).

```
10 OPEN "O",#1,"D1.DAT"
20 PRINT #1, DATE$, TIME$
30 END
```

```
10 OPEN "O",#1,"D2.DAT"
20 PRINT #1, USING "\ \ \ \ ";
DATE$, TIME$
30 END
```

```
10 OPEN "O",#1,"D3.DAT"
20 WRITE #1, DATE$, TIME$
30 END
```

When displayed or printed using the command TYPE, the data files created by these programs appear as follows:

```
B>TYPE D1.DAT
07-22-1987 20:46:57
```

```
B>TYPE D2.DAT
07-22-1987 20:47:35
```

```
B>TYPE D3.DAT
"07-22-1987","20:47:51"
```

Note the format in which the data is stored:

With PRINT #1, the data is stored in columns with the same character spacing as with the printed output when the command LPRINT is used and the data items are separated by commas.

With PRINT #1, USING, the data is stored in the format specified.

With WRITE #1, the data is stored enclosed in quotes and with commas between data items.

The following example MSBASIC program, when executed, shows how the three files can be read and the data used, in this case to simply print out the data of each file in two different formats. Note that, while the data in the three files is stored differently by the three different commands, the data read out is printed in the same format (controlled by the LPRINT statement). Also note that, with the data items separated by semi-colons in the LPRINT statement, the data items are not separated in the printout.

```
5 INPUT "FILE ", F$
10 OPEN "I",#1,"B:"+F$+".DAT"
```

```
20 INPUT #1, DAT$, TIM$
25 LPRINT DAT$, TIM$
26 LPRINT DAT$; TIM$
27 LPRINT
30 CLOSE
40 GOTO 5
```

```
FILE D1.DAT
07-22-1987 20:46:57
07-22-198720:46:57
```

```
FILE D2.DAT
07-22-1987 20:47:35
07-22-198720:47:35
```

```
FILE D3.DAT
07-22-1987 20:47:51
07-22-198720:47:51
```

## RANDOM DATA FILES

To either write to or read from a random file, it must first be opened. The statement used is the same OPEN used with sequential files; however, the mode is specified as R, for random input/output mode. In addition, a final optional integer expression, <reclen>, may be added to specify the record length. (The default length is 128 bytes.) As with sequential files opened for input, more than one random access file may be open at a time.

Note that the format for the OPEN statement in VBASICA differs from that in MSBASIC (although these differences are not reflected in the examples given in Appendix F of the '85 documentation). However, since this series of articles is concerned with MSBASIC, such details about VBASICA will be deferred until a more complete coverage of VBASICA can be given (either as a later part in this series, or as a separate series).

The statement CLOSE is used to close a random data file in exactly the same way as with a sequential file; all the prior comments apply equally.

## OUTPUT TO RANDOM DATA FILES

Each record of a random file is divided into fields using the statement FIELD. FIELD refers to the file number and specifies the length in bytes and the variable name of each field in sequence. It actually allots space in the random buffer; it takes the place of the explicit delimiting between variables in sequential files.

Unlike sequential files, all data entered into the random buffer must be in string form. Numbers must be converted to

string form by the functions MKI\$, MKS\$, and MKD\$ (for integer, single-precision, and double-precision, respectively). The loading of the random file buffer is accomplished with either the LSET or the RSET statement, depending on whether the data string is to be left-justified or right-justified in the field.

After the data in the random buffer is assembled it is converted to a packed binary format and stored in the disk file by the statement PUT. (As was noted in Part 3, the PUT statement for file I/O use is missing from Chapter 3 of the '85 Victor documentation; however examples of its use are given on pages F-9 and F-12. The PUT statement for COM use and the PUT statement for graphics use are described under the corresponding GET statements in Chapter 3.) The PUT statement writes one record at a time to the specified random file, either to a specified record number, or, if none is specified, to the next record number.

#### INPUT FROM RANDOM FILES

Reading from a random file is the reverse of writing, with the GET statement substituting for the PUT statement. GET reads a record into the random buffer, from which the program may then extract the desired field(s) of data. If numbers are to be used in non-string format, they must be converted using the functions CVI, CVS, or CVD (to convert to integer, single-precision, or double-precision form, respectively).

While packed binary is a more compact form for storage than is ASCII format, it does have the disadvantage of not being directly readable by statements such as TYPE, that work only with data in ASCII format.

#### EXAMPLE RANDOM DATA FILE

As an example, assume we want to create a random file of personnel data with name, height, and weight. We will designate these items as N\$, H\$, and W\$ and assign fields of 20, 3, and 3 characters to them, respectively. The file would be opened with

```
OPEN "R", #1, "B:TEST.DAT", 26
```

```
FIELD #1, 20 AS N$, 3 AS H$, 3 AS W$
```

The first line would open, for random input and output, a file designated as #1 and named B:TEST.DAT, with a record length of 26 characters. The second line would assign the desired number of characters to the respective fields. Note that, since the fields are simply strings of characters, the numerical data must be converted to string form before being input to the data buffer. This would be done and the data strings input, flush left, by

```
LSET N$=NM$: LSET H$=MKI$(H): LSET W$=MKI$(W)
```

The data would then be input as the Ith record by

```
PUT #1, I
```

That's all there is to writing a random data file!

To read the personnel data file just created and convert the data back to its original form and display it, the following could be used. (It is assumed the file is still open. If it is not, it must be reopened with the OPEN and FIELD statements used above.)

```
GET #1, I
NME$=N$: H=CVI(H$): W=CVI(W$)
PRINT NME$, H, W
```

That's all there is to reading and displaying the data from a random data file!

The several lines of program given above have been assembled into a working program. Other lines have been added to provide for:

- o Keyboard input of the data, with appropriate prompts
- o Shift from output to input data, and vice versa, by entering "0" or CR for record no.
- o Exit from program by typing "-1" for record no.

Type in this program and SAVE it as "B:TEST.BAS", A or whatever appropriate filename you prefer. In running it, note that record numbers may be written and read in any arbitrary order. You may leave gaps in the record numbers sequence you enter and you may also overwrite any record. Experiment with it to get an appreciation for the flexibility of the random access data file.

```
10 OPEN "R", #1, "B:TEST.DAT", 26
20 FIELD #1, 20 AS N$, 3 AS H$, 3 AS W$
30 INPUT "Input Record No. ", I
40 IF I=0 THEN 200
50 IF I=-1 THEN 300
60 INPUT "Name           ", NM$
70 INPUT "Height         ", H
80 INPUT "Weight          ", W
90 LSET N$=NM$: LSET H$=MKI$(H): LSET W$=MKI$(W)
100 PUT #1, I
110 GOTO 30
200 INPUT "Output Record No. ", I
210 IF I=0 THEN 30
220 IF I=-1 THEN 300
230 GET #1, I
240 NME$=N$: H=CVI(H$): W=CVI(W$)
250 PRINT NME$, H, W
260 GOTO 200
300 CLOSE
```

\* \* \*

# NEW EYES FOR OLD / Roger Habeck

When you first purchased your V9000, in all likelihood, one of your big reasons was the wonderful clarity of the monitor. You had looked at other computers and the awful screen resolution nearly tore the eyes out of your head.

Well, now its a few years later and that good old Victor Monitor just doesn't look so hot any more. The characters look mighty fuzzy and no matter how you fool with the intensity keys you just can't get a sharp or comfortable image.

**DO NOT DESPAIR !!!** That wonderful, sharp, crisp monitor you miss so much is only one hour away.

I am about to let you into my NO. 1 BIG SECRET. One of my standard services is a general machine cleanup for which I charge \$100.00. My customers usually get this done on an annual basis and seem to feel that it represents a good value.

This package includes the following:

1. Disassemble and clean out the dead bugs, fruit loops, paper clips and eraser bits from the keyboard.
2. Disassemble the CPU, and clean out the thick layer of dust balls that has accumulated. (Heat kills electronics and a thick layer of dust is better than a thermal blanket for holding in heat).
3. Test the power supply and run System Diagnostics. **AND.....THIS IS IT.....THE REAL BIGGIE...**
4. Service the monitor.

When Victor designed the V9000 Monitor, they not only used a much higher resolution than any one else at the time, they picked up a trick used by the very best mainframe monitor makers. They applied a very fine mesh nylon (I think) screen over the face of the monitor. However, unlike many of those very expensive mainframe monitors, Victor did not bond the mesh to the monitor (and are we ever lucky); instead, the mesh is mounted in a sort of cardboard picture frame and held very tightly over the front of the CRT (cathode ray tube) by the front bezel.

This is what has happened to your great resolution. All CRT's create a static electric charge on the face of the tube. (This is why the screen on your T.V. is so grungy.) This static charge attracts dust, smoke particles, air freshener droplets, almost anything floating around in the air. Covering up your computer won't help much, because this static charge is only present when the computer is running. Anyway, all these little particles get into the openings and onto the strands of the mesh, thereby fuzzing things up.

Victor supplied a special cloth for wiping off the CRT which was either supposed to neutralize the static charge or attract the dust with a charge of its own. Whatever... in any event, I have never noticed a particular difference between using or not using the special cloth. The CRT gets grungy anyway.

## Cleaning up the CRT

The only tool needed for this whole operation is a number 2 phillips screw driver.

**Step 1.** Unplug the monitor from the CPU. The monitor plug is a squeeze-type plug; press in on both sides and it will slide off easily. If it resists, you are not squeezing correctly.

**Step 2.** Lift the monitor off the turntable and set face (screen) down on a towel or blanket in a clear work area.

**Step 3.** Remove the two screws in the bottom. The top will now lift off. Remove top and place monitor on its feet.

**Step 4.** You will notice that the front bezel frame is secured to the CRT housing with two screws on each side. Loosen the screws about a turn and slide bezel forward until you can slide out the mesh screen holder. It should slide right out. If it is stuck, loosen the screws so that you can remove the front bezel entirely.

[Special Note. On occasion I have run into situations where someone attempted to clean the screen with windex, or spilled a beverage on it, or something and when I tried to remove the mesh, it seemed to be welded to the glass of the tube. No problem; dab a wet wash cloth over the surface of the screen. It may take a while, but almost anything that will end up on a monitor is water soluble. After a time you can peel the mesh right off. I think it unlikely that you will have this problem, since I have only run into it once or twice.]

**Step 5.** Clean the surface of the CRT with glass cleaner. Wipe off with clear water and buff dry with paper towels. You don't want any residue on the glass. Take the mesh to the sink and gently wash it with dishwashing detergent. I use my fingers only, as rags or sponges may catch in the mesh. You will be able to determine if the mesh is clean, because dirt and grunge are very noticeable when the mesh is wet. Place the mesh on paper towels and pat dry with paper towels. After the mesh is dry, examine it carefully. Sometimes grease or lipstick will require a second wash.

**Step 6.** Reverse the above to reassemble. When tightening the bezel you will be able to tell by the screw marks when you are in the correct position. Also, do not over tighten the long screws in the bottom. They are only into plastic, so you can strip them if you get carried away; finger tight is fine.

That's all there is to it. You just saved yourself \$50.00 and got back a lot of pleasure and satisfaction in your V9000.

Call me if you need to.

Roger Habeck (804) 520-2712

\* \* \*

## HOW TO CLEAN THE VICTOR 9000 SCREEN

Hermann Maseler

[For this charming "how-to" version we are indebted to VictorCircle Newdisk 27 (assembled and edited by Michael Stal of Munich, West Germany). - Ed.]

Many people touch on the screen with their fat and oil fingers when showing something to You. I always cry stop but oftenly it is to late. So after some time the screen is completely smeared.

Wiping with a cloth isn't of much success because of the grating on the front of the monitors glass. Don't use an aggressive cleaning medium, because You destroy the reflection film on the glass.

So after a time the best way to get a clean monitor is to do the work and detach it. It isn't difficult, only a little work, maybe an hour for an unused person. Unscrew the four screws on the bottom side of the monitor and detach the monitors cabinet. Than detach the frame on the front side of the monitors glass, so that You can take away the grating from the front. The grating You may clean under the shower or something similar. If it is very dirty use a new tooth brush. Without the grating in front of the glass You may now clean it very comfortable. Wait until the grating is dry again. Perhaps You want to speed it with Your hair foehn. Then mount everything back again reverse manner and You'll have a clean view on Your Screen again.

NICE LOOKING!!!

---

RE: VALID MEMORY BOARD TROUBLES IN A PLUS PC

July 14, 1987

To the Editor:

This letter is in response to Mr. Wayne Crouch's comments in the May June issue of VictorTalk. We would appreciate it if you would include this in your next issue.

The following summarizes our contact with Mr. Crouch.

1. Upon initial contact from Mr. Crouch, we sent him a new prom. As this did not fix his problem, he returned the part and received a refund.

2. We informed Mr. Crouch that we do not have a +PC on site, therefore we could not check his board against a +PC. We did inform him we had no other customers report any failure between the Valid board and the +PC. We suggested that he return the board to us so that we could check it for any component failure. Perhaps this was the problem. When he returned it, we replaced a failed controller chip, and sent it back to him.

3. Mr. Crouch sent us one letter (March 18) after receiving his board back. We attempted numerous times to contact him by telephone, to ascertain what else might be done to help him. As we did not hear from him again, we assumed he had resolved his problem.

At this time, Mr. Crouch has not contacted us since March 18.

Based on his comments in VictorTalk, he is using a "+PC(256K)" and a "Valid (+640K) board". If this is indeed his configuration, he has too much memory. The +PC uses the top 128K of memory, therefore the usable memory limit with a +PC is 768K. This might be the cause of Mr. Crouch's problem.

Mr. Crouch informed us that he installed his +PC himself. We are not aware of his qualifications to do such work, so we cannot determine whether or not his +PC was properly installed, or whether he had investigated this possibility. We are also not aware of his qualifications to diagnose the "problem" his +PC exhibits with our board.

In response to Mr. Crouch's comments, one of our dealers contacted him, offered to help, and suggested that Mr. Crouch call us. We still have not heard from him.

At this time, Mr. Crouch is the only customer, of 100's with +PC's, who has exhibited a problem. His generalization that the board "seems to work for some people, but not for others", is unfounded.

Those of you who have dealt with us directly concerning any problem with boards know our reputation for quality customer service. We can assure you that there is not a compatibility problem with the Valid Board and properly installed +PCs.

Sincerely,  
Wendy W. Cykana  
Valid Technologies

\* \* \*

# MESSING WITH MULTO / Bud Popken

A long time ago, Jack Prichard asked for a volunteer to try out Marty Moleski's utility program, MULTO. Better late than never, here is a report on my limited experiences with the program.

MULTO is a shareware program (meaning if you use the program, Marty would like some money) written for the V9000 with DOS 2.1 or later. It requires a minimum of 256K memory. It does run in the V Mode of Plus PC. The main purpose of the program is to minimize keystrokes while file handling. This is done by redefining the keyboard. I found a full size template and a list of functions and associated keys essential to moving around in MULTO. Some utilities similar to MULTO include:

mdir	cweep
ls	rendir
mv	menux
delq	list549
vdel	search
copyq	scrndump
wash	setepson

To get started, I copied MULTO's files, M.COM, MM15.HLP, and MM15.SYS onto a disk, along with the operating system. I also added RAM.SYS and SETRAM.EXE. (This is a ram drive available in the Public Domain Library.) Next, I added EDLIN, CHKDSK, FORMAT, and ARC.

Then, I set up the CONFIG.SYS file to:

```
Buffers=20
Device=Ram.Sys
```

CONFIG.BAT was set up to include:

```
Path=c:\;a:\;b:
Setram c 256
Set comspec=c:\command.com
Copy a:*. * c:
```

The purpose of all this is to use 256K of memory as ramdrive C: and to move MULTO into it to free drives A and B. The path command and the comspec are for MULTO's use. Now I load the MULTO disk in drive A, turn on the system, let it move MULTO to ram drive C, and then remove the disk from A.

To run the program, I select the C drive and type 'mf'. The directory of C drive is now displayed in the MULTO format. To move the cursor about in the directory, use either the

arrow keys or 'a', 'f', 'j', or '!'. Marty used those keys as that's where your fingers rest when on the home row. To change directories, type ALT A or ALT B for drives A and B, respectively. If you accidentally hit the wrong key (particularly one Marty has redefined) you can undo that keypress by pressing MULTO Escape, (the PLUS/MINUS key in the upper left corner of your keyboard, next to the '!' key). Another important thing to remember is an "\*" staring back at you means "press any key".

The MM15.HLP documentation will explain what the various redefined keys will do. Just remember, Marty uses the same key in combination with the SHIFT and ALT keys to do different things.

To use Multo to print the help file, or any other text file, the printer initialization string needs to be set for your printer. In MULTO, press 'F9'. This brings up the contents of MM15.SYS. These are the macros for running other programs by just pressing one or two keys. For now, the line beginning with 'P' is the printer init. string. This string sends the commands to your printer for margins, paper-skip, type style, etc. My Okidata didn't need any of this, so I deleted the string, leaving 'P' blank. 'F8' will save your changes to MM15.SYS. Now on to printing. If you are not at the main screen, hit the MULTO 'escape' key (upper left corner). Move the cursor to the desired file and press 'F4'. The printer should be off and running.

Some information on those macros. Press 'F9'. I decided to change the first macro to run the EDLIN editor for whatever file on which I placed the cursor. Using the arrow keys, I changed line 1 to read:

```
1--edlin.com ALT F,(looks like a spade)
```

'ALT F' appends the filename the cursor is on to the EDLIN command. Note you must use the full name of the command, including the extension such as .sys or .com or .exe. I changed line 2 to read:

```
2-- format.exe a: /v ALT D,(looks like a diamond)
```

This command will format a disk in drive A and then wait for a keypress before returning to MULTO. I changed line 9 to read:

```
9-- arc.exe I ALT F *.* ALT D
```

This command will, when I press 'ALT F9', list all the individual files found in an archived file selected by the cursor and then wait before returning to MULTO. For my word processor, I changed the line:

```
W--pt.bat
```



Now to run it, I place the word processor in drive A, hit 'ALT A' and then 'w'. When I leave word processing, I'm returned to MULTO.

Another function of MULTO is Treelist. Treelist searches all directories and subdirectories on the current drive and lists the file information found to a disk file, established by you. This can be done two ways:

1) In MM15.SYS, the macro "T--c:treelist" is the default, opening a file named 'treelist' on C drive. Change this to whatever drive and name you want.

2) Press 'SHIFT F10' and name the drive and file for this one use of Treelist (not permanent change). 'F10' will then run Treelist for the current drive and, by swapping disks in the current drive and pressing 'F10', you can catalog your disk collection. One word of caution -- the sort program used by Treelist uses a lot of disk space, so leave lots of room on your catalogue disk.

Hopefully, this brief wandering through Multo will arouse your curiosity or possibly answer some questions. Copies of the template I used to remember the keystrokes can be made available. Write or call:

Bud Popken  
106 Cardiff Ct.  
Santa Cruz, CA 95060  
(408) 427-3738

[Editor's Note: MULTO 15 is available on BAVUG's Public Domain disk #86. It is also available for \$10.00, and extra sets of keyboard templates for \$2.00 each, directly from MULTO's author:

Martin X. Moleski, SJ  
1225 Otis Street, NE  
Washington, DC 20017  
(202) 529-1252

Note that MULTO 15 is issued as shareware. Thus, if after experimenting with it, you intend to make continued use of it, a suitable donation to its author is suggested.]

\* \* \*

## V9000 AND VI HARDWARE AND SOFTWARE

Al Bullock, of Bullock and Associates [39522 Benavente Place, Fremont, CA 94539 (415) 651-6183], currently has available the following hardware and software, for Victor 9000 and Victor VI computers, to user group members at the prices indicated. (Hardware not shown as "new" is sold as is, without warranty.)

### Hardware

V9	Network Station 256K w/Network Board (New)	\$350
V9	Ext 10MB Hard Disk (New)	795
V9	" 60MB " " "	1495
V9	" Graymatter 10MB HD w/software	350
V9	" " 2x " " "	650
V9	Graymatter HD Controller & Interface Bds	100
V9	CPU Main Boards 256K (New)	200
V9	CPU Main Boards 128K & 256K	25
V9	Floppy Disk Controller Boards	25
V9	F3F7 Upgraded BootROMs (New)	25
V9 & VI	10MB Tandon HD Formatted (New)	200
V9 & VI	40MB Replacement HD (Formatted) (New or Rebuilt)	525
V9 & VI	60MB " " " " " " "	795
V9 & VI	Network Boards (New)	150
V9 & VI	Monitor (New)	75
	Sound Cabinet For Wide Carriage Printer (New, display model)	200
	C. Itoh Banana Dot Matrix Printer (New, but missing paper cover)	50
	Okidata Speedup Kit For Serial Printer	25

### Software

	dBASE III+	300
	Lotus 1-2-3	200
	Accounting Plus	100
	NPL/R Data Base (as used by Ford Motor Co.)	100
	Wordstar	100
	Super Calc	75
	MS-BASIC Compiler	50
	VPC MSDOS 2.11	35
	Victor WJ (Menu program)	30
	VBASICA w/Interpreter	25
	Victor Calc	25
	PC Com w/Cable	25
	Graphics Tool Kit	20
	MS-BASIC Manual	15
	Level II COBOL Manual	15
	Pascal Manual	10

\* \* \*

# FROM THE BOARDS

[FROM THE BOARDS is assembled from messages appearing on various (generally Victor-specific) bulletin boards, particularly The PUB, by John and Mary Jo Knobelsdorf. Only minor editing is done to correct obvious typographical errors and to eliminate some superfluous material. An effort has been made toward a degree of standardized formatting and the elimination of excess space to the extent practical without degrading legibility. Garbles that cannot be readily rectified are allowed to remain. Believing that considerable worthwhile material is contained in these excerpts, our intention is to include this feature in each issue of VictorTalk, as space will allow. - Ed.]

## WORDPERFECT, MULTIPLAN & KNOWLEDGEMAN and 8087 Math Coprocessor Boards

#638 LLOYD LEBLANC, Subject: WP 4.2

Well, as reported here by someone a few hours ago, WordPerfect 4.2 is available now in Victor 9000 version. However.....

SWP.COM as last revised does NOT work with the new version. I tried every combination (using the 4.1 WPKEYS and WP files, switching around) but the new WP.EXE or WP.COM must have been modified, and WP 4.2 seems to be looking for something new. Could be that tricky ability to assign a new ASCII code to the RETURN key, I surmise. If WP.EXE can recognize different ASCII codes for the RETURN or ENTER keys, it could be throwing off SWP.COM. It's up to Chuck Dayton now. Probably not a major problem. Needs SWP, too, usual slow as molasses screen updating.

Worse, perhaps -- WP CORP seems unable to implement, after all this time and many versions, either the SCREEN "Window" function or the visual reference bar function (for viewing REVEAL codes and setting a bottom tab line) on the Victor. Why, they know only in Orem, Utah. Since the new Victor version 4.2 shares some of the same quirks as the current IBM version -- namely the new screen margin handling and the "smart" (moving) cursor -- the ONLY reasons to get 4.2 in Victor format are (1) ability to use the Victor keyboard, which to me is a major reason, and (2) ability to use an appropriate SWP.COM, when and if Chuck D. can implement that.

I think I'll stick with 4.1 for a while, and use the IBM version (under the emulator) when and if I need some of the new functions.

#639 RICHARD JAFFE, Subject: Word Perfect 4.2

I just received 4.2 and the path batch file that I had used for

4.1 to have data written to b drive doesn't work. The old (pre-4.1) method of saving data to b drive (from B prompt "a:wp") also doesn't work. I called up WP customer support. They told me that at least one other V9 user has called with the same problem and that they were checking it out. Does anyone out there know how to solve this at the MS-DOS level. Of course I could change the default directory at WP, but I have a few machines and many people using them and this would just be another opportunity for operator error. Any suggestions would be appreciated.

#640 LLOYD LEBLANC, Subject: WP AND MULTIPLAN

The previous message regarding WP 4.2 is the first of a number we'll be seeing about the various "mods" let into 4.2 by WP Corp. Not all are beneficial. For a number of people, I'd suggest (unless you need the new legal features or the specific fixes to mailmerge sorting) that you stick with version 4.1; if you like the new "smart" cursor movement --- use 4.2. Be aware tho', that the SWP.COM version now available does NOT work -- so you go back to molasses screen handling.

Here's some MULTIPLAN news. The "latest" version (3.0) is substantially faster and better than the last -- version 2.0 (IBM series). I believe Victor is now offering version 2.0 in their packaging -- not sure if it's anything more than an "installed" generic version. Now -- version 3.0 WILL RUN on the V9000 -- as did the "installed" generic 2.0 -- but no thanks to Microsoft, who has in wisdom now reduced the former 50-computer 47k "installation data" file to a 12-computer 11k version (and guess who is missing, folks). Why --- gawd knows. BUT -- if you have the version 2.0 "installation" files -- you can use those to install 3.0 for the "Sirius-Victor" -- and use it. Works perfectly so far. DEFINITELY faster on screen movements, rewrites, recalcs. Other good features.

There is also a "custom installation for terminals" setup in the installation procedures that would appear to let you set MP up for the Victor and just about anything else, if you're totally conversant with screen attributes and a host of other escape codes, etc.

I recommend any prospective MP users to get the IBM version 3.0 and setup as described, or pressure Victor for the upgrade.

By the way, WP Corp does NOT intend to offer their much improved MATHPLAN program version -- 4.0 I think -- on the Victor. Period. Reason is the program now has fancy graphics routines, shutting us out.

#646 LLOYD LEBLANC, Subject: MULTI 3.0

Ernst -- others -- when you do port Multiplan 3.0 over to the

Victor, you'll save much trouble by getting the older version 2.0 installation data file from the MP3INS.ARC file on this board. I put a small DOC file in too. What's provided is simply the larger INSTALL.DAT file that contains the Victor 9000 code, eliminating the long customizing Microsoft leaves in the 3.0 package.

#31 DAVID ELESH, Subject: Sorting with WP 4.1  
I have sorted files of up to 400,000 records with WP 4.1. The trick is to avoid a SCREEN sort and to define an input and an output file. It takes a while (time depending on whether you are sorting from a HD or floppy) but it works.

#643 ERNST SCHRADER, Subject: Multiplan updates  
MULTIPLAN 2.01, 3.0  
On January 2 I reported here what Microsoft Corp. said was the proper way to obtain a V9-specific version of Multiplan 2.01. I followed my own advice, and yesterday received a package containing, not 2.01 in Victor format, but 3.0 on 3 IBM-formatted disks. The labels suggest that the program is the generic MS-DOS version; Lloyd Leblanc has suggested on the boards that this is the version we should be ordering anyway, not 2.01. I will report my experience with these disks when I find a way to port them over. To use Guy Gordon's Brooklyn Bridge, you have to have a pier on the other side of the water--an IBM-type machine of some kind. For the time being, I've chucked them in my ever-heavier box of IBM-format disks, to be gotten to Real Soon Now, and will content myself reading the large new Multiplan manual.

KNOWLEDGEMAN 2.01  
Same story, except the update was free. I made a fuss, and received 4 IBM-format disks. RSN, I'll port them over...

#100 SYSOP, Subject: SWP14.ARC  
O.K., all you WordPerfect fans. Chuck Dayton has modified SWP to work with WP Ver 4.2. You'll find it in File Section #6 or File Section #3. What's that Bunky? You say you haven't sent Chuck the measly \$5.00? You ought to be ashamed of yourself! Can't spend a fiver to make a multibuck program work the way it should've in the first place! Chuck deserves a portion of WordPerfect Corp's profits from the V9000 versions. Think he'll get it?

#32 DAVID RENSIN, Subject: wp 4.2  
I have a WP 4.1 macro that saves a document, clears the screen and remains in WordPerfect. However it doesn't work with WP 4.2. The reason why, for you who have the same problem, has to do with a 4.2 feature that allows you to add a header to each file that displays date, typist, author and comments. You can turn this off during WP setup -- otherwise old macros will elicit "unknown macro state" messages. But frankly, after trying 4.2 I went right back to 4.1. And even if Chuck Dayton modifies SWP for the new version (PLEASE) I may still not return.

To answer the complaint about the WP Shell exit to DOS leaving you with the WP keyboard .... hey, using SWP or not, that's what it does. Other programs may complicate things by leaving you frozen, of course. You can't ALT-C out because there isn't an ALT-C. To solve this, I KEYGEN-ed an ALT-C on my shifted decimal point on the numeric keypad. Saved lots of screw ups.

#39 ERNST SCHRADER, Subject: Multiplan 3.0  
Thanks to Lloyd Leblanc for MP3INS.ARC, available here on the PUB. After using a friend's Plus-PC to port over the three new Multiplan 3.0 IBM-format disks to Victor disks, I used Lloyd's program to install MP on a hard disk volume. Here are some simplified instructions:  
1. Set up a subdirectory for MP  
2. Copy into that subdirectory the contents of MP disks 1 and 2  
3. Use an ARC program to extract VER2INS.DAT from Lloyd's MP3INS.ARC. Copy VER2INS.DAT to your MP subdirectory as INSTALL.DAT, overwriting the file already there.  
4. Type INSTALL, hit return twice, type 13, return.  
5. Delete INSTALL.\* and MOUSE.\* (unless you have a mouse...)  
6. Look at your directory. If you see MP.INI, delete it. If you don't see the file, don't worry. It will show up the first time you run MP and quit--and the MP.INI file that shows up then is harmless, leave it be.

The only problem I've encountered is that MP's opening screen and copyright notice now suggest that the program is 2.0, not 3.0. Would we be asking for trouble if we asked Microsoft to re-issue the MP 2 Install program for the Victor, fixing it so the opening notices said 3.0 and posted the legally correct copyright notice?

This new MP 3.0 makes use of my 8087 chip. No need to turn automatic 11 recalculation off now--it's so fast, I thought it was off! Many new features: date and time calculations, macros, macro editing, bigger spreadsheets, print-outs of all names used, auditing and verification procedures... Microsoft, Lloyd, I'm very happy with this Program!!

#41 LLOYD LEBLANC, Subject: Multiplan Converts  
Ernst Shrader: you're right, using the older version 2.0 INSTALL.DAT file to set up for V9000 use in version 3.0 does write a version 2 label to the WP.COM file.....for fun, we simply searched in DEBUG for the label and changed it to read version 3.0 with the correct year dates -- a five minute chore. Try it if you feel the label should be compatible.

I'm most interested in use of the 8087 processor, not having had any reason before to install one. How does that get done, Ernst -- I presume it's a special card in the V9000 -- and how does one acquire the parts/user instructions (from Victor?

others?)? You are correct in that it is supported in MP now, and that alone makes it a major upgrade and better than 1-2-3 for many applications.

#42 SYSOP, Subject: Re: Multiplan Converts

Ernst Schrader sent the following question, which anticipates Lloyd's message on Board #7. The Fuzzy Bear apologizes for not posting Ernst's query earlier.

Maybe one of these days PUB users will notice that it is usually better to throw questions up for general discussion rather than count on the perpetually befuddled bruin for any kind of a reasonably timely response.

Ads for IBM clones regularly list SOCKET FOR 8087 as one of their amazing features. I used to smile at this selling of a socket (of air, really, a hole in the motherboard--why this is prostitution of the most reprehensible kind!), but recently, I've become seriously concerned. My ever-larger spreadsheets on Multiplan are taking an awfully long time to run. The new version 3.0 of Multiplan may help some, but will an 8087 help more? Alex Demaras, New York Victor Circle's hardware guru, just installed on the motherboard of my V9 the 8087 board I bought in one of Rosemary Abell's fire sales, and so far, I've not been able to detect any effects at all. Can someone tell me how the 8087 works?

Another side to my puzzlement comes from reading software flyers. Purchasers of ABSTAT must choose either an 8087 version or a non-8087 version of the statistics program; the 8087 version won't run, it seems, on machines lacking the 8087--what about the opposite? Will the 8088 version (so to speak) run on a V9 with an 8087? Then again, Borland's price list for Turbo Pascal 3.0 used to include an extra charge for 8087 support. What did you get for your money, a little program of some kind? Can that program be used to bring 8087 support to other programs (for example, the 8088 version of ABSTAT)?

I guess what I'm asking is, how does the 8087 get called in on a processing job? I would have thought that every time some math came into the 8088, it would be shunted over to the 8087, but evidently that's wrong--some signal has to come from the software telling the computer to compute with the 8087. Can I run some TSR program to make all my existing programs, from CALC.COM on up, use my shiny new 8087 chip?

Fuzzy Bear's overly simplistic answer to the above:

If you have an 8087 installed it is transparent to "non-8087" software, i.e., you can run other software and the 8087 has no effect whatsoever on that software.

If you have a software program which incorporates 8087

support it includes a routine which checks for the presence of the chip and shuts the appropriate tasks/computations through the 8087 for faster processing.

Many programming languages (e.g. Turbo Pascal) have 8087 checking and processing routines which may be incorporated in programs which you write. These routines do NOT, however, add 8087 support to programs which were not originally written to use the 8087 chip.

ABSTAT's method is unusual, to say the least. Most programs which use 8087 routines check for the chip and, if the chip is not present, run in "8087 emulation mode". Having separate 8087 and non-8087 versions sounds like a marketing scam to me.

The 8087 is \*NOT\* just for programs which are "number crunchers". If you want to see an amazing demonstration of the power of the chip, run AutoCAD on units with and without an 8087 card installed.

I just sold my last "extra" 8087 card to C.H. Carlisle. I've bought and installed the cards in six machines that have passed through the bear's den in the past year or two. As far as I know, Victor Technologies dumped all of their existing stock of the 8087 cards in one of the "close-out" sales some months back.

Good luck on trying to pry one out of the hands of existing V9000 owners now that there is real-world application (other than AutoCAD) for the cards.

Miscellaneous Single Message Topics

#96 BOB SWENDSEN, Subject: Printing fix for Emulator I've uploaded the file LTRHEAD.ARC to file area #6. This is a nice piece of Shareware for designing and printing letterheads (Epson only). It was written by Technical Service Associates. I found it on Franz Hirners' BAYVUG board. None of these niceties, however, have anything to do with my reasons for uploading it - rather I intended it to be used in case anyone wishes to verify a fix required to run it.

LETRHEAD.COM was clearly written to be run on the other machine. With EMIBM.COM it would go through the motions of working but not actually do anything - like printing letterheads. One of its features is to allow you to switch between parallel printer locations LPT1 and LPT2. Victor, of course, uses LST as its parallel printer locator not LPT1 or LPT2. We were able to fix this problem by going into MSDOS.SYS, finding the label "LST(blank)" and simply overwriting it with "LPT1". I don't know if this fix has been discussed before but, if so, I missed it. It looks useful enough to risk a second discussion anyhow.

(Continued on page 21)

# MS-DOS 3.1 - CHANGES FROM 2.11

[Among the numerous Victor Information Files and Technical Bulletins contained in Public Domain disk #83, the following file seemed particularly appropriate for publication here. It describes the enhancements to the Victor 9000 MS-DOS operating system from Version 2.11 to Version 3.1. Changes to BASICA are included. - Ed]

## Changes in MS-DOS from Ver 2.11 to Ver 3.1

MS-DOS 3.1 is used on both floppy disk and hard disk Victor 9000 systems. This version of MS-DOS is larger than 64K; you cannot boot many earlier floppy disk Victor 9000s with an operating system that is larger than 64K. If you are unable to boot your computer using this operating system diskette, you may need to update the Boot ROMs in your system to the Universal Boot ROMs, Part Numbers 107157-01 and 107158-01.

MS-DOS 3.1 contains these enhancements to MS-DOS 2.11:

- o Support for all of the following: MS-Networks, a large linker (up to 1MB), enhanced error recovery facilities, larger disks, and additional disk drives (A through P).
- o Additional utilities:
  - ANSI emulates the ANSI escape sequences common to PC-compatible computers.
  - ASSIGN, JOIN, and SUBST allow you to modify your drive and directory environment.
  - ATTRIB allows you to change the read/write attribute of files.
  - BACKUP and RESTORE allow you to back up and restore the hard disk (that is, to archive files).
  - Programming utilities DEBUG and EXE2BIN.
  - LABEL allows you to create, change, or delete volume labels.
  - New CONFIG.SYS commands FCBS and LASTDRIVE (MS-DOS 3.1 no longer supports the CONFIG.SYS commands SWITCHAR and AVAILDEV).
  - The UNIX-like utility UNIQ has been added, along with a new version of SEARCH.
  - Screen dump utilities SCRNMX80 and SCRNM84.
- GETSCRN and KILLSCRN allocate and deallocate memory for graphics screens.
- RENDIR allows you to rename directories.
- o The COMMAND, FORMAT, and HDPARK utilities have been enhanced.
- o A background PRINT utility has been enhanced to support path specifications and an internal programming interface.
- o The V BASICA interpreter and related programs ASYNCCOM and VBCONF have been added.
- o New Interrupt 21 functions and the file sharing program SHARE have been included to support file and record locking.
- o You can now reboot your computer from the keyboard by simultaneously pressing the ALT key, the LOCK key, and the Decimal key on the calculator keypad. You can press these keys in any order, but if you release one of them or press any other key before the three-key sequence is complete, the reboot command is canceled.
- o A reboot interrupt has been added at interrupt 69h. You can reboot the computer from within an application by issuing an interrupt 69h. No parameters are required for this interrupt.
- o A hot key interrupt has been added at interrupt 68h. When the keyboard scan codes are available from the hardware, they are passed in AX to interrupt 68h; interrupt 68h is called before any processing of the scan code occurs. An application that is monitoring this interrupt can do one of three things:
  - Return without changing any of the registers
  - Tell the BIOS to ignore the current key by returning zero in AX
  - Map the current key to another key by returning a different scan code in AX.
- o The MS-DOS 2.11 limit of 20 open handles per process is now more rigorously enforced in MS-DOS 3.1.
- o The configuration files for external hard disks have been corrected to work with a system that has only a single, external hard disk.

## New MS-DOS Utilities and Changes in Existing Utilities

**AUTOSET:** AUTOSET has been modified to support systems with 30MB internal hard disk drives. No earlier version of AUTOSET should be used with these systems. Hard disk drive configuration (CFG) files that begin with the number "3" are for use on 30-megabyte internal hard disk systems only.

**RENDIR:** RENDIR is similar to the REN command but is used to rename directories instead of files. Refer to the discussion of REN in your MS-DOS 3.1 Reference for the command syntax and examples of use. Do not use RENDIR if JOINs or SUBSTs are in effect.

**HDPARK:** HDPARK protects the hard disk by positioning the read/write heads over the innermost cylinder of the drive. It is recommended that you run HDPARK before turning the hard disk off, before moving the system, or any time power to the system is suspect (storms, power outages, and so on).

**SDCOPY:** Note that SDCOPY cannot copy a single-sided diskette to a double-sided formatted diskette. If you are copying a single-sided diskette, be sure to format a diskette single-sided for the destination diskette.

**SEARCH:** SEARCH provides the option AUTOTARX as a counterpart to the AUTOTARC option. It is similar to TARX in usage except that you do not have to specify the tar file name. AUTOTARX uses the same filenaming conventions as AUTOTARC.

**TAIL:** In the TAIL command, the /B option can be used in place of and is mutually exclusive with the /N option. The /B option must be followed by a decimal number specifying the number of bytes to list -- for example:

```
tail /b400
```

**SCRNMx80/SCRNM84 (Screen Dump Utility):** You can use the MS-DOS screen dump utility, SCRNMx80 or SCRNM84, to print a graphics or text screen on the Epson MX/FX or the Okidata Micro84 printers, respectively. Either utility can print text screens on any printer. The utility loads and stays resident; as long as the ASCII value of the print screen key is accessible from the keyboard, you can print a screen while running any application. Ver 1.2 of the screen dump utility works with all versions of MS-DOS 2.1 and 3.1 (except the 3.1 HD beta version).

There are two print screen keys: a Graphics Always key and a Text/Graphics key. Whenever you press the Graphics Always print screen key, the screen, whether in graphics or text mode, is sent to the printer in a graphics format specific to that printer. Whenever you press the Text/Graphics print screen key, the screen is sent to the printer in the printer's

text format when the screen is in text mode, and it is sent to the printer in graphics format when the screen is in graphics mode.

If you invoke the screen dump utility without parameters, the Graphics Always key defaults to Function Key 10, and the Text/Graphics key defaults to Function Key 9. You can change the location of the print screen keys by using the /K switch on the command line when you invoke the program. Following the /K, enter a space and two one- or two-digit hexadecimal numbers specifying the values of the new print screen keys. For example, to print the screen on your MX80 in graphics format whenever you press ALT-G, and in text/graphics format whenever you press ALT-T, use the command:

```
scrnmx80 /k 07 14
```

where 07 is the ASCII value of ALT-G, and 14 is the ASCII value of ALT-T.

If you invoke the appropriate screen dump command a second time, the program is uninstalled.

The screen dump utility is also activated by an INT 5. Thus, you can cause a screen dump from within a program by doing an INT 5. The screen dump utility has been linked to load high in memory. Thus, it will not use the memory needed for the graphics screens.

## Enhancements and Changes to BASICA

### The Bit Map

The graphics functions of BASICA require a bit map; a bit map is memory that is located in the lower 64K of RAM and is used for the graphics screen. The bit map for BASICA is usually allocated at runtime by the BASICA loader. However, if you use a load-and-stay-resident utility that loads in low memory, it will probably prevent the BASICA loader from allocating the bit map.

To overcome this problem, two utilities are provided that allocate and deallocate the memory for the bit map; these can be run at any time. To avoid conflicts with resident utilities, first run GETSCRN to reserve memory for the bit map and then load your utilities. You can then run BASICA at any time and it will be able to use the bit map you allocated earlier. If GETSCRN is run before BASICA, BASICA will recognize that memory for the bit map has been reserved and will not attempt to allocate another. You can return the reserved memory to the system by using KILLSCRN.

If you want to use the same technique for other graphics programs, you should understand how GETSCRN and BASIC communicate. GETSCRN allocates the bit map (if it can) and

stores the segment address of the bit map in the offset of interrupt D9 (location 0:364) and stores a "BM" (4D42) in the segment of interrupt D9 (location 0:366) to indicate a successful allocation. GETSCRN and BASICA check interrupt D9 with each invocation; if the bit map has already been allocated, they do not allocate another.

#### Differences Between Victor and IBM BASICA

Victor BASICA is compatible with IBM BASICA. Programs written in IBM BASICA should run under Victor BASICA; however, there are two differences between the BASICAs that you should note.

The first difference is in screen types. Victor BASICA has three screen types while IBM BASICA has one. If you have a color display unit, this difference should not affect the operation of BASICA because the color display unit is IBM compatible. Also, with an installed color display unit, Victor BASICA defaults to Screen 20 (color screen, text mode). Thus, as long as the program switches modes when needed (SCREEN 0,1, or 2), the program should run normally.

If you have a monochrome display unit, Victor BASICA defaults to Screen 40 (Victor screen, text mode). Screen 40 is not at IBM compatible screen; therefore, before the BASICA program accesses the screen, the screen mode should be changed to Screen 60 (IBM simulated screen, text mode). The easiest way to accomplish this is to add the statement SCREEN 60 to the BASICA program so that it is the first statement executed. Then, as long as the program switches modes when needed (SCREEN 0,1, or 2), the program should run normally.

The second difference between Victor BASICA and IBM BASICA is in the hardware. If a program attempts to directly access an IBM hardware feature that is unavailable on the Victor computer, the program will not work under Victor BASICA.

#### Additional Suggestions for Using VBASICA

BLOADs of a binary file from a hard disk to the color card do not work; only a part of the file is loaded. The IBM compatible color card does not work correctly with the hard disk controller because the controller attempts to write to the color card's memory using Direct Memory Access (DMA). You can overcome this problem by first reading the contents of the binary file into an array and then using POKE to move the contents of the array into the color card's memory. Color card memory begins at &hB800:0. BLOADs from diskette to the color card do work, since DMA is not used.

For some statements that include a boolean (logical true or

false) value or variable, the value used for true must be greater than zero but less than 256. For example, in the following LOCATE statement, the variable CRSR.ATTR% is a boolean value:

```
LOCATE LIN%, COL%, CRSR.ATTR%
If CRSR.ATTR% is true the cursor is on; if it is false the
cursor is off. If CRSR.ATTR% is true, it must be in the range
1 to 255. CRSR.ATTR% cannot be negative. The easiest
way to avoid problems is to use +1 for true and 0 for false.
```

\* \* \*

#### FROM THE BOARDS (Continued from page 18)

In this case the details of the fix were discovered by Bill Weinman, a Victor owning (but not modem owning) co-worker of mine. He had been having a problem with PC-FILE. The normal data printing functions would work but the SNAPSHOT feature wouldn't. Changing LST to LPT1 solved that problem. If anyone is having a similar nonprinting problem running under emulation you might give this fix a try.

Two suggestions if you intend to effect this change.

1) Be patient. MSDOS.SYS is about a 60K file and "LST" is buried well within it. In my case, the file started at 0013:0000 and I found "LST" at 0057:01E6. 2) The usual DEBUG or SST method of naming the file with "n", loading it with "l", making the change and then writing back to the file with "w" won't work. I don't know why. You have to use a DISK DISPLAY/MODIFY utility. I tried both SST (ver 1.2) and NORTON NU (ver 3.1) and both worked.

#688 RAY SIMKUS, Subject: v2 and power

I just recieved my 1.5 meg V2 board. It seems to work fine but after using the Victor for about half an hour it crashes. Then attempts to either reboot or to turn the computer off and then on seem to hangup at various points during the AUTOEXEC. Taking the V2 board out allows normal functioning. Putting it in results in it crashing again. I was told that I should check my power supply because it might be straining to supply enough power for the extra RAM. The power supply does get quite hot with the V2 board in. I did some carpentry on the back panel to move the fan directly behind the power transformer. This resulted in a cool power supply but the computer still hung after about half an hour of use.

The companies that used to service Victor in Vancouver no longer touch them. Is a weak power supply a likely cause for my problem and what can be done to check this out??? My V9000 has 256K and a 10meg hard disk. It has performed flawlessly for long hours for years.

\* \* \*

# THE LIBRARY REPORT / John Knobelsdorf

For a number of reasons, some more plausible than others, the "Library Committee" was very inactive between the end of April '87 and the middle of June '87, although we did manage to fill all of the orders for public domain disks.

In June, we got a private telephone line that was to have been installed the first week in May, and we have resumed activity with the various bulletin boards. That will permit us to collect additional public domain software for distribution.

For the benefit of new members, and those like me who have a tendency to forget details, I am repeating some observations that appeared in the March/April issue of VictorTalk concerning the various files that are on our public domain disks.

Some of the files on Victor\*Group public domain diskettes are in compressed form. Files having a filename extension of .?Q?, .LBR, or .ARC are squeezed, libaried or archived, and must either be unsqueezed or extracted from the library or the archive before they can be used.

Use NJSQ103 .COM (on PD #73) to "Unsqueeze" files.  
Use LUE220 .COM " to "Unlibrary" files.  
Use ARCE118 .COM " to "Unarchive" files  
on disks thru PD #82.  
Use PKXARC .COM (on PD #83) to "Unarchive" files  
on PD #83 and beyond.  
Use COMPFILS.TXT (on PD #73) for additional  
information.

Recent V\*G public domain disks have referred to ARC512.EXE for archive file creation, and ARCE118.COM for archive file extraction. Various of the SYSOP's and other persons who do a lot of uploading and downloading of files via telecommunications have begun using newer, similar programs for file handling. The "new" programs were developed by an individual named Phil Katz, and they do the job faster and more efficiently.

The new programs are PKARC.COM ver. 2.0, and PKXARC.COM ver. 3.4, and they are on PD #83. It is suggested that you read PKXARC.DOC, and use PKXARC.COM to unarchive PKARC.ARC to obtain PKARC.DOC and PKARC.COM.

Beginning with disk PD #83, most or all of the .ARC files and programs that appear on V\*G public domain disks will have been archived with PKARC.COM. It is recommended that you begin using PKXARC.COM for all your unarchiving activities. Current versions of PKARC.COM and PKXARC.COM are downward compatible with previous versions, and can also be used to unarchive .ARC files created with ARC512.EXE.

Please remember that in order to avoid some unexpected difficulties when extracting files from a "library" or "archive", you must have the library or archive on a separate diskette or in a separate subdirectory when it is being dismantled. Otherwise, during the extraction process from first one and then another archive on the same disk, some file may be inadvertently overwritten by another file of the same name having different content. A file named COMPFILS.TXT on PD #73 has additional information that should be helpful if you are unfamiliar with archived, libaried or squeezed files. [This file appears as an article in VictorTalk, Vol. 4, No. 5 (Nov-Dec 1986), page 18. - Ed.]

The release date for two new public domain disks is July 1, 1987. The disks are numbers 85 and 86. A copy of each is now on file in the Chapter set which is maintained by John Kimber, and they will be available for copying at the monthly meetings. They also may be ordered thru the mail in the usual manner.

The format we have been using in VictorTalk for description of newly released public domain disks has been changed. Beginning with Volume 5, Number 4 (this July/August 1987 issue), and the new disks PD #85 and #86, we will publish the list of files for each disk as it appears in their respective DISKID's.

The contents of PD #85 and #86 follow on the next two pages.

Those persons who wish to contact the Librarian about matters pertaining to public domain or the processing of orders for disks may write to John or Mary Jo Knobelsdorf, Rt. 2, Box 103, Bertram, Texas, 78605, or call us at 512/355-2084. Remember that all new orders must be sent to the V\*G address in Palo Alto, California for entry into the system. Do not send new orders directly to John or Mary Jo.



The contents of PD #85 are:

DISKID85.TXT	4608	Information about the files on this disk PD #85.
AFILTER .EXE	12544	AFILTER version 1.7 with FF expansion
ARCA122 .ARC	11264	Add/create ARC files quickly (incl. 80286 ver)
AUDIO .ARC	73257	Jan Ewing's incredible record/tape file system
AUDIO2 .ARC	54622	More files for above, w/diskid and instructions
BIGD .DOC	2688	Documentation for BIGD.EXE
BIGD .EXE	36864	Must dir util. for HD or multi-directory users
BRAD-WP .TXT	5760	Brad Chase on creating DOS text files with WP
CAT302 .ARC	40704	Victor PULSE Catalog 3.02 Update
CATS20 .ARC	20864	GREAT GENERIC DOS CATALOGING UTILITY
COLOR .ARC	1280	Set color in I mode on VI and +PC
CPUID .COM	1152	Identifies CPU chip and NDP, if installed
QWARE .ARC	3321	Revised QWARE.NDK for NewsDisk23
DL-HELP .TXT	4864	Explicit how-to-download, how-to-Kermit
DO .ARC	10240	String DOS commands on 1 line with ";"
DOSKNOW2.EXE	5760	A snapshot of the CPU; more info than ver 1.0
DSPACE .ARC	11392	Shows bar graph of sizes of sub-directories
ECHOARC .ARC	12288	Cnvt .ARCs to ASCII for transmission (and back)
ESC-BAT2.ARC	2287	Batch files for console; good version
FASTMX .TXT	4030	Enhancement notes for Epson MX80/100 Printers
FGREP140.ARC	7168	Chris Dunford's small, fast, MS-DOS grep utility
FORMGEN .ARC	52224	PCFILE custom output generator with mailmerge
FRMENU1.ARC	22016	PCFILE input customizer. Re-tailor with Turbo Pas
GAME22 .ARC	7680	Game update for NewsDisk22--Jan Ewing
GKXWORD .ARC	2345	Gundega Korst's Crossword; LIST/PRINT .GK files
GRAPH .C	8340	Graphics example using switchable graphics buffer
GRAPH .EXE	15616	Executable version of GRAPH.C
HDPLUSPC.ARC	4608	Victor HD can boot Plus-PC ( PKARC file)
HIRES .ASM	6438	Hi-lo graphics buffer selector
HIRES .COM	896	Executable version of HIRES.ASM
KILLDIR .COM	1536	Kill all sub-directories in a tree
KILLDIR .DOC	2048	Documentation for KILLDIR
LINE25 .ARC	12073	Ins and outs of the pesky Bottom Line
LTRHEAD .ARC	25984	PRINTS user defined letterheads
MATHSTAT.ARC	45184	Math and Statistics routines in BASIC
MENU23 .ARC	768	Correct MENU.DOS file for NewsDisk23

535849 Bytes in      36 Files      51200 Bytes free

The contents of disk PD #86 are:

DISKID86.TXT	4224	Information about the files on this disk PD #86.
ADDCOL .ARC	9600	Merges files by horizontally joining lines
BUZZWORD.TXT	1408	Ready Jargon maker; amusing
CHKDIR .ARC	11033	Like CHKDSK but for Directory; see DOC, use BAT
COLUMNS .ARC	16670	Rewrites a text file in columns as specified
MP3INS .ARC	31360	Data for V9000 installation of Multiplan 3.0
MULTO15 .ARC	101888	Marty's MULTiple Operations; ver 1.5; a DOS shell
P-BAT .ARC	3968	Prompt batch commands that effect the console
PC3-DOC .ARC	70656	Edited PC3.DOC (No blank pages), Jan
PDSW .ARC	2176	P. D. Sideways; modified for V9000 by Brad
PKFIND10.COM	16384	Katz's NEW FileFinder! Run to unpack contents
PKX34A20.COM	58368	PKXARC.COM ver 3.4 and PKXARC.ARC ver 2.0 in one file

PURSUIT .ARC	14329	PC Pursuit log-on script w/BBS options
QSORT20 .ARC	35840	Updated QSORT, with long lines and tagged blocks
QUIKREF .ARC	2688	PC File III Quick Reference Guide
RDCLK .C	2176	Read Valid Tech timer chip
REACTION.ARC	53248	Stand alone CHAIN REACTION game from NewsDisk22
RES .COM	256	Makes COMMAND.COM stay resident in RAM
SEVEN .BAS	1408	Wordstar to ASCII conversion in Basic
SMATHP2 .COM	2048	Lloyd's mod to SWP.COM for MathPlan; works
SMENU .ARC	33536	SUPER-MENU - loads and runs your applications
SPEEDY4 .ARC	2048	Chuck's latest DOS speed-up; works w/2.11 & 3.1
SUNTIME .EXE	43520	Use EMIBM to find sunrise/sunset times
SWITCH .ARC	57600	Stand alone version of SWITCH game from ND22
SWP14 .ARC	5120	Chuck Dayton's WP speedup; ver 1.4; works w/4.2
TEMPLATE.ARC	1280	Makes template editing keys match PC-DOS
TOSH .ARC	12544	Toshiba print utility with source code
USBBS .ARC	33664	Darwin's latest BBS list; ver 35

571648 Bytes in 29 Files 22528 Bytes free

\* \* \*

### VictorTalk

**Address inquiries to:** P. O. Box 50893  
Palo Alto, CA 94303

**Subscriptions:** Nancy Weaver

**Editor:** Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

**Contributors:** Al Bullock, Wendy Cykana, Roger Habeck, Franz Hirner, John Knobelsdorf, Mary Jo Knobelsdorf, Hermann Maseler, Bud Popken, Jack Prichard.

**VictorTalk** is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in **VictorTalk**. Copyright 1987 by Victor\*Group. Contents may be reproduced by permission only. **Permission is only a post card away.**

**VictorTalk** is an independent publication not affiliated in any way with Victor Technologies, Inc. **Victor** is a registered trademark of Victor Technologies, Inc.

**Submit Articles** on MSDOS disks in ASCII or Spellbinder format, with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

Yes, I want to join.

Enclosed is a check/money order for \$35.00 (foreign membership, \$40 in U.S. dollars) for my 1987 Victor\*Group membership.

Make checks payable to:

## VictorTalk

P. O. Box 50893  
Palo Alto, CA 94303

Name \_\_\_\_\_

City \_\_\_\_\_

Title \_\_\_\_\_

State \_\_\_\_\_ ZIP \_\_\_\_\_

Company \_\_\_\_\_

Country \_\_\_\_\_

Address \_\_\_\_\_

Phone \_\_\_\_\_

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
P.O. Box 50893  
Palo Alto, CA 94303

Date \_\_\_\_\_  
Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ & Zip \_\_\_\_\_  
Country \_\_\_\_\_ Phone \_\_\_\_\_

Office use only:  
Order rec'd \_\_\_\_\_  
Member check \_\_\_\_\_  
Comments \_\_\_\_\_  
Order shipped \_\_\_\_\_

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. Alternatively, see VictorTalk, Vol. 4, No. 2 for a brief description of disks 61-64; Vol. 4, No. 3 for disks 65-71; Vol. 4, No. 6 for disks 72-77; Vol. 5, No. 1 for disks 78-81; Vol. 5, No. 2 for disks 82-84; and Vol. 5, No. 4 for disks 85-86.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
21 22 23 24 25 26 27 28 29 30 31 32 33 34\* 35\* 36 37 38 39 40  
\*(34 & 35 are a two-disk set)  
41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56\* 57\* -- 59 60  
\*(56 & 57 are a two-disk set)  
61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80  
81 82 83 84 85 86 \_\_\_\_\_

1987 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, set of 6, \$12.00 ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_

1987 membership fee \$35 (receive all 1987 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR® TALK

## PUBLIC DOMAIN LIBRARY INDEX

The Public Domain Library Index is a comprehensive listing of books in the public domain. It includes titles, authors, and publication dates. The index is organized alphabetically by author's name. This is a valuable resource for researchers and collectors of public domain literature.

Author	Title	Year
A. A. Milne	Winnie-the-Pooh	1926
Alfred Hitchcock	The Birds	1963
Agatha Christie	Murder on the Nile	1937
...	...	...

**VICTOR\*GROUP**  
**P.O. BOX 50893**  
**Palo Alto, CA. 94303**

**FIRST CLASS MAIL**

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
P.O. Box 50893  
Palo Alto, CA 94303

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
State _____	
City _____ & Zip _____	
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. Alternatively, see VictorTalk, Vol. 4, No. 2 for a brief description of disks 61-64; Vol. 4, No. 3 for disks 65-71; Vol. 4, No. 6 for disks 72-77; Vol. 5, No. 1 for disks 78-81; Vol. 5, No. 2 for disks 82-84; Vol. 5, No. 4 for disks 85-86; and Vol. 5, No. 5 for disks 87-90.

- |    |    |    |    |    |    |    |    |    |    |    |    |    |     |                               |     |     |                               |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-------------------------------|-----|-----|-------------------------------|----|----|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14  | 15                            | 16  | 17  | 18                            | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34* | 35*                           | 36  | 37  | 38                            | 39 | 40 |
|    |    |    |    |    |    |    |    |    |    |    |    |    |     | *(34 & 35 are a two-disk set) |     |     |                               |    |    |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54  | 55                            | 56* | 57* | --                            | 59 | 60 |
|    |    |    |    |    |    |    |    |    |    |    |    |    |     |                               |     |     | *(56 & 57 are a two-disk set) |    |    |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74  | 75                            | 76  | 77  | 78                            | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | —  | —  | —  | —   | —                             | —   | —   | —                             | —  | —  |

1987 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, set of 6, \$12.00 ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_

1987 membership fee \$35 (receive all 1987 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders).. \$ \_\_\_\_\_

## PUBLIC DOMAIN LIBRARY INDEX

The Bay Area Victor Users Group maintains an extensive Public Domain library, now numbering 90 disks (including PD #58, which has been superceded by PD #77). As new disks are released, summaries of their contents are published in VictorTalk. [The Public Domain Software Mail Order Form, on the front inside cover (page 0) and the back inside cover (page 25), references these summaries for disks 61 thru 89.] A catalog (in two parts), available at no cost, contains the summaries for disks 1 thru 84. Certain Public Domain disks also contain cumulative alphabetical lists of all Public Domain filenames and the disk(s) in which the files are contained.

### Need for Index

Despite these various lists and summaries, finding some particular Public Domain file may not be easy. If the filename is known, the location of the file can be found using one of the cumulative listings, either by computer search with an appropriate utility program or by an "eyeball" search of the printed list. If only the filename extension is known, the file might also be located by computer search. (However, with many candidate files, in the case of common extensions such as .COM, .EXE, .BAS, etc., a large number of files must still be manually reviewed.) A completely manual search of a cumulative filename list for a particular filename extension is extremely tedious, with thousands of filenames to be examined.

If a user wants to find files of a particular category without knowing their specific filenames, little help is available. One could make guesses as to what the filenames might be or one could make a manual search of the summaries contained in the catalog. Unless one knows approximately the disk number, this search is also quite tedious. (The catalog is almost 16 pages long and is not easy to read.)

### Indexing Projects

In an attempt to improve this situation, two indexing projects have been started. The results of the first are embodied in this article. The second will eventually lead to a series of articles also to be published in VictorTalk. Separate articles will cover a number of categories that have been identified for cross-indexing purposes. For example, the first article of this series planned for publication will be on the subject of directories. It will summarize the characteristics of the many versions of directory utility programs that are available in the PD library (as well as those that are available with the various

versions of MS-DOS). Similar articles on such categories as Date/Time, Screen Control, Printer Control, Fonts/-Keyboards, Music, Games, Communications, Word Processing, and Data Bases are contemplated.

### Description of This Index

This index consists of three tables. The first two show which disks contain a given application category. The third simply lists the filename extensions that exist in the Public Domain (PD) library. While application category is the more useful way of indexing, at times filename extensions are helpful; hence, this type of indexing is also included. The first two tables show only if files of a specified application category exist on a specified disk. The third table only shows the meanings of (most of) the filename extensions that exist in the PD library (without regard to specific PD disk numbers). The user must then refer, for details on a particular disk's contents, to one or more of the following:

- o The summary given in the Public Domain Library Catalog
- o The summary given in the disk's DISKID file
- o The disk's directory

Table 1 lists the file categories by mnemonic symbol, briefly describes them, and lists the (PD) disk numbers of each disk containing one or more files in that category. Mnemonic symbols represent each (non-exclusive) file category. Where appropriate, the commonly-used filename extensions (.BAS, .BAT, .CHR, .KB, etc.) are used as application category symbols. However, we generally distinguish between file categories and filename extensions. For example, we distinguish between files in the category BAS (which are about the BASIC language) and files with the extension .BAS (which are written in some version of the BASIC language). Similarly, all files relating to character sets (fonts) would be categorized as CHR, although not all such files would necessarily be given the extension .CHR. Note that the categories are not mutually exclusive, so that a given file may be entered under two or more categories.

It is also planned to later use these same categories to index articles that have appeared in VictorTalk. (This joint use of the same categories was a strong input to the choice of categories and their mnemonic symbols.)

Table 2 is arranged as an array, of disk numbers vs. category, that indicates whether or not a given disk contains one or more files of a given category.

(Text continued on page 8)

TABLE 1. PUBLIC DOMAIN FILE CATEGORY DESCRIPTIONS AND DISK NUMBERS (Page 1 of 3)

Category	Description	PD Disk No.
ARC	ARChiving utilities and information about Backup and Restore, and file compression and expansion techniques for communications purposes. See also LBR and SQZ.	29 32 53 60 68 73 83 85 86 87
AUT	AUTOexec.bat programs	None through disk 90
BAS	BASic programs (including MS-BASIC, GW-BASIC, VBASICA, etc.) and tutorials on writing BASIC programs.	6 10 11 12 17 18 20 21 22 23 25 30 31 36 37 38 39 44 48 49 53 56 59 60 62 63 64 66 69 70 73 79 80 82 84 86 89
BAT	BATch files and tips on writing BATch files.	5 21 24 29 30 63 70 73 82 84 85 86
BUS	BUSiness and accounting software including accounts receivable, accounts payable, checkbook accounting, etc.	7 17 18 21 22 26 33 34 35 40 42 43 53 65 70 74
CHR	CHaRacter files and means of displaying and printing different character sets (fonts).	23 24 29 30 44 48 66 71 80 84 87
COM	CoMMunications, including transfer protocol, modems, bulletin board systems (example, FIDO) and communication software (examples, Vicmini, Kermit, etc.). Some COM software is further identified:	26 29 49 53 62 63 67 68 70 72 73 76 79 83 85 86 87
COM:FDO	FIDO BBS	67 68
COM:KER	KERmit	28 68 70 83 87
COM:VMI	VicMini	4 26 59 67 70 83
COM:XMD	XMDem	68
CPM	CP/M utilities	3 62 63 66 68 70 75
CVS	ConVerSion of programs (IBM to Victor), DOS2 TO DOS3, and programs to convert output (Wordstar to ASCII, ASCII to hex, etc.)	17 20 26 30 31 41 46 55 59 60 62 70 79 83 84 86 87
D/T	Date/Time. Programs for displaying, modifying, etc. date/time information (both at system level, and within various application programs), keeping diaries, calendars, and time records.	3 19 20 26 31 36 38 39 49 59 63 64 70 72 86
DB	DataBase programs. Most DB software is further identified:	18 25 60
DB:db2	dBASE II	32 33 34 35 40 41 65 74 75 76 84
DB:db3	dBASE III	65 75
DB:FEX	File Express	90
DB:PC3	PC-FILE III	77 85 86
DIA	DIAGnostic programs to test and find hardware and software problems.	2 9 13 31
DIR	DIRectory programs for displaying or printing lists of files or otherwise manipulating (searching, finding, modifying, sorting, re-dating, deleting, etc.) files.	3 4 8 11 18 19 20 21 24 31 36 37 38 39 44 45 46 49 50 60 61 72 85 86 87
DOS	Disk Operating Systems such as MS-DOS, PC-DOS, CP/M, XENIX, etc. and shells such as Marti Moleski's MULTO.	21 26 39 70 83 84 86 87
DTP	Desk Top Publishing, primarily what has become to be known as desk top publishing but includes information of a general nature related to publishing.	None through disk 90
EDI	EDITors, programs for line editing (such as EDLIN and similar shareware programs) but less than complete word processing programs (such as Wordstar, Spellbinder, etc.)	18 90
EDU	EDUcational programs, including typing tutorials, math problems, spelling flash cards, etc.	21 56 57 80
EMU	EMUlators, such as those emulating CP/M in MS-DOS, or emulating an IBM PC on a Victor 9000 computer.	45 62 63 70 75 77 79 87 88

TABLE 1. PUBLIC DOMAIN FILE CATEGORY DESCRIPTIONS AND DISK NUMBERS (Page 2 of 3)

Category	Description	PD Disk No.
FIL	FILE manipulation programs.	4 8 10 11 18 19 24 25 26 29 30 31 32 34 36 39 41 45 46 49 59 60 61 62 63 68 70 72 86 87 89 90
FIN	FINancial programs for making business calculations (amortization schedules, present value, future value, interest, etc.), forecasting, planning and budgeting and investment programs (such as Wall Street Weekly).	17 19 22 32 36 39 41 42 43 54 60 62 64 70 75
GAM	GAMES, including card games, board games, action games, narrative type games, ball games, crosswords, etc. (many of which include some graphics).	1 6 7 10 11 12 15 17 18 20 21 22 23 24 27 29 30 31 36 38 39 42 44 48 49 60 66 69 72 82 83 84 86
GEN	GENealogy programs, some use BASIC, some need dBASE II.	38 64
GRA	GRAphics, programs in which graphics are an important element (such as certain games and financial programs, or programs to facilitate the production of graphics other than Knee Toe Graphics.) All programs and information related to Tony Casano's Knee-Toe Graphics are identified with KTG.	4 6 14 16 17 20 22 23 24 29 30 44 45 49 59 60 66 69 71 72 80 81 85 88
HAR	HARdware, including V9000, +PC, and VI computers and computer system components such as CPU's, keyboards, monitors, disk drives, printers, plotters, the NEC 20 chip, etc.; and fixing hardware.	63 75 76 83 85 86 87
HLM	HLMor related to computers and computer users.	69 83 86 87
IND	INDEXes, including cross-indexes, tables of contents, disk indexing utilities, etc.	35 41 55 61 63 79 85
KB	KeyBOARDS, including means of changing the characters or functions associated with the keyboard keys and .KB files containing these keyboard assignments.	5 21 23 24 31 55 59 67 71 73 80 84 86 87
KTG	Knee Toe Graphics by Tony Casano.	60 71 81
LAN	Local Area Network software and related information.	None through disk 90
LBR	LIBRARY utilities for file compression and expansion.	4 26 31 36 46 49 60 67 73
LNG	LaNGuages excluding BASIC (.BAS) (which includes MS-BASIC, GW-BASIC, and VBASICA), whose programs and articles are sufficiently numerous to justify a separate category. Widely used and well represented languages are further identified:	37 38 56 61 72
LNG:ASM	ASSEMBly language (.ASM)	28 36 38 39 44 45 46 49 53 55 60 61 63 66 68 73 74 75 76 82 85 88 89
LNG:C	C language (.C)	27 36 37 38 49 53 60 74 75 83 85 86 88 89
LNG:FOR	FORtran (.FOR)	18
LNG:PAS	PAScal (.PAS)	19 26 36 38 49 53 59 62 63 78 87 88
LST	LISTs of user groups, bulletin boards, vendors, software, etc.	25 83 85 86
MAT	MAThematics, including mathematical and statistical computations (including the use of the 8087 series math chip).	21 26 32 36 38 43 53 56 57 60 62 63 66 76 82 85
MEM	MEMory and off-line storage, including RAM, ROM, floppy disks, hard disks, ram disks, the V9000 memory map, etc.	26 55 60 61 63 67 68 70 76
MIS	MIScellaneous, including BBS chit chat and any subject that does not reasonably fit in any other category.	50 51 52 80
MUS	MUSic and Sound, including audio output from the computer in any form (beeps, voice, noise effects, as well as music.)	6 7 17 20 22 23 27 45 60 66
NEW	NEWS of a general nature, as opposed to news specific to another category (such as VTI, news specific to Victor Technologies, Inc.)	None through disk 90



TABLE 1. PUBLIC DOMAIN FILE CATEGORY DESCRIPTIONS AND DISK NUMBERS (Page 3 of 3)

Category	Description	PD Disk No.
PAT	PATches and fixes, including information on the use of Disk Tools and DEBUG, etc, for restoring files and patching programs and the patch information itself.	49 53 55 60 63 67 73 74 84 85 87 88
PD	Public Domain program information, library catalogs, alphabetical listings of public domain files, etc.	25 45 59 60
PRN	PRINters, including dot matrix, daisy wheel, ink jet, laser types, etc. and including plotters, as well.	8 11 17 19 21 26 28 29 31 43 44 45 46 49 59 61 62 63 74 80 84 85 86 87
QJS	QJestionnaires.	None through disk 90
SAL	SALes of both hardware and software, including listing of materials for sale, generally with price and other specifics.	None through disk 90
SCR	SCReen utilities, including programs relating to the display of computer output and "screen dumps" (print screen on the IBM PC).	5 8 10 38 44 45 53 59 62 63 72 88 89
SPR	SPRead sheets, including shareware spreadsheets programs such as ASEASY, version 2.8 (on PD #90). Widely used and well represented spreadsheets are further identified:	30 41 90
SPR:123	Lotus 123	25 39 42 53 64 76
SPR:MP	Microsoft's Multiplan	30 41 84 86 87
SPR:MTH	WordPerfect's Mathplan	86
SPR:VC	VictorCalc	30 41
SQZ	SQzeeZe and unsqueeze utilities for file compression and expansion.	4 26 31 32 49 53 56 57 68 70 73
TEC	TECHnical notes, Victor's Additional Diskid's, tips and "how-to's" of all sorts that are not better suited for listing under a more specific category.	38 39 45 46 47 50 51 52 62 70 72 75 76 79 83 84 85 87 89
UTL	UTiLity programs of all sorts (examples: reset the computer from the keyboard, make the cursor blink, set and clear the 25th line, make menus for programs, etc.) other than those listed under a more specific category like GMM, CVS, DIA, DIR, PRN, SCR, etc. (CP/M utilities are listed under CP/M.)	4 5 7 8 10 11 17 18 20 21 25 26 28 29 30 31 32 36 38 40 41 43 44 45 46 49 53 54 55 59 60 61 62 63 70 72 74 75 76 79 82 84 85 86 87 89
VTI	Victor Technologies, Inc. news, including information on general subjects not better listed under a more specific category.	50 51 52
WP	Word Processing and utilities, including use of application programs such as Wordstar, Word Perfect, Spellbinder, etc. and utilities such as making mailing labels. Widely used and well represented application programs are further identified:	17 18 20 25 30 34 46 63 70 79 86
WP:PMA	PMAte	84 88
WP:SB	SpellBinder	None through disk 90
WP:WP	Word Perfect	85 86 88
WP:WS	WordStar	5 8 23 31 55 56 72 84

TABLE 2. PUBLIC DOMAIN FILES BY CATEGORY AND DISK NUMBER (Page 1 of 3)

PD Disk Number (Note: Disks 1, 2, 3, 9, 14, 15, and 16 are CP/M self-booting.)																																CATEGORY
CATEGORY	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	CATEGORY	
ARC																															X	ARC
AUT																																AUT
BAS						X				X	X	X					X	X		X	X	X	X	X							X	BAS
BAT					X																	X	X	X							X	BAT
BUS							X										X	X				X	X				X				X	BUS
CHR																																CHR
QMM:																											X				X	QMM:
QMM:FDO																																QMM:FDO
QMM:KER																													X			QMM:KER
QMM:WJI					X																						X					QMM:WJI
QMM:XMO																																QMM:XMO
CPM			X																													CPM
CVS																	X			X							X				X	CVS
D/T			X																X	X	X						X					D/T
DB:																		X								X						DB:
DB:db2																																DB:db2
DB:db3																																DB:db3
DB:FEX																																DB:FEX
DB:PC3																																DB:PC3
DIA		X							X			X																				DIA
DIR			X	X				X			X							X	X	X		X		X								DIR
DOS																					X						X					DOS
DTP																																DTP
EDI																			X													EDI
EDU																					X											EDU
BMU																																BMU
FIL				X				X		X	X							X	X	X				X	X	X	X		X	X	X	FIL
FIN																	X	X	X								X					FIN
GAM	X					X	X			X	X	X			X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	GAM
GEN				X		X						X				X	X			X	X	X	X						X	X		GEN
GRA														X		X	X					X	X	X								GRA
HAR																																HAR
HLM																																HLM
IND																																IND
KB					X																	X	X	X								KB
KTG																																KTG
LAN																																LAN
LBR				X																								X				LBR
LNG:																																LNG:
LNG:ASM																													X			LNG:ASM
LNG:C																												X				LNG:C
LNG:FOR																	X															LNG:FOR
LNG:PAS																		X									X					LNG:PAS
LST																																LST
MAT																											X					MAT
MEM																											X					MEM
MIS																																MIS
MUS						X	X										X		X		X	X						X				MUS
NEW																																NEW
PAT																																PAT
PD																											X					PD
PRN								X		X							X	X		X							X	X	X	X		PRN
QJS																																QJS
SAL																																SAL
SCR					X			X	X																							SCR
SPR:																																SPR:
SPR:123																																SPR:123
SPR:MP																																SPR:MP
SPR:MTH																																SPR:MTH
SPR:VC																																SPR:VC
SOZ				X																							X					SOZ
TEC																																TEC
UTL				X	X		X	X		X	X							X	X	X	X				X	X	X	X	X	X	X	UTL
VTI																																VTI
WP:																		X	X	X						X					X	WP:
WP:PMA																																WP:PMA
WP:SB																																WP:SB
WP:WP																																WP:WP
WP:WS					X			X																		X						WP:WS
CATEGORY	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	CATEGORY	

TABLE 2. PUBLIC DOMAIN FILES BY CATEGORY AND DISK NUMBER (Page 2 of 3)

CATEGORY	PD Disk Number																														CATEGORY			
	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60				
ARC		X																					X							X	ARC			
AUT																																AUT		
BAS	X					X	X	X	X				X				X	X					X				X		X	X	BAS			
BAT																																BAT		
BUS			X	X	X					X		X	X											X								BUS		
CHR																																CHR		
QMM:																																QMM:		
QMM:FDO																																QMM:FDO		
QMM:KER																																QMM:KER		
QMM:WJI																														X		QMM:WJI		
QMM:XMO																																QMM:XMO		
CPM																																CPM		
CVS	X										X					X										X				X		CVS		
D/T	X																			X										X		D/T		
DB:																														X		DB:		
DB:db2		X	X	X	X					X	X																					DB:db2		
DB:db3																																DB:db3		
DB:FEX																																DB:FEX		
DB:PC3																																DB:PC3		
DIA	X																															DIA		
DIR	X					X	X	X	X				X	X		X			X	X										X		DIR		
DOS																																	DOS	
DTP																																	DTP	
EDI																																	EDI	
EDU																											X	X					EDU	
EMU															X																		EMU	
FIL	X	X		X		X			X		X			X	X	X													X	X			FIL	
FIN		X				X			X		X	X	X											X					X				FIN	
GAM	X					X		X	X		X	X	X							X	X								X				GAM	
GEN									X																								GEN	
GRA													X	X															X	X			GRA	
HAR																																	HAR	
HLM																																	HLM	
IND					X						X																X						IND	
KB	X																										X			X			KB	
KTG																															X		KTG	
LAN																																	LAN	
LBR	X					X										X			X											X			LBR	
LNG:							X	X	X																									LNG:
LNG:ASM						X		X	X				X	X		X								X	X					X			LNG:ASM	
LNG:C						X	X	X																X						X			LNG:C	
LNG:FOR																																	LNG:FOR	
LNG:PAS						X		X																X					X				LNG:PAS	
LST																																	LST	
MAT		X				X		X					X											X			X	X		X			MAT	
MEM																															X			MEM
MIS																					X	X	X										MIS	
MUS														X																	X			MUS
NEW																																		NEW
PAT																						X		X	X					X				PAT
PD															X															X	X			PD
PRN	X												X	X	X		X													X				PRN
QUS																																		QUS
SAL																																		SAL
SCR									X					X	X										X					X				SCR
SPR:											X																							SPR:
SPR:123									X				X																					SPR:123
SPR:MP																																		SPR:MP
SPR:MTH																																		SPR:MTH
SPR:VC																																		SPR:VC
SQL	X	X																																SQL
TEC								X	X	X				X	X	X	X						X				X	X						TEC
UTL	X	X				X		X	X	X	X	X	X	X	X	X	X						X	X	X	X	X	X			X	X		UTL
VTI																																		VTI
WP:				X																														WP:
WP:PMA																																		WP:PMA
WP:SB																																		WP:SB
WP:WP																																		WP:WP
WP:WS	X																																	WP:WS
CATEGORY	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	CATEGORY			

TABLE 2. PUBLIC DOMAIN FILES BY CATEGORY AND DISK NUMBER (Page 3 of 3)

CATEGORY	PD Disk Number																		CATEGORY																	
	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78		79	80	81	82	83	84	85	86	87	88	89	90	CATEGORY				
ARC								X				X										X	X		X	X							ARC			
AUT																																		AUT		
BAS		X	X	X		X			X	X			X						X	X		X	X		X								BAS			
BAT				X						X			X									X	X	X		X							BAT			
BUS					X					X		X																						BUS		
CHR						X					X												X	X		X	X						CHR			
QMM:		X	X				X	X		X		X	X			X							X	X		X	X						QMM:			
QMM:FDO							X	X															X	X									QMM:FDO			
QMM:KER								X																X									QMM:KER			
QMM:WMI							X			X													X										QMM:WMI			
QMM:XMD								X															X										QMM:XMD			
CFM		X	X			X		X		X				X									X	X		X	X						CFM			
CVS		X								X													X	X		X	X						CVS			
D/T			X	X						X														X										D/T		
DB:																																		DB:		
DB:db2					X								X	X	X									X										DB:db2		
DB:db3					X																													DB:db3		
DB:FEX																																		DB:FEX		
DB:PC3																									X	X								DB:PC3		
DIA																																		DIA		
DIR	X										X													X	X	X	X							DIR		
DOS										X													X	X										DOS		
DTP																																			DTP	
EDI																																			EDI	
EDU																															X			EDU		
EMU																																			EMU	
FIL	X	X	X					X		X				X		X	X									X	X	X	X					FIL		
FIN		X							X																										FIN	
GAM						X			X														X	X	X		X							GAM		
GEN					X																														GEN	
GRA						X		X		X	X				X	X						X		X		X	X							GRA		
HAR															X	X							X	X		X	X							HAR		
HLM																								X			X	X							HLM	
IND	X		X																																IND	
KB							X				X	X												X			X	X							KB	
KTG																							X												KTG	
LAN																																			LAN	
LBR							X						X																						LBR	
LNG:	X											X																							LNG:	
LNG:ASM	X		X			X	X						X	X	X	X							X		X			X	X					LNG:ASM		
LNG:C														X	X								X	X		X	X	X	X					LNG:C		
LNG:FOR																																			LNG:FOR	
LNG:PAS		X	X																																LNG:PAS	
LST																																			LST	
MAT		X	X			X										X							X	X	X		X							MAT		
MEM	X		X				X	X		X						X																			MEM	
MIS																																			MIS	
MUS							X																												MUS	
NEW																																			NEW	
PAT			X				X				X	X												X	X		X	X							PAT	
PD																																			PD	
PRN	X	X	X										X											X	X		X	X							PRN	
QJS																																			QJS	
SAL																																			SAL	
SCR		X	X									X																							SCR	
SPR:																																			SPR:	
SPR:123					X																														SPR:123	
SPR:MP																									X		X	X							SPR:MP	
SPR:MTH																																			SPR:MTH	
SPR:VC																																			SPR:VC	
SQZ								X		X			X																						SQZ	
TEC		X								X		X	X	X	X									X	X	X	X	X	X						TEC	
UTL	X	X	X							X		X	X	X	X								X	X	X	X	X	X	X						UTL	
VTI																																			VTI	
WP:			X							X																									WP:	
WP:FMA																																			WP:FMA	
WP:SB																																				WP:SB
WP:WP																																				WP:WP
WP:WS																																				WP:WS

(Continued from page 1)

### File Extensions

In most cases, a file (type) extension is used with filenames. In fact, a bewildering assortment of file extensions is used. Since file extensions often help in giving clues to the purpose of a file, it is desirable to know the meaning of the file extension. Table 3 is an attempt to meet that need. It lists the file extensions appearing in the Public Domain library and is subdivided into three parts:

- o Extensions included in the Victor documentation (for MS-DOS and CP/M) as "MS-DOS File Extension Conventions" and "CP/M-86 File Extension Conventions". (Not all are used in PD library.)
- o Extensions in common use in various applications programs (i.e., Lotus, Spellbinder, etc.) and/or programming language usage.
- o Ad hoc extensions, not in common use. While the meaning of some of these is readily apparant (as in READ.ME and README.1ST); the meaning of others is cryptic and provides little or no help in deducing the kind of file. (Only a partial listing.)

Originally it was planned to give, for each disk, the number of files with each of the filename extensions used. However, this has been left for later, because of uncertainty of its usefulness and the considerable task of preparing such a list in a useful format.

### Help Needed

While considerable effort has gone into assembling the tables of this index, they undoubtedly contain a number of errors and omissions. Your help is solicited in discovering and correcting these. Further, suggestions for improvements in the table format and alternate approaches to solving this indexing problem would be greatly appreciated.

[An initial draft index was assembled by the editor. It was then reviewed and substantially improved by John and Mary Jo Knobelsdorf, whose considerable efforts are gratefully acknowledged. - Ed.]

\* \* \*

TABLE 3. PUBLIC DOMAIN FILENAME EXTENSION MEANINGS (Page 1 of 2)

#### Victor-Documented Filename Extensions

- \$\$\$ - Default extension assigned by computer under certain conditions
- A86 - 8086 assembly language source code
- ASM - ASSEMBLER (8086) language source code
- BAK - BACKUP file automatically produced by certain applications programs, (including Spellbinder)
- BAS - BASIC source code (includes MS-BASIC, GW-BASIC, and VBASICA files)
- BAT - Batch command file
- CMD - COMMAND file (executable under CP/M)
- COB - COBOL source code
- COM - COMMAND file (executable under MS-DOS)
- CRF - Cross-Reference listing file
- DAT - DATA file
- EXE - EXECUTABLE (relocatable) file
- FOR - FORTRAN source code
- H86 - Hex file produced by assembler (CP/M)
- INT - INTERMEDIATE compiled code file
- LIB - LIBRARY file
- LST - LISTING of compilation or assembly
- MAP - List file from linker
- OBJ - OBJECT code module (relocatable)
- OMR - OVERLAY module
- PAS - PASCAL source code
- PLM - PL/M source code
- PRN - PRINTOUT of compilation or assembly (or ASCII file)

- REF - Cross REFERENCE listing
- REL - RELOCATABLE object code module (CP/M)
- SUB - Command file executed by SUBMIT command (CP/M)
- SYM - SYMBOL table of assembly or compilation
- SYS - SYSTEM file
- XRF - Cross-Reference listing file (CP/M)

#### Filename Extensions in Common Use in Various Applications Programs, Etc.

- ARC - ARCHIVED (compressed) file
- ASC - ASCII file (directly readable by TYPE and similar commands)
- C - C language source file
- CHR - CHARACTER file or font installed with MDCON command
- DBF - DataBase File (dBASE)
- DIR - DIRECTORY (VSSI)
- DOC - DOCUMENTATION file (ASCII)
- KB - KEYBOARD file installed with MDCON command
- MAC - MACRO file
- NDX - INDEX file (dBASE)
- P - P code
- PRG - PROGRAM file, as used in dBASE II, etc.
- TMP - TEMPLATE for Lotus
- TXT - TEXT file (ASCII)
- WKS - WORKSHEET (Lotus)

TABLE 3. (Continued) PUBLIC DOMAIN FILENAME EXTENSION MEANINGS (Page 2 of 2)  
Ad Hoc Undocumented Filename Extensions (Partial Listing)

ABS - ABSTRACT (ASCII) (like DOC and TXT)	H1 -	PQN - Squeezed P*N file
ACG -	H2 -	PQS - Squeezed P*S file
ADV -	HBL -	PRO -
ALL -	HDR -	PV - Present Value
ANS - ANSWer messages (The Source PARTICipate BBS network)	HEX - HEXadecimal file	-----
APP -	HI - High	QQQ -
ART - ARTicle (i.e., for VictorTalk)	HLP - HeLP file	-----
ASH -	HOW -	RAW -
-----	HUM - HUMor file (ASCII)	REM - REMark
BBS - Bulletin Board System file	-----	REP - REPort
BOB -	IAT -	ROF -
-----	ID - IDentification (ASCII) file	RPT - RePorT
CAL -	INC -	-----
CBA -	INF - INFormation (like .DOC)	SAT -
CFG - ConFIguration file	INP -	SCI - aSCI file format
CI -	INX - INdeX file (ASCII)	SCO -
CNF - CoNfiguration file	IQ - Squeezed ID file	SCR -
CON - (50)	-----	SLD -
CTL -	KDL -	SMP -
-----	KEY -	SPC -
DEL - DELivery	-----	SRC -
DJA - Dow Jones Average (Wall Street Weekly)	LBL - LaBLE	SVB -
DLB -	LBR - LiBRaried file(s)	SVL -
DOG - Disk OrGanizer (DOG101A)	LOG -	SML -
DOS - File directly readable by DOS (ASCII file)	LOW - LOW	-----
DQC - Squeezed .DOC file	LQR - Squeezed LiBRary file	TIP - TIP as in .DOC, .INF (how to ___)
DRY -	LSP - LiSP source code)	TQN - Squeezed T*N file
DTA - DaTA file	-----	TQT - Squeezed T*T file
-----	MEM - MEMorandum	-----
FIL - FILE	MOD -	UNP - to UNProtect a file
FIV - (FIVE) Fifth language source file	MRG -	UNT -
FIX - FIX or program patch (to fix a bug)	MUT -	UPD -
FMT - ForMaT	-----	UWL -
FOS -	NAM -	-----
FRM - FoRM	NOT - NOTe file	VOC - VOiCe data file
FV - Future Value	NRO -	-----
-----	NYS -	WL -
GAM - GAMe file	-----	-----
GEN - GENerator	ORG -	
-----	OV2 - OverLay 2	
H -	OVL - OverLay file	
HO - (1-time use in ROFF.* files)	-----	
	PAR -	
	PAT - PATch for program (fix a bug)	
	PHN - PHoNe utility (i.e., a file in XTALK?)	
	PIC - PICTure file	
	PIE -	
	PLT -	

Note: Middle letter Q denotes squeezed file with same first and last letter as the regular extension. (i.e., ?Q?) (Ref: VictorTalk, Vol. 4, No. 5, p. 18)

\* \* \*

# MS-BASIC, Part 5 / Jack Prichard

Part 4 of this series (VictorTalk, Vol. 5, No. 4, Jul-Aug 1987), introduced the subject of data files. Simple examples were given of both sequential and random data files. Part 5 gives additional notes on random data files. (These might better have been included at the end of Part 4.) The examples (of Part 4) will be expanded with the introduction and explanation of a set of practical programs using both types of data files. These programs, additionally, will serve to illustrate some good programming practices.

## CAUTIONS ON USE OF RANDOM DATA FILES

When using a program such as TEST.BAS, introduced in Part 4 of this series, there is potential for several kinds of trouble. These possibilities arise from the provision for keyboard input of any desired record number.

1. When a record number is keyed in for reading data and that number corresponds to a record for which data has not yet been input, the data read will be blank if that portion of the disk was not previously written on. Otherwise, the most recently written data, although "deleted", will be read. If the data is in ASCII form, it will be readable on the screen. If the prior data contains certain control characters, the secure video mode will be invoked. This renders the displayed data unintelligible. To exit the secure video mode the computer must either be rebooted or the secure video mode otherwise turned off. One way of doing this is to return from MSBASIC to MSDOS level by typing ALT-C and SYSTEM and then running the command DECODE. (See Public Domain disk 59 or VictorTalk, Vol. 3, No. 9, p. 12.) [This ability to read "deleted" (but not yet written over) files provides the basis for recovering such files. The details of accomplishing this in a simple and practical way are beyond the intended scope of this article. However, working out these details would be an excellent exercise in the use of data files, etc.]

2. When a record number is keyed in (for output) and that number is larger than any previously keyed in, but within the capacity of the available disk space, the file size will be expanded accordingly. However, the file may now occupy an unintended large portion of the disk space. The only way to recoup this (perhaps inadvertently) assigned disk space is to delete the file and start over. Of course, if the file contains information that is to be kept, it must be read out and written into a new, appropriately shorter, file.

3. When too large a number is keyed in (for input), the disk may have inadequate free space for a file of the specified size. In this case, the message "Disk full in

\_\_\_\_\_ " will be displayed and program operation will be suspended.

4. When a record number that is larger than any previously recorded record is keyed in for output, the record will be shown as blank. However, the file size will not be expanded (as it would be for data input) and the record is not actually read (since it does not exist).

5. The maximum number of records is limited to 32,767 (the largest integer number available). If an attempt is made to input a larger number of records, an error message is displayed: "Input out of range".

## ADDITIONAL EXAMPLES OF SEQUENTIAL AND RANDOM DATA FILE USE

The additional examples of both types of data files will serve both to illustrate the use of data files in a practical working situation and to illustrate a number of useful programming practices and comment on their use in a practical program.

The examples chosen were written some time ago by the author for use in maintaining a log of computer usage. The files are contained in Public Domain disk 59; the set is comprised of six files whose names start with LOG, plus an AUTOEXEC.BAT file. The set of files is described in LOG.DOC, the essential parts of which follow:

AUTOEXEC.BAT - Provides automatic execution of log-on function after date and time have been entered after computer turn-on.

LOGON.BAT - Provides for manual initiation of log-on function if function is not to be provided by AUTOEXEC.BAT file.

LOGOFF.BAT - Provides for execution of the log-off function.

LOGON.BAS - Files date and time of log-on in file LOGON.DAT.

LOGOFF.BAS - Files date, time and purpose of usage in random file LOGOFF.DAT, first displaying last prior log entry and prospective new entry.

LOGDEP.BAS - Provides for reading LOGOFF.DAT (or any similar log-off file) to Display/Edit/Print the contents. More specifically, its functions are to:

- o Display the entire file.
- o Edit any desired file entry (to correct for mis-entered date, time or purpose or to add special notes).
- o Display and/or print an extended log-off file. Extension provides elapsed time, cumulative usage time and daily cumulative time.
- o Initialize the file before its first use; or reinitialize the file, to prepare for reuse, after copying and renaming the file to be saved.
- o Delete a record.
- o Insert a record.

- o The use of TOD is helpful, but not essential. While the command TOD was included in MS-DOS 1.25, it is not included in MS-DOS 2.11 and 3.1. However, its operation can be approximated by the following batch file in MS-DOS 3.1:  

```
A> type tod.bat
PROMPT $D $T
PROMPT
```
- o The command line A:MSBASIC LOGON must be the last line in this file. LOGON.BAS does not return control to the MS-DOS system level; it could if SYSTEM was added as its final command. However, even though this was done, AUTOEXEC.BAT would not be reinvoked, even though it contained command lines beyond A:MSBASIC LOGON.

## THE BATCH FILES

The batch files provide a simple means of entering MS-BASIC and running the logon and logoff programs. Their content follows: (It is assumed that all files are on drive a:.)

```
A> type autoexec.bat
DATE
TIME
TOD
A:MSBASIC LOGON
```

```
A> type logon.bat
A:MSBASIC LOGON
```

```
A> type logoff.bat
A:MSBASIC LOGOFF
```

### Notes on AUTOEXEC.BAT:

- o If your system includes a real-time clock, substitute the appropriate command line for DATE and TIME to set system time from the clock. (For example, use VT-ICLK with a Valid Technologies clock board.)

## LOGON.BAS

The LOGON.BAS program (Figure 1) is the same as the example given in Part 4, with these minor changes:

- o The program filename, date of last revision, and purpose have been added.
- o The date and time are truncated (using the function LEFT\$) to five characters each to save space on the display and printout for other items.
- o A display of log-on date and log-on time has been added.

(In earlier versions of the log-on program the functions DATE and TIME were used in place of the string functions DATE\$ and TIME\$. However, as was noted in earlier parts of this series, DATE and TIME were undocumented in the versions of MS-BASIC supplied with MS-DOS 1.25 and MS-DOS 2.11, and were subsequently dropped from VBASICA supplied with MS-DOS 3.1. Consequently, the program was changed to use the string functions.)

The data is input to the sequential data file using the command PRINT #1 with the explicit delimiter "," for no special reason.

FIGURE 1. LOGON.BAS

```
10 REM "LOGON.BAS" 10/14/85 - DISPLAYS AND FILES LOG-ON DATE AND TIME
20 DTE.CN$=LEFT$(DATE$,5)
30 TME.CN$=LEFT$(TIME$,5)
40 OPEN "O",#1, "LOGON.DAT"
50 PRINT #1, DTE.CN$; ", "; TME.CN$
60 PRINT "LOG-ON DATE = "; DTE.CN$; " LOG-ON TIME = "; TME.CN$
70 END
```



## LOGOFF.BAS

LOGOFF.BAS (Figure 2) is a somewhat longer and more complex program than LOGON.BAS. It reads the LOGON.DAT (sequential data) file created by LOGON.BAS; it both reads and writes to the LOGOFF.DAT (random data) file created by prior runnings of LOGOFF.BAS. The program

is divided into two parts: the main program (which starts with remarks, initialization and definitions) and five short subprograms (one for error handling and four for display formatting). To facilitate locating these subprograms, they are set off by a row of asterisks (usually with a title embedded in the asterisks).

FIGURE 2. LOGOFF.BAS (Page 1 of 2)

```

10 REM "LOGOFF.BAS" 10/14/85 - FILES DATE & TIME OF LOG-OFF & PURPOSE
20 I.MAX=100                                'MAX NO. OF ENTRIES
30 ESC$=CHR$(27)
40 DEF FNM$(LIN,COL)=ESC$+"Y"+CHR$(31+LIN)+CHR$(31+COL) 'MOVE CURSOR
50 CS$=ESC$+"E"                              'CLEAR SCREEN
60 S1$="## \ \ \ \ \ \ \ \ \"             'DISPLAY FORMAT 1
70 S2$="\ \ \"                               'DISPLAY FORMAT 2
80 ON ERROR GOTO 520
90 OPEN "I",#1, "LOGON.DAT"
100 INPUT #1, DTE.ON$, TME.ON$
110 OPEN "R",#2, "LOGOFF.DAT",61
120 FIELD #2, 1 AS F$,5 AS D.ON$,5 AS T.ON$,5 AS D.OFF$,5 AS T.OFF$,40 AS P$
130 FOR I=1 TO I.MAX
140   GET #2, I
150   IF ASC(F$)=255 THEN 170
160 NEXT I
170 IF I<I.MAX-3 THEN 260
180 PRINT: PRINT
190 PRINT " * * * CAUTION - File is nearly full * * *"
200 PRINT
210 PRINT "   Rename file LOGOFF.DAT to desired permanent file name"
220 PRINT "   and reinitialize file (destroying data ), using LOGDEP.BAS."
230 PRINT "   Alternatively, change line 20 to increase file size."
240 PRINT: INPUT "   Press CR to continue ", A$
250 PRINT: PRINT
260 I.LAST=I-1
270 IF I.LAST=0 THEN 290
280 GET #2, I.LAST
290 I=I.LAST
300 GOSUB 580                                'DISPLAY HEADING
310 GOSUB 640                                'DISPLAY COLUMN HEADINGS
320 IF I.LAST<0 THEN 340
330 PRINT "FILE EMPTY * * * * * * * * * * * * * * * * *"
340 GOSUB 680                                'DISPLAY LAST PRIOR LOG ENTRY
350 GOSUB 720                                'DISPLAY POTENTIAL NEXT LOG ENTRY
360 PRINT
370 INPUT "PURPOSE OF USAGE (MAX. OF 40 CHAR.) "; PURPOSE$
380 PRINT
390 DTE$=LEFT$(DATE$,5)                      'MMDD PART OF MMDD-YY OF DATE$
400 TME$=LEFT$(TIME$,5)                      'HHMM PART OF HHMM:SS OF TIME$
410 GOSUB 720                                'DISPLAY NEW LOG ENTRY AS FILED
420 PRINT FNM$(22,1);
430 LSET F$=CHR$(0)
440 LSET D.ON$=DTE.ON$

```

Line 10 (as in LOGON.BAS) gives the filename, date of last revision, and purpose of the program

Line 20 sets the size of LOGOFF.DAT at 100 records. This is done to reserve space for the file to reduce fragmentation of the file that would otherwise occur as the size of the file was gradually increased with each additional entry. This size is otherwise quite arbitrary; it may be increased or decreased, as desired.

Lines 30-70 define strings that are later used for displays, as was done in the programs of Part 2.

Line 80 enables error trapping and specifies the first line of the error handling routine.

Lines 90-100 read the log-on data from LOGON.DAT.

Lines 110-160 open the random data file, defines the data fields and reads through the file to locate the end of the record entries. Note that, prior to its initial use, LOGOFF.DAT must be initialized (F\$ set to CHR\$(255) and the remaining fields set to blank) by LOGDEP.BAS. This both provides a means for recognizing the end of the active part of the file and for really deleting any data left by a "deleted" prior file using the same disk space.

Lines 170-250 provide a display warning when the file is within three records of full.

Lines 260-290 reads the last record recorded, if there is one.

Lines 300-360 displays the last prior log entry and potential new log entry; if there is no prior log entry "FILE EMPTY" is first displayed.

Line 370 provides for keyboard input of purpose of usage.

Lines 380-490 truncate the current date and time to five characters (year and seconds omitted),

FIGURE 2. LOGOFF.BAS (Page 2 of 2)

```

450 LSET T.ON$=TIME.ON$
460 LSET D.OFF$=DTE$
470 LSET T.OFF$=TIME$
480 LSET P$=PURPOSE$
490 PUT #2, I.LAST+1          'FILE NEW ENTRY
500 PRINT "LOGOFF COMPLETED": END
510 ***** ERROR ROUTINES *****
520 IF ERR=53 AND ERL=90 THEN 530 ELSE 550
530 PRINT "NO LOGON.DAT FILE * * * * *
540 STOP
550 PRINT "ERROR OTHER THAN #53 (FILE NOT FOUND) ERR=", ERR, " ERL=", ERL
560 STOP
570 ***** SUBPROGRAMS - DISPLAY DATA *****
580 PRINT CS$;                'CLEAR SCREEN
590 PRINT "LAST PRIOR AND POTENTIAL NEXT LOG ENTRY:": PRINT
600 DTE$=LEFT$(DATE$,5)        'MM/DD PART OF MM/DD-YY OF DATE$
610 TIME$=LEFT$(TIME$,5)      'HH:MM PART OF HH:MM:SS OF TIME$
620 RETURN
630 *****
640 PRINT "ENTRY DATE   TIME   DATE   TIME   PURPOSE"
650 PRINT " NO.   ON   ON   OFF   OFF"
660 RETURN
670 *****
680 PRINT USING S1$;I.D.ON$;T.ON$;D.OFF$;T.OFF$;
690 PRINT USING S2$;P$
700 RETURN
710 *****
720 PRINT FNM$(6,1);
730 PRINT USING S1$;I.LAST+1;DTE.ON$;TIME.ON$;DTE$;TIME$;
740 PRINT USING S2$;PURPOSE$
750 RETURN

```

Line 500 displays "LOGOFF COMPLETED" and ends the program (closing all files). Note that this message is displayed on line 22 of the display (by program line 420) to set it off from the log entry display.

Lines 510-560 provide a display "NO LOGON.DAT FILE" when there is none and a display of error number and error line for other types of errors; in both cases the program stops.

Lines 570-750 format the display of the prior and new log entries. Note that the cursor is relocated by line 720 so that the new log entry is re-displayed as actually written.

LOGDEP.BAS

LOGDEP.BAS (LOG Display/Edit/Print) is the longest and most complex of the three LOG\*.BAS programs. It deals only with the random data file LOGOFF.DAT (or other keyboard-specified .DAT file; not with the sequential data file LOGON.DAT). The program is divided into two parts: the main program and nine subprograms, set off by a

row of asterisks (and usually a title). The program proper (like LOGOFF.BAS) starts with remarks, initialization and definitions and then reads the random data file to determine the end of data entries. It then continues with a menu for selecting one of eight possible functions. These functions are written as separate program modules set off (like the subprograms) by a row of asterisks surrounding the module title (except the exit module, which is untitled).

FIGURE 3. LOGDEP.BAS (Page 1 of 7)

```

10 REM "LOGDEP.BAS" 10/14/85 - DISPLAY/EDIT/PRINT LOG-OFF DATA
20 I.MAX=100                'MAX NO. OF ENTRIES; IF CHANGED, ALSO CHANGE DIM'S
30 ESC$=CHR$(27)
40 DEF FNM$(LIN,COL)=ESC$+"Y"+CHR$(31+LIN)+CHR$(31+COL) 'MOVE CURSOR
50 CS$=ESC$+"E"           'CLEAR SCREEN
60 EES$=ESC$+"J"         'ERASE TO END OF SCREEN
70 EL$=ESC$+"I"         'ERASE LINE

```

Line 10 (as in LOGON.BAS and LOGOFF.BAS) gives the filename, date of last revision, and purpose of the program.

Line 20 (as in LOGOFF.BAS) sets the size of LOGOFF.DAT at 100 records. This parameter should be changed to agree with I.MAX in LOGOFF.BAS, if it is changed.

FIGURE 3. LOGDEP.BAS (Page 2 of 7)

```

80 RLF$=ESC$+"A" 'REVERSE LINE FEED
90 FF$=CHR$(12) 'FORM FEED
100 S1$=" ## \ | | | | | | | " 'DISPLAY FORMAT 1
110 S2$="\ |" 'DISPLAY FORMAT 2
120 LINE.MAX=50 'MAX DATA LINES/PAGE
130 DIM D.ON$(100), T.ON$(100), D.OFF$(100), T.OFF$(100), PURPOSE$(100)
140 T1$=" \ | | || \###.####.#" 'DISPLAY/PRINTOUT FORMATS
150 T2$="\ ||"
160 T3$="###.# \ |"
170 ON ERROR GOTO 2020
180 INPUT "NORMAL LOGOFF.DAT FILE? (Y/N) ", A$
190 IF A$="Y" OR A$="y" THEN F.NAME$="LOGOFF": GOTO 210
200 INPUT "ENTER FILENAME (.DAT EXTENSION WILL BE APPENDED) ", F.NAME$
210 F.NAME$=F.NAME$+".DAT"
220 PRINT "FULL FILENAME = "; F.NAME$
230 OPEN "R", #2, F.NAME$, 61
240 FIELD #2, 1 AS F$, 5 AS D.ON$, 5 AS T.ON$, 5 AS D.OFF$, 5 AS T.OFF$, 40 AS P$
250 FOR I=1 TO I.MAX
260 GET #2, I
270 IF ASC(F$)=255 THEN 290
280 NEXT I
290 I.LAST=I-1
300 ***** MENU *****
310 PRINT CS$;
320 PRINT "SELECT ONE OF FOLLOWING:"
330 PRINT " 1 - DISPLAY ENTIRE LOGOFF FILE"
340 PRINT " 2 - EDIT A LOGOFF ENTRY"
350 PRINT " 3 - DISPLAY EXTENDED LOG-OFF FILE"
360 PRINT " 4 - PRINT EXTENDED LOG-OFF FILE"
370 PRINT " 5 - INITIALIZE FILE (DESTROYS DATA IN IT)"
380 PRINT " DEL - DELETE A RECORD"
390 PRINT " INS - INSERT A RECORD"
400 PRINT " X - EXIT"
410 PRINT: INPUT "SELECTION "; A$
420 IF A$="X" OR A$="x" THEN 620
430 ON VAL(A$) GOTO 520, 650, 1060, 470, 1480
440 IF A$="DEL" THEN 1570
450 IF A$="INS" THEN 1730
460 GOTO 310
470 PRT=>1: GOTO 1060 'PRINT FLAG ON
480 ***** PAUSE AND RETURN TO MENU *****
490 PRINT: INPUT "PRESS CR TO CONTINUE ", A
500 GOTO 310
510 ***** DISPLAY ENTIRE FILE *****
520 PRINT CS$;
530 GOSUB 1980
540 GOSUB 2060 'PRINT HEADING
550 FOR I=1 TO I.LAST
560 GET #2, I
570 IF ASC(F$)=255 THEN 600
580 GOSUB 2100
590 NEXT I

```

Lines 30-110 and 140-160 define strings that are later used for displays, much as in LOGOFF.BAS.

Line 120 sets the maximum number of data lines per page of printout.

Line 130 dimensions the one-dimensional arrays used for temporary storage of the record data fields. (This dimension should match I.MAX.)

Line 170 enables error trapping and specifies the first line of the error handling routine.

Lines 180-220 provide for specifying a .DAT filename other than the default LOGOFF.DAT. Note that line 190 accepts either "Y" or "y" as a yes answer and that line 210 automatically appends the filename extension .DAT. As a check, line 220 displays the full filename (including extension).

Lines 230-290, essentially as in LOGOFF.BAS, opens the random data file, defines the data fields, and reads through the file to locate the end of the record entries.

Lines 300-470 present the menu of eight possible functions and transfers control to the appropriate subprogram after the selection is keyed in. (Note that both "X" and "x" are acceptable for exiting the program)

Lines 480-500 are used as the re-entry point after all selections, except edit, to allow the display to be viewed as long as desired before it is cleared and replaced by the menu.

Lines 510-600 displays the entire log-off file. The displays scrolls until the end is reached. Of course, the display may be stopped by pressing Alt-S, and restarted by pressing it again.

FIGURE 3. LOGDEP.BAS (Page 3 of 7)

```

600 GOTO 490
610 !*****
620 PRINT FN$(22,1);
630 END
640 !***** EDIT *****
650 GOSUB 1980 'CHECK IF FILE EMPTY
660 PRINT: INPUT "ENTRY NO. TO BE EDITED "; I
670 IF I<1 OR I>I.LAST THEN 660
680 GET #2, I
690 PRINT CS$;
700 PRINT "PRESENT AND PROPOSED (EDITED) VERSION OF ENTRY:": PRINT
710 GOSUB 2060 'DISPLAY COLUMN HEADINGS
720 GOSUB 2100 'DISPLAY LINE TO BE EDITED
730 ND.ON$=D.ON$
740 NT.ON$=T.ON$
750 ND.OFF$=D.OFF$
760 NT.OFF$=T.OFF$
770 NP$=P$
780 LIN=8
790 GOSUB 2140 'DISPLAY PROPOSED VERSION
800 PRINT "IN FOLLOWING EDITING, CR ALONE LEAVES ENTRY ITEM UNCHANGED": PRINT
810 INPUT "DATEON "; A$: IF A$="" THEN 840 ELSE GOSUB 2200
820 IF INVAL THEN PRINT RLF$;; PRINT EL$;; GOTO 810
830 ND.ON$=A$: LIN=11: GOSUB 2140
840 INPUT "TIMEON "; A$: IF A$="" THEN 870 ELSE GOSUB 2340
850 IF INVAL THEN PRINT RLF$;; PRINT EL$;; GOTO 840
860 NT.ON$=A$: LIN=12: GOSUB 2140
870 INPUT "DATEOFF "; A$: IF A$="" THEN 900 ELSE GOSUB 2200
880 IF INVAL THEN PRINT RLF$;; PRINT EL$;; GOTO 870
890 ND.OFF$=A$: LIN=13: GOSUB 2140
900 INPUT "TIMEOFF "; A$: IF A$="" THEN 930 ELSE GOSUB 2340
910 IF INVAL THEN PRINT RLF$;; PRINT EL$;; GOTO 900
920 NT.OFF$=A$: LIN=14: GOSUB 2140
930 PRINT "PURPOSE (MAX. OF 40 CHR.)";
940 INPUT " "; A$: IF A$<>" " THEN NP$=A$: LIN=15: GOSUB 2140
950 PRINT: INPUT "DONE EDITING THIS ENTRY NO.? (Y/N) ", A$
960 PRINT: PRINT FN$(8,1); EES$;
970 IF A$<>"Y" AND A$<>"y" THEN 800
980 LSET D.ON$=ND.ON$
990 LSET T.ON$=NT.ON$
1000 LSET D.OFF$=ND.OFF$
1010 LSET T.OFF$=NT.OFF$
1020 LSET P$=NP$
1030 PUT #2, I
1040 GOTO 310
1050 !***** DISPLAY/PRINT EXTENDED DATA *****
1060 GOSUB 1980 'CHECK IF FILE EMPTY
1070 FOR I=1 TO I.MAX
1080 GET #2, I
1090 IF ASC(F$)=255 THEN 1230
1100 D.ON$(I) =D.ON$
1110 T.ON$(I) =T.ON$

```

Lines 610-630 provide the exit from this program. If control is to be returned to the MS-DOS system level automatically, line 630 could be changed to SYSTEM.

Lines 640-1040 provide for editing any specified record. It demonstrates the full flexibility of the random data file in calling up any one record, changing any or all fields, and filing the edited record without touching any other record.

Line 670 insures that the record number to be edited is within the range of the previously recorded records.

Line 700 displays lines 1 and 2 of the edit display format.

Line 710 (via subprogram) displays lines 3 and 4 (column headings).

Line 720 (via subprogram) displays line 5 (present version).

Line 790 (via subprogram) displays line 6 (proposed version), leaving cursor on line 8, column 1.

Line 800 displays instructions for editing.

Line 810 allows entering a new date, which is checked for validity by lines 2200-2320. If date is valid, line 5 (proposed version) is erased.

Line 820 (applicable if proposed date is invalid) erases DATEON prompt and redisplay prompt, allowing another try at providing a valid entry.

Line 830 (applicable if proposed date is valid) redisplay proposed version of record with accepted editing.

Lines 840-940 similarly allow entering new TIMEON, DATEOFF, TIMEOFF, and PURPOSE with appropriate validity checks (except for PURPOSE). In all editing, the entry "CR" is interpreted as no change desired.

Lines 950-970 allow the editing cycle to be repeated if the user is not satisfied with the editing accomplished. Again, note that either "Y" or "y" is acceptable as a "yes" answer.

Lines 980-1040 record the edited record and return to the menu display.

FIGURE 3. LOGDEP.BAS (Page 4 of 7)

```

1120 D.OFF$(1) =D.OFF$
1130 T.OFF$(1) =T.OFF$
1140 PURPOSE$(1)=P$
1150 IF I > 1 THEN 1220
1160 IF D.ON$="" THEN 1190
1170 IF D.ON$ < "01-01" OR D.ON$ > "12-31" THEN 1180 ELSE 1190
1180 PRINT: PRINT "DATE ON OUT OF RANGE (ON RECORD NO. 1): =" ; D.ON$; STOP
1190 IF T.ON$="" THEN 1220
1200 IF T.ON$ < "00:00" OR T.ON$ > "23:59" THEN 1210 ELSE 1220
1210 PRINT: PRINT "TIME ON OUT OF RANGE (ON RECORD NO. 1): =" ; T.ON$; STOP
1220 NEXT I
1230 PRINT: INPUT "TITLE (OR FOR TITLE AS PREVIOUSLY ENTERED) = "; A$
1240 IF A$="" THEN 1260
1250 TITLE$=A$
1260 LTH=.5*LEN(TITLE$)
1270 CUM.TIME=0: CUM.TIME=0: LINE.NR=0: PAGE=1 'INITIALIZE FOR DISPLAY/PRINT
1280 FOR I=1 TO I.LAST
1290 IF LINE.NR < LINE.MAX THEN 1320
1300 LINE.NR=0: PAGE=PAGE+1
1310 IF PRT THEN LPRINT FF$ 'FORM-FEED FOR NEW PAGE
1320 LINE.NR=LINE.NR+1
1330 IF LINE.NR > 1 THEN 1360
1340 PRINT CS$: PRINT: PRINT 'CLEAR SCREEN
1350 GOSUB 2520 'DISPLAY/PRINT HEADINGS
1360 H.OFF=VAL(LEFT$(T.OFF$(1),2))+VAL(RIGHT$(T.OFF$(1),2))/60
1370 H.ON =VAL(LEFT$(T.ON$(1),2))+VAL(RIGHT$(T.ON$(1),2))/60
1380 IF D.ON$(1) > D.ON$(1) THEN H.OFF=H.OFF+24
1390 IF D.OFF$(1) = "01-01" AND D.ON$(1) = "12-31" THEN H.OFF=H.OFF+24
1400 EL.TIME=H.OFF-H.ON 'ELAPSED TIME
1410 CUM.TIME=CUM.TIME+EL.TIME 'CUMULATIVE TIME
1420 CUMD.TIME=CUMD.TIME+EL.TIME 'CUMULATIVE TIME FOR DATE
1430 GOSUB 2660 'DISPLAY/PRINT DATA
1440 NEXT I
1450 IF PRT THEN LPRINT FF$ 'FORM-FEED AT END OF PRINTOUT
1460 PRT=0 'PRINT FLAG OFF
1470 GOTO 490
1480 '***** INITIALIZE FILE *****
1490 INPUT "ARE YOU SURE? (DATA WILL BE DESTROYED) (Y/N) ", A$
1500 IF A$ <> "Y" AND A$ <> "y" THEN 310
1510 PRINT "INITIALIZING FILE * * * * * "
1520 GOSUB 1900
1530 FOR I=1 TO I.MAX: PUT #2, I: NEXT I
1540 PRINT "INITIALIZATION COMPLETED * * * * * "
1550 GOTO 490
1560 '***** DELETE A RECORD *****
1570 GOSUB 1980 'CHECK IF FILE EMPTY
1580 PRINT
1590 INPUT "RECORD NUMBER TO BE DELETED = ", A
1600 IF A > 0 AND A < I.LAST+1 THEN 1630
1610 PRINT "RECORD NUMBER OUT OF RANGE * * * * * "
1620 GOTO 1590
1630 I.DEL=A

```

Lines 1050-1470 allow for either displaying or printing the log with extended data. Extended data means the addition of columns giving cumulative time (in hours and tenths) and daily time subtotals (in hours and tenths).

Lines 1170-1210 check that date on and time on of the first record are not obviously in error before proceeding.

Lines 1480-1550 initialize the file space prior to the first use of LOGOFF.DAT or prior to the reuse of an old file. Note that the user is asked "ARE YOU SURE" so that he has a second chance to avoid destroying his data if initialization is inadvertently invoked. Initialization stores CHR\$(255) in field F\$ and blanks in all other fields. Note that, while initialization is in progress a "INITIALIZING FILE" message is displayed. (This is a good practice when any potentially time-consuming operation is in progress. It puts the user at ease by letting him know that the computer is doing its thing.)

Lines 1560-1710 allows any specified record within the established record numbers to be deleted. All higher records are moved down one record number and the highest record number is reinitialized.



FIGURE 3. LOGDEP.BAS (Page 6 of 7)

```

2160 PRINT USING S2$;NP$;
2170 PRINT FM$(LIN,1);
2180 RETURN
2190 ***** SUBPROGRAM - CHECK DATE *****
2200 INVAL=0
2210 IF LEN(A$)=3 THEN A$="0"+LEFT$(A$,2)+"0"+RIGHT$(A$,1)
2220 IF LEN(A$)<4 THEN 2250
2230 IF MID$(A$,3,1)="-" THEN A$=LEFT$(A$,3)+"0"+RIGHT$(A$,1)
2240 IF MID$(A$,2,1)="-" THEN A$="0"+A$
2250 IF LEN(A$)>5 THEN 2320
2260 GOSUB 2480
2270 IF M1$<"0" OR M1$>"1" OR M2$<"0" OR M2$>"9" OR M3$<"-" THEN 2320
2280 IF M4$<"0" OR M4$>"3" OR M5$<"0" OR M5$>"9" THEN 2320
2290 Z$=LEFT$(A$,2): IF Z$<"01" OR Z$>"12" THEN 2320
2300 Z$=RIGHT$(A$,2): IF Z$<"01" OR Z$>"31" THEN 2320
2310 RETURN
2320 INVAL=1: RETURN
2330 ***** SUBPROGRAM - CHECK TIME *****
2340 INVAL=0
2350 IF LEN(A$)=3 THEN A$="0"+LEFT$(A$,2)+"0"+RIGHT$(A$,1)
2360 IF LEN(A$)<4 THEN 2390
2370 IF MID$(A$,3,1)=":" THEN A$=LEFT$(A$,3)+"0"+RIGHT$(A$,1)
2380 IF MID$(A$,2,1)=":" THEN A$="0"+A$
2390 IF LEN(A$)>5 THEN INVAL=1
2400 GOSUB 2480
2410 IF M1$<"0" OR M1$>"2" OR M2$<"0" OR M2$>"9" OR M3$<"-" THEN 2460
2420 IF M4$<"0" OR M4$>"5" OR M5$<"0" OR M5$>"9" THEN 2460
2430 Z$=LEFT$(A$,2): IF Z$<"00" OR Z$>"23" THEN 2460
2440 Z$=RIGHT$(A$,2): IF Z$<"00" OR Z$>"59" THEN 2460
2450 RETURN
2460 INVAL=1: RETURN
2470 ***** SUBPROGRAM - SEPARATE ANS *****
2480 M1$=MID$(A$,1,1): M2$=MID$(A$,2,1): M3$=MID$(A$,3,1)
2490 M4$=MID$(A$,4,1): M5$=MID$(A$,5,1)
2500 RETURN
2510 ***** SUBPROGRAM - DISPLAY/PRINT HEADINGS *****
2520 IF PRT THEN 2590
2530 PRINT SPC(38-LTH); TITLE$; SPC(32-LTH); "PAGE "; PAGE
2540 PRINT
2550 PRINT " DATE TIME TIME ON-OFF CUM. DAILY PURPOSE OF USAGE"
2560 PRINT "      ON  OFF  TIME TIME TIME"
2570 PRINT "(M-D-A) (H-M) (H:M) (H-R) (H-R) (H-R)"
2580 RETURN
2590 LPRINT SPC(38-LTH); TITLE$; SPC(32-LTH); "PAGE "; PAGE
2600 LPRINT
2610 LPRINT " DATE TIME TIME ON-OFF CUM. DAILY PURPOSE OF USAGE"
2620 LPRINT "      ON  OFF  TIME TIME TIME"
2630 LPRINT "(M-D-A) (H-M) (H:M) (H-R) (H-R) (H-R)"
2640 RETURN
2650 ***** SUBPROGRAM - DISPLAY/PRINT DATA *****
2660 IF PRT THEN 2730
2670 PRINT USING T1$; D.ON$(1); T.ON$(1); T.OFF$(1); EL.TIME; CUM.TIME;

```

Lines 2190-2320 checks the keyboard-entered date to eliminate obviously erroneous entries. The acceptance criteria are:

- o Initial length must be 3, 4, or 5 characters. Month and day must be separated by a dash (-). If month or day is less than 10 and without a leading zero, one is inserted.
- o In final form:
  - o First two characters (month) must be in the range of 01 thru 12.
  - o Third character must be a dash (-).
  - o Fourth and fifth characters (day) must be in the range of 01 thru 31.

Lines 2330-2460 checks the keyboard-entered time to eliminate obviously erroneous entries. The acceptance criteria are:

- o Initial length must be 3, 4, or 5 characters. Hours and minutes must be separated by a colon (:). If hours or minutes are less than 10 and without a leading zero, one is inserted.
- o In final form:
  - o First two characters (hours) must be in the range of 01 thru 23.
  - o Third character must be a colon (:).
  - o Fourth and fifth characters (minutes) must be in the range of 01 thru 59.

Lines 2470-2500 are used by the CHECK DATE and CHECK TIME subprograms to separately check each character.

Lines 2510-2580 provide headings for the extended log display.

Lines 2590-2640 provide headings for the extended log printout.

Lines 2650-2720 format data for the extended log display.

FIGURE 3. LOGDEP.BAS (Page 7 of 7)

```

2680 IF D.ON$(I+1)>D.OFF$(I) OR D.ON$(I+1)=" THEN 2710
2690 PRINT USING T2$; BLANK$; PURPOSE$(I)
2700 RETURN
2710 PRINT USING T3$; CUMD.TIME; PURPOSE$(I)
2720 CUMD.TIME=0: RETURN
2730 LPRINT USING T1$; D.ON$(I); T.ON$(I); T.OFF$(I); EL.TIME; CUM.TIME;
2740 IF D.ON$(I+1)>D.OFF$(I) OR D.ON$(I+1)=" THEN 2770
2750 LPRINT USING T2$; BLANK$; PURPOSE$(I)
2760 RETURN
2770 LPRINT USING T3$; CUMD.TIME; PURPOSE$(I)
2780 CUMD.TIME=0
2790 RETURN

```

Lines 2730-2790 format data for the extended log printout.

### SUGGESTED GOOD PROGRAMMING PRACTICES

Following is a consolidated listing of a number of good programming practices (many of which are illustrated and commented upon in the foregoing). Use of these practices will aid in making your programs easy to use, understand, and (when necessary) modify.

- o At the beginning of a program, identify it by its filename, date of last revision, and purpose.
- o If a program is to be used by others, also at the beginning, identify the author and preferably give a mailing address and/or telephone number at which the author can be reached.
- o Put all definitions, dimensions, initial conditions, etc. ahead of the program proper, where they can be easily found.
- o Keep the program logic simple and straightforward; avoid cuteness or ego trips.
- o Break the program into short logical modules (preferably no more than 50 lines or so in maximum length). Separate these modules in some easily recognized way (as with a line of asterisks) and, wherever practical, give them a short name indicating their function. In many cases these modules can be written as subprograms.
- o In initially writing a program, leave plenty of unused line numbers, to facilitate insertions. A line interval of 10 numbers is commonly used and generally works well. Upon completion of a program, renumbering makes reading easier. (There is nothing wrong with using a final line spacing of 10 numbers.)
- o Provide copious useful remarks to help in understanding

the program. (You, or perhaps another user, will be grateful in the future for the help provided.) Aligning the beginning of the remarks assists in easier reading.

- o Indent loops so that they can be easily found.
- o Use short or mnemonic names for variables; thus, they will either be easy to use (because of their shortness) or their meaning will be easily remembered.
- o When naming related variables (or files, etc.) make the first part of the name the same, so that they will appear together in an alphabetic list. For example, use time.on and time.off, rather than on.time and off.time.
- o When temporarily deleting or removing a statement, simply preface it with REM. Thus, it will become a remark and no longer an active part of the program. If the statement is to be later reinstated, simply delete the REM.
- o Use ', rather than REM, to indicate a remark where REM would be too obvious (or otherwise objectionable) in the program listing.
- o Make prompts for data input easy to understand and to answer properly. For example, for questions requiring a yes/no answer, a good practice is to give Y/N as part of the prompt but write the program so that either "Y" or "y" is acceptable as a yes answer. Generally, any other answer, including a carriage return (CR) should be accepted as no.
- o Where operations with possibly damaging consequences are contemplated (such as deleting a file), provide reasonable safeguards (such as requiring a yes answer to "are you sure?").
- o When making data changes by keyboard entry, provide for reviewing the results and making further revisions, if required,



before the changes are finally incorporated.

- o Where keyboard inputs are crucial, program checks to eliminate obviously erroneous inputs.
- o Make displays and printouts logical and easily understood.
- o Insure that displays are not automatically cleared or overwritten before the user is finished viewing them.
- o Provide message displays that tell what the computer is doing when any operation is not obvious and may take more than a few seconds.
- o Provide error trapping, so that errors can be found and rectified.
- o Anticipate possible modes of failure and provide for graceful recovery from errors.
- o Provide graceful means of exiting a program, saving those items that should be saved, junking items that are not desired, properly closing files, etc.

\* \* \*

#### VICTOR COMPUTERS FOR SALE

V9000 hard disk (10MB), 512K RAM  
VI hard disk (10MB), 576K RAM

Both machines are in full working order and have no known problems. Neither has been used for more than a year so they are not worn out. Both machines have MS-DOS 2.11 and some software with them, but not latest versions. Best offer takes either or both. Call before coming in.

(408) 425-5454

Michael R. Corder, Capitos Software,  
108 Ross Road,  
Aptos, CA 95003

V9000 with 512K RAM, dual 5.25 floppies, external hard disk (10M) with software. Software includes dBASE II, Wordstar, AutoCAD, Programmer's Tool Kit, Graphics Tool Kit, Audio Tool Kit and board, Fortran, Pascal, and C. \$500 or best offer.

Larry Bryant  
(408) 251-3333  
(San Jose, CA).

#### USED VICTOR 9000'S

William Bailey  
(805) 541-8262

(Correction to VictorTalk, Vol. 5, No. 4, Jul-Aug 1987, p. 7)

\* \* \*

#### VICTOR TECHNOLOGIES RELOCATING TO EAST / Don Clark

[The following article appeared in the San Francisco Chronicle, 8/29/87. Ed.]

Victor Technologies Inc., a venerable electronics company that nearly died during the personal computer boom, said yesterday it will relocate its corporate headquarters from Scotts Valley to Pennsylvania.

The move will affect about 40 employees. All are being offered the chance to relocate and keep their jobs.

Al Kraus, Victor's president and chief executive, said the company decided to move largely to shorten delivery times for its Japanese-built personal computers and other office products. Eighty percent of the company's customers and most of its dealers are located east of the Mississippi, he said.

The company will now import goods to the East Coast directly from Japan, Kraus said. The new location and time zone also will help communication and air connections with its Swedish-based parent, Datatronics AB.

It is but one of many dramatic changes in Victor's long history. It's name dates back to 1918 and the Chicago-based Victor Adding Machine Co., maker of one of the first adding machines.

Sixty-four years and several reorganizations later, the company merged with a Scotts Valley microcomputer maker.

When it went public in 1983, Victor employed nearly 2,000 workers, and conducted its own manufacturing and research and development.

Less than a year later, the company filed for Chapter 11 bankruptcy protection as one of the first acknowledged victims of IBM Corp.'s power in the PC business. Victor lost corporate sales because its personal computers could not run software designed for the IBM PC.

Datatronics, led by a Swedish turnaround artist named Mats Gabriellson, bought 90 percent of the company's stock in 1985. It reportedly paid \$21 million to get \$43 million in assets.

Since that time, Victor has emerged from Chapter 11, and dramatically reduced the size of its operations. It no longer builds any of its own products.

[However, it is understood that the two Regional Repair Depots will continue to offer repair service. - Ed.]

\* \* \*

# FIX FOR BUG IN PRINT ROUTINE OF KNEE TOE GRAPHICS / J.P. HasteY

[This article appeared on NewsDisk 24 (edited by Dale Cops) as KTGFIX.DOC; the related files KTF.BAT and EDITOR.EXE also appear there. (All are in archived form under ND24OMNI.ARC.)]

print routine which was limited to one screen. Knee Toe Graphics prints two screens to get an 8 1/2 x 11 page.

The problem turned out to be a LF (line feed, ASCII 0A hex) character in the printer initialization string. Once this was changed to a CR (carriage return, ASCII 0D hex), the program printed the second screen without a gap.

### Additional Fixes (Warning - these may NOT work)

The two C. Itoh printers also had LF's in their init strings (see table below). Assuming a similar problem with them, I changed those to CR's as well. I cannot confirm that the patch will work for them as I do not have access to C. Itoh printers for testing. If it doesn't, there is additional info below to allow you to change it back.

The following table lists the printers supported by Knee Toe Graphics along with their initialization strings (in hexadecimal).

Thanks go out to John & Mary Jo Knobelsdorf of Victor Talk who posted my initial inquiry on the boards, and to Clint Chaplin who responded with a nudge in the right direction. Thanks also to Charles Jennings for the use of his copy of Murray Sargent's SST+ Tracer/Debugger. SST+ is one NICE program.

### The Bug

The print routine for the Epson MX-80/100 would leave an 8-dot high gap halfway down the printed page. This was very bothersome when the gap appeared in the middle of a graphic.

### The Fix

Clint Chaplin's clue was that Tony Casano used the Grafix

Printer	Original Init String	Patched Init String
Epson MX-80/100 (GrafTrax ROMs)	0A 1B 41 08	0D 1B 41 08
Mannesman Tally	00 00	N/A
C. Itoh	1B 45 1B 54 31 36 0D 0A	1B 45 1B 54 31 36 0D 0D
Okidata	1D	N/A
Epson FX (with Victor ROMs)	1B 41 08	N/A
C. Itoh S. (with Victor ROMs)	1B 2A 37 0D 0A	1B 2A 37 0D 0D

The printer configuration tables begin at stack segment offset 0BD1h in EDITOR.EXE, when loaded by DEBUG or SST+. Only SST+, however, will allow you save an .EXE

file. The LF's in question are indicated with a caret (^) in the file segment below. The first is the Epson MX-80/100, the second the C. Itoh, and the third the C. Itoh S.

```

ss:0BA0 0A 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
ss:0BB0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
ss:0BC0 00 00 00 00 32 00 00 00-00 00 00 00 00 00 00 ....2.....
ss:0BD0 00 01 01 08 00 00 00 04-0A 1B 41 08 02 1B 32 00 .....A...2.
          ^
ss:0BE0 00 02 0D 0A 02 1B 4C 00-00 FF 01 01 F8 FF C8 00 .....L.....x...
ss:0BF0 00 00 00 00 00 00 02 0D-0A 02 1B 4C 01 20 FF 01 .....L.....
ss:0C00 02 F8 FF 00 00 08 1B 45-1B 54 31 36 0D 0A 04 1B .x.....E.T16....
          ^
ss:0C10 4E 1B 41 00 00 02 0D 0A-03 1B 53 30 00 00 FF 01 N.A.....S0....
ss:0C20 03 F9 FF 01 00 01 1D 02-03 02 00 00 02 03 0E 01 .y.....
ss:0C30 03 00 00 FF 01 01 08 00-00 00 03 1B 41 08 02 1B .....A...
ss:0C40 32 00 00 02 0D 0A 03 1B-2A 07 00 00 FF 01 02 F8 2.....*.....x
ss:0C50 FF 00 00 05 1B 2A 37 0D-0A 04 1B 4E 1B 41 00 00 .....*7.....N.A..
          ^
    
```

```

ss:0C60 02 0D 0A 03 1B 53 30 00-00 FF 00 00 00 00 00 00 .....S0.....
ss:0C70 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
ss:0C80 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....

```

The patched version of the main Knee Toe Graphics file EDITOR.EXE is included on this disk. [NewsDisk 24, archived within ND24OMNI.ARC] To distinguish the difference between the buggy version and the patched version, the word "[Patched]" appears after Tony Casano's name on the title screen. I had to shorten "Copyright" to

"(C)" and "Toe Knee" to "Tony" to get it all to fit in the same space. I hope Tony doesn't mind.

Copy the patched version of EDITOR.EXE onto your working copy of Knee Toe Graphics and run as usual.

\* \* \*

MORE ON KNEE TOE GRAPHICS [Downloaded from BAVUG.]

#36 14 31 Aug 87 15:21:53 (RECV'D)  
 From: Bert Happel  
 To: Franz Hirner  
 Subj: Knee or Toe deep

Well, I'm ashamed to admit I solved the problem shortly after posting the message (then a busy weekend and busy PC-P kept me from replying back sooner).

The Situation:

KTG had run fine on the V9 with DOS 3.1. "Suddenly" it loaded up with a garbage banner screen. KTG would run fine from the floppy disk under 3.1. I tried re-copying the editor and KTG.\* files to the HD, but no change in result. Since the program ran fine from the floppy, there must be some difference between the files on the floppy vs. the hard disk. Using Multo, I compared the two directories, and found (remembered) that I had deleted a couple of .CHR files from the HD directory since when they were loaded from the TEXT option in KTG, the keyboard produced not letters, but small increments of cursor movement.

The Result:

I re-copied SHADES.CHR onto the HD from the floppy. KTG loaded fine and ran fine. The bottom line is that KTG has a support file called SHADES.CHR that \*must\* be available to KTG. SHADES cannot be used from within the program, but must exist all the same.

This support file situation should be noted in the DISKID file on the V\*G pd disk #71.

(BTW - I got the page printed that I was working on. Very nice result. I did update the EDITOR.EXE file with the revised version on NewsDisk 24. It eliminates a blank line transmitted in the middle of a full page of graphics.)

New KTG question: The first graphic that I print on my Star NX10 (telling KTG I have an Epson MX) comes out double spaced. All prints after that come out proper. Anyone know why? I'm printing a small junk file upon starting KTG sessions to overcome this.

#53 15 02 Sep 87 13:16:37 (RECV'D)  
 From: Franz Hirner  
 To: Bert Happel  
 Subj: Graphic blank line

Only suggestion I have is check your dip switches carefully. The fact that you do not get the blank line after exiting KneeToe indicats that it is a software or setting problem.

#59 11 02 Sep 87 21:17:29 (RECV'D)  
 From: Bert Happel  
 To: Franz Hirner  
 Subj: To space or not to space

Knee Toe, as a program, runs fine. The problem arises when printing a screen. Once I have the .SCR files loaded (or new graphics created), and issue a "Print" command, the printer inserts a blank line between the lines of graphics. After this double-spaced printout, I can issue another "Print" command and the graphic comes out fine - no extra spaces.

The why has me baffled. As I said, to overcome this glitch I waste a sheet of paper printing out any screen to cure the double spacing, then go on to the KTG editing I want to do.

#61 11 02 Sep 87 21:49:21 (RECV'D)  
 From: Franz Hirner  
 To: Bert Happel  
 Subj: KTG

KTG must be resetting your printer after printing a screen, rather than before. You should be able to come up with the same initial settings by playing with the dip switches.

I don't think it's related, but some MX's will do that ALL the time depending on the ROMS. In your case I think it's your switch settings though. If you can't get your switches right you could always do a small COM file to initialize the printer. See the examples in Waite's "Assembly Language Primer" using DEBUG. It's not too hard to do.

\* \* \*

# FROM THE BOARDS

[FROM THE BOARDS is assembled from messages appearing on various (generally Victor-specific) bulletin boards, particularly The PUB, by John and Mary Jo Knobelsdorf. Only minor editing is done to correct obvious typographical errors and to eliminate some superfluous material. An effort has been made toward a degree of standardized formatting and the elimination of excess space to the extent practical without degrading legibility. Garbles that cannot be readily rectified are allowed to remain. Believing that considerable worthwhile material is contained in these excerpts, our intention is to include this feature in each issue of VictorTalk, as space will allow. - Ed.]

NOTE: The first discussion (2+ pages) relates to Word Perfect.

#1 14 19 Jul 87 08:59:16 (RECV'D)

From: Franz Hirner  
To: Lloyd Leblanc  
Subj: Word Perfect

You caused me to dig out a few Word Perfect reviews last night. Needless to say, the reviews weren't based on the Victor, which prompts the following questions:

PC Magazine indicates 4.2 is the latest version of WP. There are few differences between Ver. 4.1 and 4.2, but they didn't say what those were. Do you know the differences? Which is the latest for the Victor? I do a fair amount of programming which requires ASCII or non-document files. Does WP have it's own particular file format? Can an ASCII file be produced without going through a conversion process?

In the case of commands, WP seems to use unshifted, shifted, Alt and Control function keys to access all of the control, format and other functions. Are all functions available in the Victor format? How do they get around the Alt/Control key?

Is there such a thing as a demo disk in Victor format? How about IBM format?

I have a dozen other questions in mind, but this is a start. The ASCII is critical for "C", dBASE and other programming.

That's all for now. I gotta go pull a keyboard apart and clean it up so that the next Plus PC will switch modes. Incidentally, TV tuner cleaner, available from Radio Shack or and electronic supply house, works great for cleaning keyboard contacts.

#7 13 19 Jul 87 16:16:33 (RECV'D)

From: Lloyd Leblanc  
To: Franz Hirner  
Subj: THANKS

On Wordperfect: the "latest" version 4.2 is available for the Victor and if you wait more than a couple of weeks, you'll benefit from the availability of a new, faster and much improved Victor version 4.2 (I was a beta tester along with a number of other people -- WP Corp has FINALLY taken advantage of the actual Victor screen handling routines (improving speed) and given us all the IBM version features (some of which had not been implemented as of the early 4.2 version). All of the Wordperfect features are fully "ported" to the V9000 and it runs well indeed. The function keys are all used, as are ALL the special V9000 keys, and in pretty decent ways at that. Plus, you can fool around with Keygen and improve on your own.

Getting an ASCII file out of work is as simple as using a file save keycode (for a "text" file), no sweat, same on importing an ASCII file for use in Wordperfect. You can also "print to disk" in ASCII, getting centering, tabs, etc. as you want them to read in DOS.

There are many, many features not all of which are important to all people. But it is clearly one of the superior programs -- and of those, the only one now supported for the V9000 -- so it's worth a look. Sorry there are no demos. I'd be glad to run you through it if you want to see in action, or maybe you can get input at the Victor Group meetings. HINT: There is a "student" version (but with all features I think) available through EGGHEAD and other outlets. Your son or daughter could easily get a copy for you. I'm NOT talking about the abbreviated "Personal Wordperfect" which was out or still is -- cheaper but truncated. Or Wordperfect "Executive" -- different yet -- but the "Student" version of 4.2. I think it's less than \$150 discounted.

#10 14 19 Jul 87 16:56:01 (RECV'D)

From: Franz Hirner  
To: Lloyd Leblanc  
Subj: Wordperfect

I made the trek to Computer Literacy this afternoon and picked up "The ABC's of Word Perfect". This should give me a pretty good idea of the capabilities. Sybex has a hardcover "Desktop Reference for Word Perfect" which looked really great if you are using the program. I don't know how it compares to the official manuals, but you might take a look.

#17 16 21 Jul 87 05:51:04 (RECV'D)  
From: Franz Hirner  
To: Lloyd Leblanc  
Subj: Word Perfect

When will the revised WP 4.2 be out, where do you get it, and what is the price.

In the Victor version, does the help menu represent the Victor key combinations (unshift, alt, and shift) or does it represent the IBM version with control-key definitions also?

I found a user in our office and had a peek at WP yesterday. It would be a pain for programming since you have to write an ASCII file via a print option and the file printed to disk always has the same name. Other than that, it's looks like it deserves it's fine reputation.

#19 17 21 Jul 87 07:45:32 (RECV'D)  
From: Lloyd Leblanc  
To: Franz Hirner  
Subj: WP and PC+

No, you do not have to use the print routine in WP to create an ASCII (in WP's lingo, a "text") file. You simply use one of the menu choices which is file as a "text file" (using any name you wish).

You CAN "print to disk" as I indicated, and then you get all the formatting you had used (centering, non-DOS tabs, etc.) - - but you CAN then easily rename that file.....and for programming I would not think you would print to disk necessarily?

One of the very nice things about WP is that they are good at integrating their products. The new WP "Executive" program (for laptops -- one disk required only) loads INTO MEMORY a smaller version of WP that is totally compatible with the big brother. Etc. etc. Not too many other companies doing this right.

The "latest" fixed Victor version should be available in another couple of weeks, if they get a couple of bugs out without great difficulty. I can keep you posted. Of course upgrades from WP Corp are usually \$35, and if you buy just before an upgrade, usually free.

#28 14 21 Jul 87 13:11:35 (RECV'D)  
From: Franz Hirner  
To: Lloyd Leblanc  
Subj: WP

OK. We found the "TEXT" command and it works as you describe.

It's a little difficult to work on someone else's machine, especially when you just popped in the door and asked to sit

down at the computer.

Source code quite often is wider than 65 characters. How does the screen scroll when you have a document that is 80 or 90 characters wide? Does the entire screen scroll or just the line in question?

I did look at the WPHelp.FIL with debug. It looks like there are embedded codes in the area of the file where help for the function keys is located. Is the help screen for the Victor accurate or must you remember what key does what? You did not indicate if they use a substitute for the control key so I presume that the function keys don't work quite the same way as on the IBM. Am I right?

#30 13 21 Jul 87 16:36:24 (RECV'D)  
From: Lloyd Leblanc  
To: Franz Hirner  
Subj: WP USE

In the Victor version of WP 4.2 -- there is not a revised function key "map". Thus you must use the printer templates (there is one for the function keys, comes with program) and the rather good small booklet that has ALL the key functions in it, color coded (blue for shift, green for alt, etc.). It IS a bit trying jumping from the IBM to the Victor versions, as the FUNCTION KEY ASSIGNMENTS ARE NOT THE SAME.....

And -- yes, you are right -- without a CTRL key on the Victor some of the assignments are different (most of the IBM CTRL functions are shifted to the V9000 "special" keys).

I use both on a weekly basis and find it a challenge, but not too mind-bending. On wide screens: you simply set margins to 80, or whatever, and you get what you see, whole screen continues to scroll -- you can "right margin" to see what's outside the initial screen. One keystroke. I use a lot of 110 column tables, and jump back and forth. It's almost like a change screens command in Lotus. Thanks for ideas on the PlusPC. I'll call someone right away, maybe there are others left who know.....

#31 16 21 Jul 87 17:29:22 (RECV'D)  
From: Franz Hirner  
To: Lloyd Leblanc  
Subj: WP

Have you tried generating your own keyboard so that the function keys are more nearly identical on both machines? Wonder what would happen if you used Guy Gordon's IBM.KB. This implements a Control-Key on the "Lock" key.

#32 12 21 Jul 87 19:33:31 (RECV'D)  
From: Franz Hirner

To: Lloyd Leblanc  
Subj: Wordperfect

Well, I'm back again. I finished "The ABC's of Word Perfect". I need to go down and get the Desktop Reference Manual I mentioned to you. One nice thing about Computer Literacy - you can return any book within seven days for full credit towards another purchase.

I think I have one last question regarding WP. When you are doing a mail merge using data created from dBASE III, is there a command to skip a line entirely if the data element is void? Also, can your list to merge be just plain delimited characters without the ^R and ^E after the elements of the record and the record itself?

dBASE produces a delimited merge file like:

```
"firstname","lastname","addr1","addr2","city","state","zip"
```

If there were no address # 2 that field would be merely "". This is a pure ASCII file and there are no ^R's or ^E's as you can see.

Wordstar MailMerge allows an option of "/o" when defining a field. This means to skip the field entirely, without a line feed, if it is null or "" in value.

I must admit, "The ABC's of Word Perfect" makes the program look very impressive. I do a lot of MailMerge, though that requires the above mentioned ability to not print anything if a data field is void. Also, dBASE III does not produce the ^E and ^R's that "ABC" indicates are required.

If I understand correctly, the fields being merged could be several lines long, or even paragraphs, by using the special delimiters ^R and ^E which might be an advantage in some situations, but if you can't skip a null field I'd still have to fall back to wordstar for some or all merge applications.

#37 13 22 Jul 87 08:43:34 (RECV'D)  
From: Lloyd Leblanc  
To: Franz Hirner  
Subj: WP

I'm afraid I don't really know much about mail merge usage -- but I have looked at various "how to" books and WP seems very powerful. The major books cover that pretty well. The Sort functions are useful too -- I have used those.

On function keys: As far as I know, WP Corp uses the function key codes in the system file code and no one has successfully modified those to match the IBM. I myself already use a KB with the LOCK key (unshifted) as a CTRL key. But since the Victor 9000 4.2 version only uses UNSHIFT, SHIFT and ALT function key routines, you don't need it. Truthfully, I prefer

the upper level function key placement and the way the Victor routines are implemented -- to the columnar (olb) IBM layout. The Toshiba also uses the upper level and I get a little confused, but not too much.

Important -- and you may not know this -- the IBM VERSION RUNS WELL ON THE V9000 UNDER THE EMULATOR.....except for printing out the serial port and I'm not sure about the parallel. So you COULD get by with one copy, and still run it on the Victor. I've run it this way myself and all the features seem to work fine. Cursor control is a little less happy, "flickering" on some screen reads.

NOTE: The next discussions (2 pages) concern the latest and "best" IBM emulator for the V9000.

#95 15 26 Jul 87 18:23:02 (RECV'D)  
From: Franz Hirner  
To: John Navas  
Subj: XTBIOS

I've placed a file named XTBIOS.ARC in the pick-up area. It seems to be a fairly decent IBM emulator for the V-9000. It will run CODEVIEW's demo in the TEXT MODE although a bit slow. Only problem, it uses the German Keyboard rather than the American Keyboard. I wonder if you'd run this through your disassembler and see if we can find the keyboard translation table. It might be possible to install a U.S. Keyboard which should make it useable, at least for Codeview.

I believe I have an early copy of the source for a program that this has been adapted from. I'll try to dig that out and see what it shows also. I tried to compile it once but there were errors. I gave up on it.

#106 14 27 Jul 87 17:08:36 (RECV'D)  
From: Franz Hirner  
To: John Navas  
Subj: XTBIOSA.ARC

Thanks for your efforts. I uploaded the quick mods I made today so that other's might have the benefit of XTBIOS.

Into every life some sun must shine. The sun was bright last night when I was able to run CODEVIEW in all it's glory on the V9. Dual screen in the mono mode is the limit at the moment since graphics aren't included in XTBIOS yet. Still GREAT!!!

Looking at the ASM file the keyboard sticks out loud and clear. Did you do the XTBIOS or the HXTBIOS? HXTBIOS requires 896K but also supports Codeview's dual screen.

I must confess that after seeing Codeview in operation and not being able to run it on the V9 except in "DEBUG" mode caused me to have second thoughts. In spite of all the great software for the V9 that is not available in the IBM world, I did "lust in my heart" to be able to run Codeview. You know what I'll be doing tonight!

#109 16 27 Jul 87 19:46:07 (RECV'D)  
From: Franz Hirner  
To: John Navas  
Subj: XTBIOS & HXTBIOS

A few more keyboard changes tonight make things even better. "Codeview"!!! Can you imagine that!!!

Now all I have to do if figure out how to rearrange memory between my various machines.

#114 14 28 Jul 87 21:03:49  
From: Franz Hirner  
To: All  
Subj: HXTBIOS

I've continued to work with the code of XTBIOS and HXTBIOS. As a result Codeview is running like a champ. I still need to make a few changes to fine tune the keyboard though.

I loaded a copy of Wordstar 2000 onto the trusty Victor tonight and went through the install process under HXTBIOS. The installation was a real mess because the screen would not clear after the answer was given to the various questions and before the next question was asked. Fortunately, I have installed Wordstar enough times that I know most of the answers by heart and managed to bungle through the process. To make a long story short, Wordstar 2000 now runs (90%) on the Victor with HXTBIOS. It hasn't been given a full test by any means, but text is entered, on-screen bold and underline work, and the file can be printed to my Juki (Diablo 630 clone) and it comes out looking like the screen. There is a problem with BLOCK MOVES and there may well be other problems, but the first attempt is encouraging. There are still some things (some???, try a lot) I don't know about how the HXTBIOS operates which makes it a little difficult to figure out where to patch things.

If anyone else wants to take a try at HXTBIOS, it's in the upload directory.

#115 14 28 Jul 87 21:12:42 (RECV'D)  
From: Franz Hirner  
To: John Navas  
Subj: XTBIOS

I really appreciate your help the other day on this. Codeview

does require the HXTBIOS version which is just a little slow, but not really bad compared to the 5 MHz. VI. Great.

The previous message (in case you didn't see it) relates to a quick attempt at installing Wordstar 2000 onto the Victor. Other than I'm not sure that I still have the keyboard right (I know I need to make a few more change) it looks like Wordstar will run with some tuning. It's really great looking at the Victor resolution even compared to the Hercules Card in an IBM.

As you could tell from my messages re Codeview, my pulse is really beating. Now that a quick pass at Wordstar shows a lot of promise I have to force myself to slow down, think things out, and proceed a step at a time. I don't really like the Windows environment or interface, but there's no question that a lot of software would have been made available for the Victor if enough programming support had been available from the group. It looks, at this point, like XTBIOS will expand the Victor's software library considerably. The first experiments show a lot of promise! Now, if only there were multi-tasking?

#121 16 29 Jul 87 06:04:14  
From: Franz Hirner  
To: All  
Subj: HXTBIOS

A few more patches were made to XTBIOS last night. It looks like Wordstar 2000 is running fine. At least, the earlier problems noted with "blocks" have been corrected.

We need to establish a list of software that runs under the HXTBIO and XTBIOS interface.

Now if the IBM world only had a V9BIOS so they could run all the great programs for the Victor that aren't available in the IBM world. [grin]

#122 15 29 Jul 87 06:10:08 (RECV'D)  
From: Franz Hirner  
To: Lloyd Leblanc  
Subj: HXTBIOS

If you have 896K in one of your machines you might check Word Perfect and see if your IBM version will run on your V9. (Did I hear your PlusPC is obsolete?)

#128 16 29 Jul 87 17:46:09 (RECV'D)  
From: John Navas  
To: Franz Hirner  
Subj: XTBIOS

It looks, at this point, like XTBIOS will expand the Victor's software library considerably.

Glad to hear it! Perhaps it would be worth the effort to try

and contact the author in Germany to obtain the latest version and/or source code.

#129 18 29 Jul 87 18:17:47 (RECV'D)

From: Franz Hirner  
To: John Navas  
Subj: XTBIOS

For the first time, I'm one step ahead of you. I wrote him today offering any assistance in converting XTBIOS to U. S. keyboard, etc.

Once programs are set up for the MONO TEXT CARD every things to run well. I had a tough time setting Wordstar up though. HXTBIOS checks for an EGA card and when it doesn't find it reports a "possible problem - continue at your own risk". This is OK as you can step right through the error code, but apparently when WS doesn't get the answer it expects it forgets to clear the screen before presenting the next page of installation instructions. After a couple of screens laid one atop the other it becomes hard to read the instructions. Once set up though, WS is expecting a MONO TEXT card and has no problems at all.

#141 14 30 Jul 87 10:20:42 (RECV'D)

From: Lloyd Leblanc  
To: Franz Hirner  
Subj: XTBIOS

Wordperfect 4.2 (IBM version) seems to run fine. They did so under the old emulator, too. How does XTBIOS affect serial port transmission? Would that not restrict its use with ANY program requiring serial port mods? Including word processors? (of course you could use the parallel port or a converter).

The concluding messages (from BAVUG) concern WSSI, the disk indexing program.

#89 11 06 Sep 87 12:28:12

From: John Knobelsdorf  
To: All  
Subj: WSSI320

I have started working with WSSI320 to catalog and index my collection of programs and files on a multitude of disks. I still have a lot to learn about WSSI320 before I can evaluate its usefulness.

Some observations are:

1. WSSI320.EXE seems to run without any difficulty on my V9000 when the XTBIOS.COM emulator program is installed.
2. When running WSSI320.EXE with XTBIOS.COM, there is no need to install any other emulator, screen driver or

keyboard program.

3. WSSI320.EXE modifies the keyboard, so I must modcon back to the Victor keyboard when I exit the program.
4. It is advisable to disable any kind of "resident" SCREENSAVE program before using the WSSI320 programs.
5. The related program IDADD100.EXE (which is part of the WSSI package) seems to run O.K. with XTBIOS.COM, and it does not modify the keyboard.

I run the WSSI programs from a sub-directory on my internal harddisk, and use the following batch file to run WSSI320.EXE:

```
scrnsave -d
xtbios
wssi320
modcon vic.kb
```

I have not yet tried to use any of the programs in the WSSI package other than WSSI320.EXE and IDADD100.EXE, but will report on the others as I puzzle thru them.

If anyone else is working with the WSSI package (particularly ver. 3.2), please post messages to the PUB reporting on your own experiences.

#91 9 06 Sep 87 12:47:38 (RECV'D)

From: Sysop  
To: John Knobelsdorf  
Subj: WSSI

I'm not sure which version of WSSI I'm running without looking. It's the same as posted on BAVUG. If you have a later copy I'd appreciate a copy.

XTBIOS handles most ANSI screen commands and eliminates the need for ANSI to be loaded. All in all, not a bad package.

I think you will find that WSSI will handle most of your library chores. You might want to take a look at the dBASE file library system that is available for downloading from the dBASE Directory.

#103 5 07 Sep 87 06:16:40

From: Franz Hirner  
To: Judith Moore  
Subj: WSSI

When you run WSSI on a Victor you must use one of the emulators to simulate an IBM. IBM uses BIOS calls not available plus several machine dependent DOS calls. The emulator's take control of the keyboard as well as use timer interrupts which blocks SCRNSAVE from seeing a keystroke. When SCRNSAVE turns down the screen it stays dark under the emulator.



#86 6 09 Sep 87 15:53:51 (RECV'D)  
From: Judith Moore  
To: Franz Hirner  
Subj: WSSI/Victor

Ahhh. Now I see. That WSSI is great! Just shove in a disc and type the space bar. I entered about 120 discs and the same afternoon was saved about 2 hrs. of hunting time to find specific files. Somehow the first time I'd tried it, I'd selected the wrong options and thought you had to comment each file in the database. Don't think it should be too hard to keep up, as one can get a Volume listing to see where you left off numbering discs, and then just add the new one in an eyeblink. (Maybe John will read this).

#89 5 09 Sep 87 16:18:01  
From: Franz Hirner  
To: Judith Moore  
Subj: WSSI

WSSI does make a fairly easy chore of keeping track of what's where. Comments are optional but sure help when you are looking for something you have not seen for six months.

#99 5 09 Sep 87 19:03:50 (RECV'D)  
From: John Knobelsdorf  
To: Franz Hirner  
Subj: WSSI

The latest version of WSSI that I have seen is ver. 3.2, otherwise known as WSSI320.EXE. The current package of files is rather large, even in archived form, and is being handled in two parts, WSSI320A.ARC and WSSI320B.ARC. It takes about twenty-five minutes to transmit each part, so I'll not send it tonight because of the other messages I want to leave posted, and the downloading of messages I wish to capture. I promise to upload both parts of WSSI320 on 09/10/87.

#101 4 09 Sep 87 19:08:10  
From: John Knobelsdorf  
To: Judith Moore  
Subj: WSSI PUB etc.

I see that Franz Hirner has already responded to the messages that you posted for me. That's good, because he knows that I log on to BAVUG rather sporadically. Sometimes I appear here frequently, and other times only on rare occasions.

Franz told you about the PUB in Boston. I can't add much to what he said, except that it is very Victor specific. References to itty bitty machines and clones are not tolerated.

As far as WSSI is concerned, Franz is much more enthusiastic than am I. The set of programs seems to offer much in the way of indexing files and disks, but I can't seem to get the set

to do what I want in the way of cataloging and/or identifying all the various files and programs in the Victor\*Group public domain. As I said in my earlier message, I still have a lot to learn about WSSI, and remain hopeful that it will indeed really be of assistance. What ever the outcome, I'll post a report to BAVUG and the PUB.

Please let me and other folks know of your own experiences with WSSI if you do start working with it yourself.

Welcome to BAVUG. Have fun.

#102 3 09 Sep 87 19:15:31  
From: Franz Hirner  
To: John Knobelsdorf  
Subj: WSSI

I'm sure that the more you work with it the more you will like it. If you have not already, print out the DOC's for reference. You'll need them at least until you have used the program for a while.

There is a dBASE III program here if you want to take a look at it. dBASE may be more appropriate for your purposes in that you can write your own routines to do whatever you wish. All in all, I think WSI should do most everything you could ask for once you get the hang of it.

#122 2 10 Sep 87 15:01:46 (RECV'D)  
From: Judith Moore  
To: Franz Hirner  
Subj: WSSI

Yes, nice to have comments. But even nicer to be able to organize a whole flock of discs and at least find things by file names by doing the absolute minimum of input. Even on a bad day I can handle 'push in disc; hit space bar; take out disc; write "3F" on label'.

#123 2 10 Sep 87 15:06:52  
From: Judith Moore  
To: John Knobelsdorf  
Subj: WSSI

See previous message re:my satisfaction with WSSI. Even without comments, I can FIND FILES without shoving each one in the drive and looking at the directory (yes, I know I should label them properly as I go along, but I don't.) When I have time (i.e, when the chill sets in in Hades), I'll see about comments and categories and such. Tonight I was just happy to be able to pick out the disc that had READBIN on it.

Didn't know Fido let you address mail to manufactured names. Thanks for the message. Will comment later on use of WSSI after I've become more familiar.

\* \* \*

# THE LIBRARY REPORT / John Knobelsdorf

Four new public domain disks, PD #87 through #90, have been released. Copies are now on file in the Chapter set which is maintained by John Kimber, and will be available for copying at the monthly meetings. They also may be ordered thru the mail in the usual manner.

NOTE: Some files on these disks are in compressed form.

Use NUSQ (on PD #73) to "Unsqueeze"  
Use LUE " " to "Unlibrary"  
Use PKXARC (on PD #87) to "Unarchive"

All of the files on PD #90 require the use of a Victor VI, Plus-PC, a PC-Compatible or one of the various emulators. Try the files and programs, and let us know what works and what doesn't work. We gather files like this for your

potential benefit. Help us do a better job by advising of your experiences.

\*\*\*\*\*

## SPECIAL NOTE:

All of the files and programs on these disks were downloaded from the PUB, a private user's board in Boston, owned and controlled by Michael Wishnietsky and operated by Marty Moleski, the Sysop. Our special thanks to Michael and Marty and all of the members and users of the PUB who contribute so generously of their time, talents and telephone expense.

\*\*\*\*\*

The contents of the DISKID files for each of these disks follow.

The contents of disk PD #87 are:

DISKID87 TXT 6144 Information about the files on this disk PD #87.

### \*MISCELLANEOUS UTILITIES\*

AFILT172.EXE	13440	Bug-Fixed version of AFILTER. FF exp. works now
AFILTER.DOC	3328	Sam Martin's notes; how to use it in many ways
CONFIG.ARC	17408	All about Config.sys and related matters; helpful
ENVFREE.ARC	3968	Shows free/used bytes of current environment.
ENVXPD.ARC	3200	A bit more light on the Mysterie of "environment"
FFSOURCE.ARC	24576	Good file finder w/MS-C 4.0 source; FDA is better
FGREP151.ARC	8192	Chris Dunford's fast GREP utility; latest
GT-TXT.ARC	13568	Helpful insights into BBS communication problems.
HEBREW.ARC	3584	Hebrew .CHR & .KB from the only known Jewish bear
HIDE.COM	768	Quickly hides file/directory, wild Ok (*.*=all)
KBFIX23.ARC	14336	Keyboard fix for VPC/clones-foiled IBM again
MSDOS31.ARC	9728	More than Bruce ever wanted to know about 3.1
NOTE11.ARC	17792	Save notes directly from command line; no fuss.
P6-CODES.TXT	8960	Helpful hints for the NEC P6/P7 printer.
PASCSEM.ARC	27904	Good introduction to Turbo Pascal; unprinterized.
PCP-TIPS.ARC	10240	Technical info of interest and usefulness; read.
REPEATS.ARC	8960	Lists filename repeats in dir's; see, then tidy.
REWMP3.ARC	51456	Lloyd's revised install for MP Ver 3.0
SEEK-TXT.ARC	2176	Like FGREP, but faster; find string any dir/file.
SEEKTX2.ARC	3968	Seek-txt.arc + perceptive review (read that first
SPLIT.ARC	12800	Split long Ascii file into several; V9, see doc
STALLMAN.ARC	14848	Interview with Richard Stallman, by Judy Getts
UNHIDE.COM	768	Quickly unhides file/directory, wild Ok (*.*=any)
V20DC-FM.ARC	23552	DISKCOPY and FORMAT to use with the NEC V20 chip
VF150.ARC	12160	Screen oriented DOS shell/test HXTBIOS operation
VIEWCHAR.ARC	21120	Type decimal, see char.; hit key, see coding.
XANADU.ARC	17152	HOTDIR and SFIND Utility
XTBIOS.ARC	25344	Kurschat's emulator w/ Hirner's .KB changes
ZDEL3001.ARC	7424	Hanlin's DELete w/verification; useful for bears

\*COMMUNICATIONS PROGRAMS\*

KERM229 .ARC 34048 Victor Kermit; Ver 2.29; w/concise documentation  
Note: This is the same as VKERM229.ARC on PD #83.

\*PRIMARY BBS UTILITIES\*

Note: The following files are the latest versions of files  
that are on PD #83.

PKARC35 .ARC 28800 Phil Katz's ARChive utility; fast; full-featured  
PKXARC .COM 12288 FAST! unARChiver; ver 3.5; you need this first!  
PKXDOC .ARC 8832 doc for above; Type PKXARC PKXDOC to unARC it

\*ANOTHER FUN FILE\*

HAL9000 .TXT 3712 It should sound familiar...

\*THE FOLLOWING REQUIRE THE USE OF "EMIBM"\*

ALSEARCH.ARC 3072 Searches all drives/directories for file; Emibm!  
EFFIC .ARC 18304 Shows efficiency of disk usage by files; Emibm!  
EXTRACT .ARC 13696 Extract all or specified part of Asc file; Emibm.  
WHEREIS4.ARC 2048 File finder, see doc; requires Emibm!

\* THESE ARE THE LATEST VERSIONS OF FILES ON PREVIOUS PD DISKS \*

BAC .ARC 1920 Dickinson Associates' HD backup utility; decent  
BIGD .ARC 9600 Best "quick and dirty" HD/Floppy file DIR utility  
DISK .ARC 8192 More info/features/paper than BIGD; w/pipes  
DPATH16 .ARC 7936 DataPath Ver 1.6; PATH to data files; ASM in Pad  
DPATH30 .ARC 7296 Another approach; older; powerful; no source code  
ND .ARC 1408 Exclude files w/wildcards from DOS commands; neat  
PATCH .ARC 14592 Guy Gordon's automatic file patch utility  
PKFIND11.ARC 10368 Katz's file finder; many switches; hard-to-beat

574976 Bytes in 47 Files 8192 Bytes free

The contents of disk PD #88 are:

DISKID88.TXT 4224 Information about the files on this disk PD #88.

\* THESE ARE THE LATEST VERSIONS OF SOME USEFUL FILES \*

EMIBM .ARC 25567 ib-emulator; the latest version.  
GRAFIX .ARC 58919 Ver 2.5; fixes bugs; works w/DOS 1.25/2.11/3.1  
MASM4FIX.ARC 5450 Various MASM 4.0 bug reports/fixes

\* SOME STUFF TO USE WITH PMATE \*

PMATE .ARC 6528 Victor 9000 configuration file for PMATE 4.0

\* USE THE FOLLOWING WITH "C" and "MS-C" \*

HIRES .ARC	19666	Bill Burrough's C graphics demo; w/C & ASM source
MSC4ADD .ARC	14080	Excellent function listing; add to MS-C refs
OV143B .ARC	165888	Complete MS-C source for OverView, Ver 1.43
V9CSCRN .ARC	33024	Screen stuff for MS C 4 Compact Model
BOSS687 .ARC	197632	Window BOSS for C compilers; works with MS, CI, Lattice, Turbo C; source code when registered!

\* SOME FILES TO USE WITH "WORDPERFECT" \*

KEYS .BAT	128	A simple WordPerfect keybd installation utility
WPKEYS2 .DOC	2304	DOC file for WPKEYS2.KB
WPKEYS2 .KB	1408	A revised WP KB file to fix EXIT TO DOS problems

\* A FILE TO USE WITH "TURBO-PASCAL" \*

TSRSRC25.ARC 62976 Kim Kokkonen's TSR managers w/Turbo-P source

598016 Bytes in 14 Files. 8192 Bytes free.

The contents of disk PD #89 are:

DISKID89.TXT 4224 Information about the files on this disk PD #89.

\* PROGRAMS FOR PROGRAMMERS \*

ADVBAS34.ARC	119808	Hanlin's superb QB functions; last shareware ver
UNDOC .QB	2560	Undocumented Q-BASIC switches
ADVCI1 .ARC	14080	Tom Hanlin's ADVanced MS-C functions

\* PROGRAMMING UTILITIES \*

ASMGEN21.ARC	23013	Best of the disassemblers; more to come!
DOSKNDW2.ARC	29312	What DOS knows about your computer; interesting
GM .ARC	3021	Dayton's Graphics Memory SET/FREE; w/ASM source; more reliable than Victor's GET/KILLSCRN
MEMMAP5B.TXT	7040	Victor 9000 memory map; do you have any input?

\* ASM SOURCE CODE \*

BAC-ASM .ARC	6993	Source code for Dickinson Associates' BAC.COM
DP16-ASM.ARC	12986	Source code for Bruce Dubbs' DataPath, Ver 1.6
LIST-ASM.ARC	13918	Source code for Buerg/Holubow/Chase LIST ver 5.50

\* PROGRAMS THAT RUN ON PC-COMPATIBLES OR REQUIRE USE OF AN EMULATOR \*

OV150 .ARC	84992	Latest ver of Mathews excellent HD/File Manager
ZANSI-12.ARC	29696	Hanlin's is shorter/faster than NANSI; w/ASM code.
TDRAW221.ARC	108416	Best of the shareware ANSI screen editors; PC-Comp.
AUTOM401.ARC	142336	AutoMenu 4.01; best of the menu-makers; EMIBM?

603264 Bytes in 15 Files. 2048 Bytes Free

The contents of disk PD #90 are:

(Note: All of the files on PD #90 require the use of a Victor VI, Plus-PC, a PC-Compatible or one of the various emulators.)

DISKID90.TXT 4096 Information about the files on this disk PD #90.

\* A SHAREWARE SPREADSHEET \*

ASEASY28.ARC 101376 ? Best shareware spreadsheet; will EMIBM work?

\* A SHAREWARE DATABASE MANAGER \*

FE383-1 .ARC 115712 FILE EXPRESS: reportedly the best of the shareware database mgrs; even better than PC-FILE.  
FE383-2 .ARC 130048  
FE383-3 .ARC 90112 - anybody wanna try +PC and/or the emulator? -  
FE383-UP.ARC 38912 Update info on FE; new report utility

\* A SHAREWARE EDITOR UTILITY FOR PROGRAMMERS \*

QEDIT136.ARC 99328 Best of the shareware programmer's editors

\*LATEST IBM VERSION OF A GREAT FILE LISTER/HANDLER\*

LIST62A .ARC 25984 Latest ibm version of Vern Bueg's great program

605568 Bytes in 8 Files 10240 Bytes free

\* \* \*

### VictorTalk

Address inquiries to: P. O. Box 50893  
Palo Alto, CA 94303

Subscriptions: Nancy Weaver

Editor: Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

Contributors: Don Clark, J. P. Hastey, John Knobelsdorf, Mary Jo Knobelsdorf, Jack Prichard.

VictorTalk is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in VictorTalk. Copyright 1987 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

VictorTalk is an independent publication not affiliated in any way with Victor Technologies, Inc. Victor is a registered trademark of Victor Technologies, Inc.

Submit Articles on MSDOS disks in ASCII or Spellbinder format, with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

## Yes, I want to join.

Enclosed is a check/money order for \$35.00 (foreign membership, \$40 in U.S. dollars) for my 1987 Victor\*Group membership.

Name \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

Make checks payable to:

## VictorTalk

P. O. Box 50893  
Palo Alto, CA 94303

City \_\_\_\_\_

State \_\_\_\_\_ ZIP \_\_\_\_\_

Country \_\_\_\_\_

Phone \_\_\_\_\_

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
P.O. Box 50893  
Palo Alto, CA 94303

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
State _____	
City _____ & Zip _____	
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. Alternatively, see VictorTalk, Vol. 4, No. 2 for a brief description of disks 61-64; Vol. 4, No. 3 for disks 65-71; Vol. 4, No. 6 for disks 72-77; Vol. 5, No. 1 for disks 78-81; Vol. 5, No. 2 for disks 82-84; Vol. 5, No. 4 for disks 85-86; and Vol. 5, No. 5 for disks 87-90.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40	
														*(34 & 35 are a two-disk set)						
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56*	57*	--	59	60	
														*(56 & 57 are a two-disk set)						
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	
81	82	83	84	85	86	87	88	89	90	__	__	__	__	__	__	__	__	__	__	__

1987 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, set of 6, \$12.00 ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_

1987 membership fee \$35 (receive all 1987 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR® TALK

TIME TO RENEW!  
VICTOR\*GROUP  
The box at the bottom of this page contains a renewal card for your subscription to Victor\*Talk. It is important that you fill out this card and return it to us as soon as possible. If you do not, your subscription will expire at the end of the year. We will be sorry to lose you as a subscriber. Please take the time to renew your subscription today.

USERS GROUP UPDATE  
VICTOR\*GROUP  
PO Box 50893  
Palo Alto CA 94303  
We are pleased to announce that we have a new Users Group. This group will provide a forum for users to discuss their experiences with our products and to share their ideas for improvements. We will be holding regular meetings and will also have a newsletter. If you are interested in joining, please contact us at the address above.

## 1988 RENEWAL NOTICE

Dear Subscriber:  
We are pleased to have you as a subscriber to Victor\*Talk. We are now in our 10th year of publication and we are proud to have a large and loyal readership. We are now offering a special discount on our 1988 subscription rates. If you renew your subscription today, you will receive a 10% discount on the 1988 rate. This offer is good for the next 30 days. Please take advantage of this offer by renewing your subscription today. If you do not, your subscription will expire at the end of the year. We will be sorry to lose you as a subscriber. Please take the time to renew your subscription today.

**VICTOR\*GROUP**

**P.O. BOX 50893**

**Palo Alto, CA. 94303**

**FIRST CLASS MAIL**

---

VICTOR\*GROUP

The San Francisco Bay Area Victor Users Group

---

**Please Remit To:**

VICTOR\*GROUP  
PO Box 50893  
Palo Alto CA 94303

Invoice Date: 09 Dec 87

Terms: Payable Upon Receipt

1 9 8 8  
MEMBERSHIP RENEWAL NOTICE

-----  
January 1988 through December 1988  
Annual Membership Fee for the San Fransico  
Bay Area Victor\*Group:

U.S. Residents:	\$35.00
All Other Countries:	\$40.00
U.S. Dollars Only	

Membership includes all issues of VictorTalk published during this membership period...Access by mail to our Software Library at 1/2 price... Group purchase discounts offered through Affiliated Computer Groups, Inc...Participation in our Bay Area monthly meetings to exchange ideas and information...Access to BAVUG (Bay Area Victor Users Group) Bulletin Board...

-----

Is mailing label on reverse side correct?      No      Yes

Please tear off this page and return it with your check to ensure that we maintain the proper listing for your membership.



# VICTOR<sup>®</sup> TALK

Bi-Monthly Magazine for Victor 9000 users

VictorTalk, Vol. 5, No. 6, Nov-Dec 1987

## TIME TO RENEW!

This is the last issue of VictorTalk for 1987. It is time to renew your membership so you will be a member of Victor\*Group in 1988. Your membership includes bi-monthly issues of VictorTalk, access to the BAVUG Bulletin Board, access to the Public Domain Library at reduced cost, and for members close to the San Francisco Bay area, monthly meetings. Please take the time NOW to fill out the renewal form on the preceding page (front inside cover) and mail it with your check today.

## USERS GROUP UPDATE

Recently, in reviewing the various listings of Victor Users Groups, a number of instances were found where the information was believed to be out of date. It was decided to update these lists by soliciting information directly from the user groups. A questionnaire, with the following format, was mailed to each of some 45 user groups:

Group Name:

Address:

Principal Contact:

Phone:

Is this group dedicated specifically to users of Victor 9000's and related products?:

Number of Current Members:

Annual Dues:

Do you maintain a Public Domain Software Library?:

If you publish a newsletter or any other type of publication, please complete the following:

Title:

Frequency:

If you have regularly scheduled meetings, please complete the following:

Date/Time:

For further meeting information, call (name/phone):

Please list any other information you feel pertinent to your users group:

Return completed form to:

Victor\*Group

PO Box 50893

Palo Alto, CA 94303

We thank those who responded to our questionnaire and particularly those who provided additional comments and material. We would greatly appreciate receiving additional responses. If you did not receive a questionnaire and are an active Victor users group, please submit the pertinent information in the suggested format. As additional responses are received, we will publish the information in VictorTalk. We gratefully acknowledge the capable assistance of Nancy Weaver, our BAVUG Secretary/-Treasurer, in preparing and mailing the questionnaires and handling the responses.

The responses received, thus far, are summarized (alphabetically by state and zipcode) as follows:

### Phoenix Area Victor Users Group

26618 So. Novajo

Sun Lakes, AZ 85248

Pat Elwonger (602) 895-7130

"Have had no meetings for several months and no one seems interested. Many have gotten rid of their Victors. Not I - I love mine and intend to use it until it dies. I have subscribed to VictorTalk and appreciate it very much."

### Orange County Victor 9000 User Group

12062 Valley View, Suite 218

Garden Grove, CA 92645

Don Henderson (714) 730-6015

35 members; maintain a PD library; no newsletter.

Meet 4th Wednesday 7:30PM at Mayward Siler & Assoc. at above address; contact Don Henderson, 1201 Drayton Way, Tustin, CA 92680, (714)-730-6015.

### (San Francisco) Bay Area Victor Users Group (BAVUG and Victor\*Group)

PO Box 50893, Palo Alto, CA 94303 (Nancy Weaver)

300 members; \$35 annual dues; maintain PD library; VictorTalk (about 24 pages) 6x/yr; meet first Wednesday, 7:30PM, Holiday Inn, Emeryville, CA, contact Boardman Moore (415) 283-3285.

VictorTalk Editor: Jack Prichard, Box 267, Pt. Reyes Station, CA 94956, (415) 663-8062.

Public Domain Librarian: John Knobelsdorf, Rt 2, Box 103, Bertram, TX 78605, (512) 355-2084. (Send request for free catalog or orders to Palo Alto address.)

Bulletin Board (BAVUG RBBS): (415) 574-7914, Franz Hirner, Sysop.

(BAVUG is a consolidation of three earlier groups: San Francisco Victor Users Group, East Bay Victor Users Group, and South Bay Victor Users Group.)

### Sacramento Victor Users Group

Attn: Somerby, Rm 340

1220 'N' St.

Sacramento, CA 95814

Ronald E. Somerby (916) 445-4521

18 members; Vol. stamp money in lieu of dues; maintain PD library; no newsletter; meet 4th Thursday, 7:00PM, contact Ron Somerby (916) 445-4521 or Roy Cousineau (916) 428-4544 (evenings).

#### Capital Area Victor Users Group (CAVUG)

PO Box 6255  
Washington DC 20017  
or

5708 33rd St. NW  
Washington DC 20015  
3-70 members ("depends on how you count"); "no longer hold Victor meetings per se but some members meet as a special interest group (SIG) of the Capital PC Users Group after its regularly scheduled meetings" ("For further information call for recorded info to (301) 656-8372 or write Capital PC Users Group, 4520 East-West Highway, Suite 550, Bethesda, MD 20814"); do not maintain a PD library; no longer publish a newsletter. ("Back issues of our newsletters are available for \$10 a set.")

#### Indiana Victor Information Network

3504 State Rd. 9N  
Anderson, IN 46012  
Bert Happel, (317) 643-0611 (evenings)  
19 members; no dues; no PD library; IVINewsDisk published 1 so far (a year ago); no meetings scheduled.  
"IVIN is an 'ad hoc' groups with no dues, organization, officers, or meeting. The objective of IVIN is to provide support to one another in our use of the V9000, and to provide timely notice of software and hardware 'special deals'."

#### Baltimore Regional Association of Victor Operators

1825 North Forest Park Avenue  
Baltimore, MD 21207-6506  
George Tyson, (301) 448-2362  
67 members; \$10.00 annual dues; maintain PD library; publish BRAVO News, 10 issues per year; meet alternate months (last was on November 10), contact George Tyson (301) 448-2362.

#### Memphis Area Victor Users Group (MAVUG)

4890 Mead Rd.  
Nesbit, MS 38651-9748  
Stan Crump (601) 368-5772  
No PD library; no newsletter; no scheduled meetings.  
"The status of our group is unknown at this time. The previous president resigned and turned the group over to me earlier this year. I plan to try to reorganize and hold regular meetings this winter. Depending on the size of the group, we may join an existing group and disband MAVUG."

#### Victor PULSE

PO Box 705  
Exeter, NH 03833  
Brad Chase (no phone)  
500+ members; no dues; maintain PD library; no newsletter.  
"A complete disk based catalog of all our disks is available for \$2.00."

#### Victor Circle

Various addresses  
No principal contact  
Upwards of 500 members; no dues, no PD library; NewsDisks published monthly(?); meet last Tuesday, 7:00PM, Philosophy Hall, Columbia University, New York, NY.

#### Victor Users of Western Pennsylvania

Box 4592  
Pittsburgh, PA 15205  
Larry Christy (412) 921-6900  
30 members; no dues; maintain PD library; no newsletter; no scheduled meetings.  
"We don't have regular meetings or publish any newsletters, but we do keep a library of public domain software and as much Victor related literature as we can find. We do whatever we can to help Victor users."

#### Philadelphia Area Victor 9000 Users Group

c/o Restat Systems, Inc.  
100 N. 22nd St. #123  
Philadelphia, PA 19103  
(215) 569-3461  
\$8 annual dues; "if we can get some help, we would like to prepare a newsletter"; "meetings held in November and December, later meetings will be scheduled for February, April and May (most likely on the third Wednesday)";

#### Puget Sound Victor Users Group

8025 W. Mercer Way  
Mercer Island, WA 98040  
Don Wacker (206) 236-1977  
18 members; annual dues \$10.00; "Have various disks from Victor Pulse, BAVUG, PUB, but don't edit our own"; PSVUG Newsletter published monthly; meet 3rd Thursday, 7:00-9:00PM, contact Don Wacker or Mary Wacker (206) 236-1977.

#### Problem Solvers of Switzerland

Wyttlenbachstr. 28  
CH 3013 Bern, Switzerland  
Ueli HAENNI 011-41/31/427156  
150 members; annual dues 40SFR (about \$30); do not maintain PD library, "just spread Brad's"; publish ViSi Club Informationen (ViSi is acronym for Victor Sirius) planned 4x/yr, effectively less, newest 1986 Nov!"; "one national meeting every year"; "several(?) local groups, call Ueli haenni or Eduard K. Mueller, in Swikold (031) 244244".  
"We're actually suffering from some Incredibly Bad Machine (IBM) that's taking its toll from our Victor friends."

\* \* \*

# VT CACHE

[VT-Cache (a trademark of Valid Technologies) is an interesting new software accelerator for the Victor 9000. The following description has been paraphrased and condensed from their flyer. - Ed.]

Programs that require frequent access to disk can be significantly speeded by using disk block caching. This is a technique that uses some portion of RAM to store a number of disk blocks. This speeds access for reading and writing, by avoiding reading and writing directly to disk. Thus, its performance can approach that of direct memory access.

## Generic Features of Disk Caching

- o Assures disk data integrity  
The cache approach provides automatic backup to the non-volatile physical disk. Thus, it is unnecessary to manually write the files back and the risk of data loss due to power failure is reduced.
- o Executes programs faster
  - o As much as 14 times faster on flexible disk systems
  - o As much as 2 times faster on hard disk systems
  - o Faster MS-DOS command processing
- o Increases life of both flexible disks and hard disk
  - o By reducing the number and duration of disk start-ups and rotations, head movement and head contacts with diskettes
- o Makes operation program-transparent
  - o Can be used by all programs that use disk memory
  - o Selection of files placed into the "fast disk" is completely automatic, rather than manual
- o Operates with both flexible and hard disk systems
  - o Caching may be enabled or disabled for each drive individually
  - o Hard disk drives or partitioned hard disk drives up to 32 MB supported

Design features unique to VT-Cache (14 user-definable parameters, each with multiple options):

For a particular disk drive you can:

- o Enable or disable cache on the drive
- o Select the number of sectors to prefetch on disk read
- o Select the number of sectors to write on disk read
- o Select the method for controlling sector writes
- o Select the number of sectors to buffer in cache memory before writing to disk
- o Select the number of sectors to write to disk while

there is no keyboard activity

- o Select the keyboard time interval
- o Select the method for controlling file close

For the total system, you can establish global values for:

- o Cache size
- o High/low loading
- o Multi-sector buffer limit
- o Write/clear key
- o Tone/no tone upon pressing write/clear key
- o Keyboard status counter

Compared to Ram Disk software accelerators, VT-Cache has the following additional abilities:

- o Selection from various sector buffering options
- o Use system idle time to write modified sectors back to disk
- o Write out sectors on disk read
- o Force sector writing to disk on closing a file
- o 14 user definable parameters compared to 2 or 3
- o Programmer interface enables software developers to incorporate control of disk caching into their own software
- o Support disk drives up to 32 MB
- o Automatic insertion of data into cache memory, thereby avoiding explicit copy of data into memory

For further information contact:

Valid Technologies  
1401 Pioneer Parkway West, Suite 107  
Arlington, TX 76013-6251  
(817) 261-0998

- - -

Al Bullock has done limited testing of VT-Cache and has obtained speed-up factors approaching those mentioned above. He notes that VT-Cache will apparently only work properly on the Victor 9000 without Plus-PC board and with MS-DOS 1.25 or 2.11. (Purchaser must specify version desired; a version for MS-DOS 3.1 is planned for Spring 1988 release.) He states the retail price is \$139; his price to BAVUG members is \$100.

Bullock & Associates  
39522 Benavente Pl.  
Fremont, CA 94538  
(415) 651-6183

\* \* \*

# DIRECTORIES / Jack Prichard

Regardless of the particular use made of a computer, a common recurring need is to display and/or print a directory of the files on a floppy or hard disk. Each version of MS-DOS has a different revision of DIR (as an internal command) to answer this need. (Because of dwindling interest in CP/M, no coverage of similar CP/M programs has been attempted here.)

## Directories Provided with MS-DOS

MS-DOS 1.25 (see "Operator Reference" page 9-7), provides only two switch options: /P, the page mode, wherein the display pauses after the screen is filled and /W, the wide display mode, wherein only the file names are displayed, five per line. No provision is made for listing the files in any order other than the usually random-appearing order of the directory. However, DIR, like essentially all commands and public domain programs to make directory listings, does have the ability to use wild-card characters: ? for any one character (or none), and \* for multiple characters (or none).

MS-DOS 2.11 (see "User's Guide" page 7-108), provides the same two switch options as MS-DOS 1.25, but adds the capability of dealing with subdirectories. A new external command, SORT, is available to provide help in sorting the directory; however, the help is limited by SORT being column oriented (sorts on the data starting at the specified column). (Sorting by date in the usual US format of month-day-year and time in a 12-hour AM/PM format is not very useful.) MS-DOS 2.11 also provides a new external command, LS, (List Files) which produces a sorted directory. Different switches are allowed to: sort backward, sort by file extension, sort by date and time, recursively list files in subdirectories, "together list" (files not separated by directory), and list in long format (including file attributes). While this new command does provide additional useful formats for directory listing, it is not completely satisfactory for most users. For example, except for the long format, only filenames are listed (no size, date or time).

MS-DOS 3.1 (see "Reference" page 6-43), provides substantially the same facilities with DIR, SORT, and LS as does MS-DOS 2.11. [The documentation adds an example of a LS /L (long listing).]

## Directories Available in Public Domain Library

The general inadequacy of the commands provided with MS-DOS has led to a proliferation of programs to fulfill this need. Fortunately, many are now available in public domain

software. The problem is to locate them and determine which is most suitable to meet a particular need. The purpose of this article is to assist by summarizing the characteristics of these programs in a convenient form. While an effort has been made to include all pertinent programs of the public domain library [of the Bay Area Victor User's Group (BAVUG)], undoubtedly some have been missed. Notice of such omissions, as well as of errors, in these tables would be greatly appreciated.

It has been our intention to include, not only programs that display or print the directory (or a specified sub-set of directory files) of the specified disk, but also programs whose purpose is to rewrite the directory of a disk in a specified sorted order. (This order is typically by: filename, extension, size, date and time of last change, or the reverse order of any of these.)

Two classes of programs that also relate to directories, but that are not included here are:

- o Programs that display the directory of librated or archived files, without "unlibrarying" or "unarchiving" the files themselves.
- o Programs that provide other directory-related functions such as changing filenames, date/time, attributes, etc.

Information on the public domain programs that are listed in this article were found using these steps:

- o Referring to the DIR category in Tables 1 and 2 of "Public Domain Library Index", VictorTalk, Vol. 5, No. 5, Sep-Oct 1987. (Actually this index was assembled using essentially the steps that follow.)
- o Computer searching an unpublished cumulative listing of public domain disks 1 through 77 (ALL77.ASC), prepared by the public domain librarian (John Knobelsdorf), for files having DIR as part of their filename or otherwise suggestive of a directory connection.
- o Manually searching the Public Domain Library Catalog, Parts 1 and 2, (covering disks 1 through 84).
- o Printing and manually searching the DISKID file and directory of each disk believed to possibly contain the desired files.

- o Printing and studying the .DOC file of each candidate directory file.
- o Running each of the candidate files to check its performance and printing sample directories using the available printers (Epson MX-80 and Diablo 630). (It was considered impractical to attempt to examine every one of the many program/options available or to check the operation with a wider variety of printers.) All testing used MS-DOS 2.11. Many programs have been previously run using MS-DOS 1.25; however, no concerted recheck has been made using MS-DOS 1.25. All working programs will likely also run under MS-DOS 3.1; however, this has not been verified by actual test.
- o Where the available options include different formats or numbers of columns, additional lines of information are given under "Column". However, information in other columns is not repeated, unless also changed. (Although these extra lines may extend opposite other entries, they refer to the .COM or .EXE entry.) The first line normally defines the default format.
- o The column labeled "Ver." gives the version (or, in some cases, revision) number of the program.
- o Where remarks are necessary, they are given on the facing (odd numbered) page, following the alphabetic order of the groups of files (or commands). In the interest of conserving space, horizontal alignment of the remarks with the facing page information has been abandoned.

#### Description of Tables

Information on the programs and commands covered is contained in Table 1. Explanations of the table follow:

- o Entries are arranged alphabetically by filename (or command).
- o Entries having the same filename are grouped and listed in order of public domain disk number (or, in some cases, by date, where meaningful).
- o Within groups the listing order is by filename extension. The executable version of a file (.EXE or .COM) is listed first; the related documentation file (.DOC), if any, is listed second; other related files (such as .ASM or source code) are listed last. Unless size and last change date/time agree exactly, a separate entry is made for each filename. When only time is different, the variant time is noted under remarks.
- o The outputs produced by the programs and commands are characterized by five columns of information: sort options, additional options, heading information, columnar information, and footing information. The symbols used for these characterizations are defined in Table 2.
- o Option symbols in Table 1 are listed alphabetically and are aligned vertically to facilitate comparisons between programs.
- o Heading, column, and footing information is listed in the order of occurrence to suggest the listing format appearance.

- o In the remarks portion of Table 1 and in the subsequent text, the ambiguity of duplicate filenames has been eliminated by coupling the filename with the public domain disk number(s) on which the specific filename appears. For example: SDIR.COM 44; SDIR.COM 46,50; etc.

#### Conclusions from Tables

What does this all mean? It means that there are many programs of varying capabilities from which to choose. The choice of which programs to use will depend on the type of work you are doing, your computer configuration, and your personal preferences. Perhaps relating my situation and what I have found useful will help.

Since taking over the chores of editor (writer, publisher, whatnot) of VictorTalk, my work has centered around those activities, there being little time left for anything else. Thus, my work has involved the use of Spellbinder word processing for writing, editing, formatting, etc. and the use of MS-BASIC in connection with the series on that subject in VictorTalk.

The computer equipment includes:

- o Victor 9000 with dual single-sided floppies, 896k RAM (Valid Technologies C3-1000 memory board with clock), 600K of which is normally used as drive C: using White Crane Systems Ram Drive. This is the workhorse on which most work is done.
- o Victor VI with single double-sided floppy and 10MB internal hard disk, used mainly for reading double-sided Victor disks and IBM-formatted disks.

(Text continued on page 10)

TABLE 1. PROGRAMS AND COMMANDS FOR LISTING, PRINTING AND SORTING DIRECTORIES (Page 1 of 4)

Filename (or Command)	Size	Last Change	PD No.'s	Sort Options	Add'l Options	Head	Column	Foot	Ver.
COVER .COM	1088	01/19/84 19:53	08	A	F U	TFD	4 FB	-	
COVER .DOC	1614	02/19/84 16:19	08						
COVER .COM	1152	12/09/84 18:35	31	A	F U	TFD	4 FB	-	
COVER_TB.COM	1152	09/03/84 16:59	49	A	F U	TFD	4 FB	-	
COVER_TB.DOC	640	05/26/85 16:12	49						
D .COM	1280	09/03/84 22:23	46	A	U	-	4 FB	BNF	
DDIR .COM	896	09/29/85 09:57	61						
DIR	39168	01/13/83 13:26	MS-DOS 1.25 (Int. command)	N	F PUW F PUW	- -	1 FBD 5 F	N	
DIR	59584	02/13/84 12:53	MS-DOS 2.11 (Int. command)	N	F PUW	RLP	1 FBD 5 F	NB	
DIR	78016	07/07/86 17:16	MS-DOS 3.1 (Int. command)	N	F PUW	RLP	1 FBD	NB	
DIRSORT2.COM	11008	09/02/85 14:51	61	A	-	-	-		2.0
DIRSORT2.DOC	3584	09/02/85 14:51	61						
DISK .COM	5632	01/01/80 00:17	53	N	AD PUW	VARL	1 FBD 1 FB%DA 5 F	BNF BN%F NF	2.1
DISK .DOC	9472	01/01/80 00:18	53						
FDIR .COM	640	01/01/80 00:10	04,24,55	X	L	-	1+6 F	-	
FDIR .COM	512	11/29/83 16:39	08	X	L	-	1+6 F	-	
LF .COM	9600	10/24/85 00:51	61	ADNRSX	DEF LP W	VALNB	1 FABD 1-8 F	F	2.9
LF .DOC	4480	12/29/85 16:57	61						
LIST .EXE	30336	01/01/80 00:20	46	AD R	UWZ	-	5 F 1 F 1 ABDF	N	
LIST .DOC	1157	10/12/84 18:25	46						
LS .EXE	31480	11/23/83 11:31	MS-DOS 2.11	AD R X N	A UW	-	5 F 1 F 1 ABDF	- - -	
LS .EXE	17346	01/22/86 11:54	MS-DOS 3.1	AD R X N	A UW	A	5 F 1 F 1 ABDF	- - N	
LS .EXE	16640	03/06/83 15:13	11,29	A X	AD HL A L W	- -	5 F 1 ASDF 2 AFCD	N	1.0
LS .EXE	16640	06/15/83 14:58	18	A X	AD HL	-	5 F	N	1.0
LS .EXE	12544		61	ADNRSX	AD FHLPWZ	-	1-5 F	-	
LS .DOC	7539	08/10/85 19:14	61				1 ASDF	BN	

TABLE 1. PROGRAMS AND COMMANDS FOR LISTING, PRINTING AND SORTING DIRECTORIES (Page 2 of 4)

	Remarks
	COVER.COM 08 and COVER.COM 31, while differing slightly, perform the same function. They provide a printed disk "cover" directory on an Epson MX-80 printer in compressed print. Program prompts for drive to list and title; these are only options (defaults are shown in table).
	COVER_TB.COM 49 is similar to COVER.COM 08 and COVER.COM 31, but is for use with a Toshiba P1350 printer.
	DDIR.COM 61 does not run; gives message "COMMAND Problem".
	DIRSORT2.COM 61 sorts directory alphabetically and writes it back to disk. Long file for doing a simple function.
	DISK.COM 53 requires MS-DOS 2 or later; extensively documented; many options (and restrictions on option combinations); however, provides only unsorted listings. Percent slack space based on 2k byte clusters on hard disk, so does not apply to floppy disk; ram disk percent slack space based on 256 byte clusters, however. Apparently not possible to list second level subdirectories.
	FDIR.COM 04,24,55 are identical, except last change time of FDIR.COM 24 is actually 00:48.
	FDIR.COM 08, while slightly shorter, appears to produce identical listings to FDIR.COM 04,24,55.
	LF.EXE 61, in addition to providing the options and information listed, allows for turning on or off almost all information selectively (including: files, directories, date/time, size, attributes, pause, number of columns from 1 to 8, heading and footing information). It also provides a unique capability: to not only include, but also exclude, file groups by the use of wild cards.
	LIST.EXE 46, contrary to LIST.DOC 46, does allow sorting in its long format. Very long file for doing much less than many other directory files!
	LS.EXE 11,29 are identical; LS.EXE 18 is trivially different from LS.EXE 11,29 (except for date and time). Documentation for LS.EXE 11 is contained in DISKINFO.TXT on PD #11; documentation for LS.EXE 29 is contained in DISKID.TXT on PD #29; no separate documentation has been found for LS.EXE 18. Documentation appears to contain some errors: for example, -a gives files in alphabetically sorted order, not in stored order. PPRINT.EXE 11,29 can be used to print the directory saved in an intermediate disk file.
	<p>LS.EXE 61 includes options not found in any other programs (nor included in Add'l Option column):</p> <ul style="list-style-type: none"> <li>o Files only ending in .BAT, .COM, or .EXE listed.</li> <li>o Files only ending in .COM or .EXE listed.</li> <li>o Directories listed before files.</li> <li>o Stream output (filenames only, separated by commas).</li> </ul>

TABLE 1. PROGRAMS AND COMMANDS FOR LISTING, PRINTING AND SORTING DIRECTORIES (Page 3 of 4)

Filename (or Command)	Size	Last Change	PD No.'s	Sort Options	Add'l Options	Head	Column	Foot	Ver.
SD .COM	1280	10/08/83 15:35	20	AD SX	A	VRDB	2 FBD	-	1.1
SD .ABS	607	10/09/83 15:35	20						
SD .HEX	2657	10/09/83 15:35	20						
SD .COM	3328	09/03/84 22:23	46	ADN SX	PUW	VLRDOP	4 FB	BNF	2.6
SD .DOC	512	09/03/84 22:23	46				2 FBAD		
SDIR .COM	1616	05/29/83 10:50	44	AD X	PUW	V	1 FBD	NAB	2.1
SDIR .DOC	1310	03/09/85 20:11	44						
SDIR .ASM	7175	05/29/83 10:44	44						
SDIR .LST	44893	05/29/83 10:49	44						
SDIR .COM	1584	09/10/83 10:39	37,38,39,50,55	ADN SX	A PUW	AVORD	2 FBD	BNF	1.28
SDIR .DOC	1536	11/10/83 14:18	37, 39,50						
SDIR .COM	1400	10/09/83 16:07	19,21	ADN SX	A PUW	AOVDRLB	2 FBD	N	2.3
SDIR .DOC	3440	09/28/84 01:29	19,21						
SDIR .ASM	19770	10/09/83 16:46	21						
SDIR .COM	2779	09/04/84 21:42	46,50	ADN SX	A PUW	AVOLDR	2 FBAD	NBF	2.50
SDIR .DOC	2748	09/04/84 22:03	46,50						
SDIR4 .COM	3584	01/01/80 02:32	36,45	ADN SX	PUW	VLRDO	4 FB	BNF	2.6
SDIR4 .DOC	571	03/09/85 12:21	36,45				2 FBAD	BNF	
							6 F	BNF	
SDL .COM	3072	09/03/84 22:23	30	ADN SX	A PUW	VRD	1 FBD	-	3.0
SDL .DOC	2560	09/03/84 22:23	30			VRD	2 FBD	-	
						VRD	4 F	-	
SORTDISK.COM	1136	09/04/84 20:23	46	AD SX	-	-	-	-	1.00
SORTDISK.COM	1120	09/10/84 02:50	49	AD SX	-	-	-	-	1.10
SORTDISK.COM	1152	01/01/80 00:13	61	AD SX	-	-	-	-	1.10
SORTDISK.DOC	947	09/10/84 03:18	49						
TD .EXE	18048	04/05/84 20:11	31,61	AD S	-	-	1 FBDA	BN	
TD .DOC	128	12/09/84 18:50	31						
TD .DOC	203	01/01/80 03:18	61						
TD .EXE	18048	09/03/84 22:23	46	AD S	-	-	1 FBDA	BN	
TREE .COM	1536	09/01/84 22:53	46	-	U	L	1 F	-	
TREE .DOC	772	09/01/84 17:03	46						
TREEDIR .COM	7680	09/02/84 16:16	46	N	U	V	1 FDB	AB	1.0
TREEDIR .DOC	1536	09/02/84 16:16	46						
WHEREIS .COM	649	02/12/84 20:42	46	N	LW	-	1 PF	-	
WHEREIS .DOC	197	05/31/84 20:42	46						
WHEREIS4.COM	1244	02/04/86 08:40	87	N	PUW	-	1 PFD	-	
WHEREIS4.DOC	1792	07/19/87 08:16	87						
XDIR .COM	1920	02/25/84 09:20	60	N	A PU	LP	1 FBDA	N	



TABLE 1. PROGRAMS AND COMMANDS FOR LISTING, PRINTING AND SORTING DIRECTORIES (Page 4 of 4)

Remarks
SD.ABS 20 is the documentation file for SD.COM 20.
SD.COM 46 documentation (SD.DOC 46) is skimpy; program appears nearly identical to SDIR4.COM 36,45 (also identified as version 2.6); behavior appears identical.
SDIR.COM 19,21 are identical, except last change time of SDIR.COM 21 is actually 16:46. While this file is generally quite useful, option /E must be used to avoid system hang up. SDIR.DOC 19,21 are identical.
SDIR.COM 50 of SDIR.COM 46,50 is named SDIR211.COM on PD #50, as it was designed for use with MS-DOS 2.11.
SDIR.COM 50 of SDIR.COM 37,38,39,50,55 is named SDIR125.COM on PD #50, as it was designed for use with MS-DOS 1.25.
SDIR4.COM 46,50 and SDIR4.DOC 46,50 documentation is unclear; while COLUMN [2,6, or 4(default)] and SORT options (/No /Size /Date /eXt) work correctly, other options need further explanation (and/or fixing): /CIs: Hangs up the system. /WriTe (and /W also): Prints ok on Diablo 630 printer; produces condensed characters on Epson MX-80 printer. /Pause: Eliminates pause that is obtained by default at the end of screen. /All: Provides list of directories as well as files (including \. and \.. directories). The use of any option restricts directory listing to that of the default drive. Printout obtained by usual Alt-P produces undesired characters in place of displayed "ruled" lines.
SORTDISK.COM 46,49,61 differ slightly; performance appears identical. They all sort the root directory on the specified drive and write it back to disk without displaying or printing the directory. SORTDISK.DOC 49 appears to apply to SORTDISK 46,61 as well.
TD.EXE 31,46,61 differ trivially in coding; performance appears identical.
TD.DOC 31,61 differ only in that a portion of two last lines is missing in TD.DOC 31.
TREE.COM 46 lists the directory paths to the specified (or default) directory; optionally, it also lists the files in the directory. (Both display and printout have annoying waste of space, with unnecessary double and triple line spacing. WHEREIS.COM 46 is more useful and compact.)
TREEDIR.COM 46 provides only the option to start listed directory path with current directory; default start is with root directory; has no provision for wild cards; gives file size totals by sub-directory and directory. To obtain printed output, use intermediary listing file.
WHEREIS.COM 46 provides directory tree for any specified file(s) on default drive only. This appears to be best program for locating files when the directory is not known.
WHEREIS4.COM 87 is similar to WHEREIS.COM 46 except that: it requires the use of EMIBM.COM, can search any specified drive, and, additionally, provides date and time for each file.
XDIR.COM 60 has no companion XDIR.DOC 60 file; it provides only unsorted files in directory, not sub-directories; only option apparently available is P, obtained by switch /P (or /p).

TABLE 2. - DEFINITIONS OF SYMBOLS USED IN TABLE 1

Note: The symbols used are listed alphabetically here; in Table 1, except for options, they appear in their order of occurrence in the directory listing described.

Sort Options: (Defaults in some cases)

- A = Alphabetic sort
- D = Date/time sort
- N = Non-sorted
- R = Reverse order (of other sort criteria chosen)
- S = Size sort
- X = Extension sort

Note: These symbols are generally similar, but not always identical, to those used in the switch options (since they vary).

Additional Options: (Defaults in some cases)

- A = Hidden files included
- D = Directories only listed
- E = Exclude specified files
- F = Files only listed
- H = Help
- L = Lower case listing
- P = Pause when screen full
- U = Upper case listing
- W = Wildcards (? and \*) supported
- Z = Recursive directory listing (Lists, with path, all matching files in current directory or subdirectory below)

Head: (Information provided in heading of directory listing.)

- A = Author
- B = Bytes (free)
- D = Date/time (of listing)
- L = Label (volume)
- N = Number of files
- O = Option list
- P = Path to directory
- R = Drive
- T = Title
- V = Version, release, or revision date

Columns: (Information provided by columns in body of directory listing.)

- Number indicates number of groups of columns
- A = Attributes
- B = Bytes
- C = Clusters
- D = Date/time of last change
- F = Filename (common to all)
- P = Path
- % = Percent slack space

Foot: (Information provided at end of directory listing, or, in some cases as subtotals.)

- A = Bytes (actual)
- B = Bytes (allocated)
- F = Bytes (free)
- N = Number of files
- S = Sectors
- % = Percent slack space

- o Epson MX-80 printer, used principally for normal printouts, including directory printouts.
- o Diablo 630 printer, used principally for drafting, editing, formatting and printing of final "camera ready" pages for VictorTalk.

Evaluation of the usefulness of the various directory programs and my choice of programs is as follows:

- o Except for very short directories or short sub-sets selected using wild cards, DIR is little used, because of the lack of any built-in sorting ability. About the best that can be said for it is that "it's there" (an internal command that is part of MSDOS.SYS).
- o The MS-DOS command SORT in conjunction with DIR does provide filename, extension, and size sorting (and reverse order sorting), but fails to

produce a sensible date/time sort (because of our illogical date/time system). When compared to other directory programs, this combination is unnecessarily awkward and incomplete.

- o The MS-DOS command LS does directly provide a sorted directory listing. However, it has these shortcomings:
  - o No size sort.
  - o No total bytes or free bytes listing.
  - o No compact (2-column) long-form listing.
  - o Programs are quite long compared to others producing superior results.
- o SDIR.COM 46,50 meets most needs well. It provides for all desired sorts, supports subdirectories, provides volume label, compact (2-column) long-form listings, supports wild cards, and provides number of files, allocated bytes and free bytes together in the

footing (where it is most convenient to use). (Although they are seldom needed, attributes are also listed.)

- o SDIR.COM 19,21 does essentially the same things as SDIR.COM 46,50; however, it has one important shortcoming: the option /E must be used to avoid system lock-up. Although, seemingly a minor defect, this defect is a trap that sooner or later will catch you; the consequent waste of time to reboot is annoying! (Attributes are not listed.)
- o SDIR.COM 37,38,39,50,55 also does essentially the same things as SDIR.COM 46,50, except it was not intended to support subdirectories and does not show the volume label, nor does it list attributes.
- o WHEREIS.COM 46 is particularly convenient to quickly find the path to a file or group of files. While it can provide a complete listing of all subdirectories and their files, it was not designed for that use. (It provides only path and filename information and in unsorted order.)
- o The three files, COVER.COM 08, COVER.COM 31 and COVER\_TB.COM 49, all produce a neat, compact directory (with only filename and size columnar information) that fits in a disk envelope. However, such directories are of limited use; directories on 8 1/2 x 11-inch sheets, filed in a three-ring loose-leaf binder, better meet most needs.
- o Except for the rare occasions when a special capability (such as reverse order sorting) is desired, the remaining programs and commands in Table 2 are of little use to me.

For my work, filename or date/time sorted directories are most used. To simplify obtaining directory displays, the following six .BAT files are used:

```
A.BAT    SDIR A: /A
B.BAT    SDIR B: /A
C.BAT    SDIR C: /A
DA.BAT   SDIR A: /A/D
DB.BAT   SDIR B: /A/D
DC.BAT   SDIR C: /A/D
```

When needed for special purposes, additional ad hoc .BAT files are used. These all reside in drive C: (the RAM drive), as does SDIR.COM 46,50 and WHEREIS.COM 46. Thus, when current directory information is in RAM, no disk access is required to obtain a directory display for any drive.

At boot-up, these files, along with frequently-used

Victor-provided MS-DOS files, Spellbinder, and a number of frequently-used Public Domain files, are all loaded into drive C: by AUTOEXEC.BAT. As needed, other files are loaded into drive C: to speed their subsequent use. When additional space is required in drive C:, files that are no longer needed (or less frequently needed) are deleted.

\* \* \*

#### HELP WANTED

As was stated in "Public Domain Library Index" (VictorTalk, Vol. 5, No. 5, Sep-Oct 1987), a series of articles on various categories of Public Domain programs is planned for publication in VictorTalk. The foregoing article on "Directories" is the first of that series. A second article on "Date/Time" is in preparation. As was suggested, similar articles on such additional categories as Screen Control, Printer Control, Fonts/Keyboards, Music, Games, Communications, Word Processing, Data Bases, etc. are being contemplated. Actually, useful articles could be prepared on almost all of the categories used in describing the contents of the Public Domain library. Undoubtedly, some further subdivision may be desirable; in other cases, categories might be usefully combined (as in Fonts/Keyboards).

The preparation of such articles is certainly a great deal more than a one-person task! Volunteers are needed to share the load and expedite the task. Help is particularly desired from persons with some degree of expertise, or at least experience, in the subject matter. It should be stressed, however, that inventiveness or originality is not needed. The articles should be surveys of the material available in the public Domain library. The information should be of a comparative nature, telling what is good, what is bad, where the documentation can be found, etc., along the lines of the foregoing "Directories". Where appropriate, personal experience with the programs might be summarized as an aid to other users.

To volunteers who commit to prepare a survey article suitable for publication in VictorTalk, we will provide, at no charge, those Public Domain disks (beyond those you already possess) needed to fulfill your commitment. (This will generally be limited to the first volunteer for a particular category.)

To volunteer, please call or write to the editor (see masthead, page 24) with particulars: category, disks required, ideas, suggestions, comments, etc., and estimated date of completion. As always, the editor will be responsible for final editing, formatting, etc.; hopefully, this work will be kept to a practical minimum.

\* \* \*

## QUESTIONS AND ANSWERS FOR THE VICTOR 9000 NOVICE / Sam Martin

[This article is "reprinted" from Victor Circle NewsDisk 25 (assembled and edited by Fuzzy Bear), complete with its original editorial notes. - Ed.]

[NewsDisk editors, BBS sysops and user group directors always face the problem of balancing the needs of a diverse range of interests. One of the most dangerous traps is talking "over the heads" of the new users. Not everyone can "fly with the eagles," ...some are doomed to muck about with the bears. From the very beginning of the NewsDisks, Dale has been concerned about offering items of interest to the novice and keeping the "mechanics" of the NewsDisk simple enough so that new Victor 9000 owners would be able to learn from, as well as participate in the creation of, the NewsDisks.

We are fortunate to have an article from Sam Martin which should prove of inestimable worth to the absolute beginner. Sam has an exceptional memory and he remembers every step along the road from absolute tyro to his present status as a V9000 "power user". We guarantee that new (and some old) Victor users will avoid frustration with some of the tips in this article. Fuzzy Bear even learned something while editing this! ]

### 1. What should I do when my cursor mysteriously "freezes"?

- (1) Turn printer on; put it "online". (Locked "handshake".)
- (2) Press ALT-PAUSE (or S). You may have inadvertently turned on the pause toggle, or forgotten to turn it off.
- (3) If the program you are running has any "toggle" keys, try pressing them.
- (4) You can try C, A, Esc, and whatever is the exit code for your program (often X or X, or Q or Q, or "exit") but usually these don't help.
- (5) As a last resort, you have to reboot from the button. The keyboard is disabled so you can't use Boot.com; under DOS 3.1 use the keyboard boot (LOCK ALT pad-dot).
- (6) To avoid future recurrences: through Keygen check the keyboard you were using to see if one of the codes is garbled or mistaken. (Or, recopy the file that came with the original program disk.)

Perhaps you forgot to Modcon to the keyboard required by the program (Keygen and Efont are quite sensitive). Make a ".bat" file to bring up the program, for example "E.bat": "Modcon Standard.kb [save]", "Efont", ["Modcon save"].

### 2. Why does my "calculator" program sometimes freeze?

It may be that you have (probably inadvertently) tried to divide something by 0, which is a no-no (DOS 2.11 Manual 8-0.). Some programs (e.g., DOS Calc) prevent that from happening, but others do not. Spellbinder's elegant CALC will hang if you try to divide BY or INTO zero; the only way to unfreeze the keyboard is reboot, losing all your text and any figures you have calculated for it.

Victor's Calc will not hang when you try to divide into zero, but it will if you try to divide anything OTHER than zero into "0.00". Back to the pocket calculator!

### 3. What should I do if the cursor disappears/misbehaves when I have just started a program?

If it's just that you can't see the cursor, try reducing the level of contrast. Some programs lower the intensity of the cursor; if on top of that you have lowered the contrast, it may make the cursor disappear altogether. (The cursor still works, it's just invisible.) Notice that contrast "up" DIMS a low intensity and raises a high, contrast "down" does the opposite, reducing the difference. (Brightness is independent.) If that's not the problem, exit (boot if you must) and make sure you Modcon to the proper keyboard before entering the program. But first, command "prompt", which brings back the system prompt if it has been changed by a batch command; if you want the subdirectory given after the drive designation, type "prompt \$p\$g".

### 4. How can I retrieve a cursor that gets imprisoned on Line 25?

That depends on how it got there. Try "Leave25.bat"; it will usually put your prompt back home with a clear screen, and it is a good way to clear Line 25 even if you aren't on it. (Clear25.com will be frustrated by a prompt that keeps rewriting Line 25.) If you got onto Line 25 with

## FROM THE BOARDS

Set25.com or the like you may have trouble getting a prompt for "Leave25". Try spacing across the line (it will probably produce blank reverse video) and do a carriage return at the end. Another technique for escaping from Line25: go into a program that rewrites the screen (such as a WP); then exit, and your prompt should be back where it belongs. For the commands mentioned above, and more information, see Line25.arc on The PUB.

### 5. How can I speed up the word processors?

Use the latest version of Speedy (Speedy4.com); command "Speedy4 /h". Put the command in your Auto-exec.bat. If you are using WordPerfect, use SWP.com. [Fuzzy Bear's Note: SWP.COM no longer required if you have the July 1987 version of SP 4.2] These are widely available and much appreciated. Editors that frequently rewrite the screen are inherently slow. You can speed them up by doing everything in RAM; set up a Ramdrive (Guy Gordon's recommended), copy your essential WP files there for each session, and work in that drive. You can do all that with a batch command, once you have it figured out. See Manual on "---.bat" commands.

### 6. How can I get rid of unwanted screen residue from a resident "interrupt" program like Smartkey?

Do anything that will produce a screen rewrite (try page-down and page-back, "help" and back, or "status" and back). The "view" (screen-print) commands are good; in Spellbinder, use "v1" (view one line). Spellbinder often clears with just Esc. If you are in DOS, command "cls".

### 7. I like the Timeup and Timeup2 commands, but sooner or later they split the screen, reversing the two halves. Any help?

Get the newer Timeup12.com. It will not split the screen. On the other hand, it puts the time more to the center of the line than you may like, depending on the programs you run. Too bad you have to reboot to take Timeup off the screen; it seems to be permanently resident, once invoked. Can someone create a "Timeoff.com"?

### 8. Does the Concat utility work? (Not for me.)

Yes, it works, if you follow the directions carefully. But you don't need it. What you want can best be done by using the Copy command with appropriate choices. See the Manual.

### 9. The Graphics Tool Kit manual makes frequent mention of "VT52". What is "VT52"? Presumably a Video Terminal numbered 52?

On Disk 2 there is a 128-character set called "VT52T.chr". (Is the second T for "Test"?). The usual characters, except for #127, which is the paragraph mark, and the first 31, of which #1 and #2 are blank and the rest peculiar: #4 superscript-1 with slash (left side of one-half), #5 super-3 with slash, #7 super-7 with slash, #8 degree, #9 plus-minus, #10 right-arrow, #11 small triple-dot, #12 division-sign (colon with hyphen), #13 down-arrow, #14-#21 dashes descending like stairs, #22-#31 subscript numerals 0-9 (Esc is subscript 5).

### 10. What is the difference between "Save[.kb]" and "Saved[.kb]"?

Both versions confusingly turn up in the Victor manuals. Functionally, there is no difference; the operating system saves your keyboard/characters either way, but the file it creates (or overwrites) for "Saved" will be kept separate from one named "Save". Don't use "Saved" unless you want to keep one keyboard as special; in that case, why not use the original name? In fact, why not use the original name instead of "Save"? (The original files do not have to be on the disk at the time. The formula saves what you are using.) To ascertain the systems on the boot disk: Modcon (to anything) with Save.kb Save.chr, and use Keygen to check the "save" files. Incidentally, there is nothing sacred about "SAVE". You can substitute any word, and that will become the name of the saved file; if you command "Modcon Etc.kb Old.kb", there will be a file "Old.kb". The safest thing to do with Modcon is avoid the save operation altogether; modcon to one specific file at a time, and modcon back to the earlier file(s) by name. Your operating system character set is probably called "Intl'01.chr" but forget the apostrophe (see below, 12). Rename it "Standard.chr" (to go with "Standard.kb") and you will be happier.

### 11. A handy trick:

If you are doing a batch file to get in and out of a program that requires special keyboard or character set, do this:

```
modcon special.kb [OR special.chr] temp.kb
[OR temp.chr]
[program call]
modcon temp
del temp
... [whatever else you want afterwards]
```

You could use "save" instead of "temp" but "temp" reminds you to delete the original keyboard once you are back into it. A wise move, and a space saver.

12. How come the Victor DOS-2 disk gives us "Int!01.chr" when filenames with apostrophes (while not forbidden by the DOS manual) are unacceptable to Keygen, Efont, and the various directory commands?

Somebody goofed. Best solution: rename "Int!01.chr" as Intl01.chr. 'Tain't easy, except with Disk Tools. A do-it-yourself approach: "Ren Intl?01.chr Intl01.chr". You can use the same technique for other filenames that get garbled beyond easy manipulation. It works for deletion, too: if your directory is cluttered with a file "abc de.doc" that is inaccessible because of the space and you want to get rid of it, simply command "Del abc?de.doc".

13. When interrupted by a disk error that has to be "aborted", is my work totally lost?

Not necessarily. If there is a temporary Write file set up and you have written to it, that part is intact. You can see how many bytes are in the file with the command "Dir:[D]\*.\$\$\$". Close the "\$\$\$" file (the Spellbinder command: "wd") and read what is there; or, read and then close. Be cautious, for when a WP closes a Write file it makes the original file a ".BAK" and deletes the earlier ".BAK". A good idea: before you close the "\$\$\$" file rename the earlier file in order to avoid "BAK" aches: "Ren Filename.doc Filename.baq" or "... Zilch.doc". If you are inadvertently get out of Spellbinder, you can still save anything you were working on in the RAM "workspace" with the SRecover.com.

Hint: In the batch file for entering/leaving your WP, put a line "break off" toward the beginning and a line "break on" toward the end. That disables C, which may have a different function within the WP (as it does in Spellbinder).

Be wary of answering "y" to "end batch command? y/n".

14. I know how to escape the right margin of the screen by adjusting the wordwrap margin [line length] of my wordprocessor so as to switch (or extend) into another screen, but is there any way to escape the left margin?

No. You can only indent the adjacent lines to give that illusion. But you can get your printer to break the left margin by putting one or more backspace commands at the beginning of the text line. (Begin a Spellbinder line with "!H!H" and it will start printing two places to the left of the margin.)

15. My WP doesn't seem to behave the way the instructions say it should. What am I doing wrong?

Check the configuration file. It is easy to forget that the program has to be told how many logical drives you plan to use and other housekeeping details, as well as your printer type. You might also check the system configuration (CONFIG.sys and, if you made them, Config.bat and Autoexec.bat.) Be sure that the the WP configuration file has a different name (Config.edx, Configsb.com, ...).

16. When I try to open a Write file I get the message "DIRECTORY FULL", yet I have very few files on my disk. What gives?

The message is a cover term for "Directory rejects". You may have offered an "illegal" name (e.g., more than eight letters). Try a different name. According to the documentation, the DOS directory isn't actually full until it has 128 entries (single-sided disk) or 256 (double-sided). But the latter information is in error. If you try to go beyond 128 (including the ".bak"s, of course), your attempt is rejected with the message "FILE CREATION ERROR". The directory is full with 127 entries. By making subdirectories, however, we can get any number of files on a disk, provided there are enough memory segments available to hold the contents. This discussion is about floppies; the hard disk allows you to choose a larger number. In both cases, you can get around any restrictions by setting up subdirectories; that is a good idea, in any event.

17. Can you back-tab on a V9000?

Apparently not, despite the label on the front of the TAB key. (There must be some versions of the Sirius that do that ...?) You can program that or another key to move back eight spaces at a time, but that is not quite the same thing, for the Tab can be set to moduli other than the default eight.

\* \* \*

# FROM THE BOARDS

[FROM THE BOARDS is assembled from messages appearing on various (generally Victor-specific) bulletin boards, mainly The PUB and BAVUG, by John and Mary Jo Knobelsdorf. Only minor editing is done to correct obvious typographical errors and to eliminate some superfluous material. An effort has been made toward a degree of standardized formatting and the elimination of excess space to the extent practical without degrading legibility. Garbles that cannot be readily rectified are allowed to remain. Believing that considerable worthwhile material is contained in these excerpts, our intention is to include this feature in each issue of VictorTalk, as space will allow. - Ed.]

NOTE: The following messages discuss Hard Disks and the setup of hard disks. Much of the discussion concerns 30meg hard disks.

#44 17 29 Dec 86 08:45:47 (RECV'D)  
From: Lloyd Leblanc  
To: Franz Hirner  
Subj: VICTOR HD NOTES

Since our last conversation about Victor drive labels, etc. I have had opportunity to talk to a tech in Scotts Valley about the 30Mb and larger formatting/assignment problems. Here's a nifty one for you: The AUTOSET file shipped with even the latest version of DOS 2.11 (HD) "has a bug" that can not only ruin the FAT etc. on a properly setup drive, but also ERASE THE LABELS. Nice, huh? Even tho the documentation gives you umpteen "internal" and "external" drive combos to choose from in AUTOSET. Advice from the tech department -- use only the DOS 3.1 AUTOSET file. Hmmm. I wonder how many drive label copies Victor has charged people for who have tried to reconfigure their larger drives with the DOS 2 routines. Apparently the HDSETUP files are OK.

#46 15 29 Dec 86 13:16:28 (RECV'D)  
From: Franz Hirner  
To: Lloyd Leblanc  
Subj: AUTOSET

I haven't used autosect in some time. I, for some reason, prefer to set the drives up manually. Perhaps I got in the habit from having autosect clobber the ID - I can't remember. Interesting though. That should explain a lot of 'lost' ID bytes.

#60 13 31 Dec 86 08:46:35  
From: Lloyd Leblanc  
To: Fritz Harner  
Subj: MORE HD QUERIES

I'm very curious about a couple of things now with this new 30 Mb Victor-Tandon drive. First -- Victor's last setup on this (for the previous owner) was under DOS 2.11, so it's got the 12-bit FAT structure. This is OK for a while since we're not yet fully converted to DOS 3.1 (mainly to avoid conflicts with some favorite files). But what if we want to go to DOS 3? What's involved here -- the manual is of little help. I would think at least a new "init": routine and then a config would be necessary, huh? Or is another base format required? I'm confused about how DOS actually sets up and reads the different FAT sizes, I guess.

Also, curiously, with the 30Mb back from Victor and reading (and booting) fine under 2.11, I now CANNOT get any of (3) versions of HDSETUP.EXE to access the disk. Error message says "unable to read drive label" or some such. Sounds funny. Any thoughts? Possible we're running older boot ROMS on this last acquired machine (I was told they were F3-F7, tho)? Would that restrict HDSETUP access? The DOS 2.11 setup file is straight from the (latest) HD DOS diskettes sold by Victor last year.

Makes me more than a little nervous not to be able to get in and manually assign and configure -- esp with those comments from Victor people about buggy AUTOSET files.

#62 15 31 Dec 86 20:16:38 (RECV'D)  
From: Franz Hirner  
To: Lloyd Leblanc  
Subj: 30 Meg, Tandon

If you can't read the ID Byte I'd guess that you have either software or ROM error. F3-F7 ROMS should be good I think. The large hard disks do take special software to set-up. I'm not sure of the version at the moment, but I have no problems with the 30 Meg I've been putting into my V9's lately. Pull the cover and check the ROMs and let's see what you have there.

#70 12 01 Jan 87 06:53:37 (RECV'D)  
From: Franz Hirner  
To: Lloyd Leblanc  
Subj: Hard Disk

The low level software system just writes the ID Sector which tells the controller the configuration of the drive itself, that is how many clusters, cluster size, number of tracks, etc. The low level format forces NULL data bytes onto the disk but does nothing more. Setup software needs to be able to access the entire disk, though and the Victor software has problems with the large hard disk drives. Apparently, initially it was thought that the 10 Meg drives were as large as anyone would want and no provisions were

made for the larger drives. You have to write your own software.

The HDSETUP actually partitions the drives into up to 16 logical drives. Each logical drive has it's own DOS (or CPM) partition and File Allocation Table (FAT). The FAT is fixed by the version of HDSETUP you use and is constant for each logical drive. This explains why you can use smaller minimum file size with smaller disk partitions.

The IBM only allows four logical drives. This is usually enough unless you have hard disks over 120 MEG, but until DOS 3 you could not use 2K cluster sizes. Ideally, you will set up one logical drive with a small partition and small cluster size for your BATCH files, small utility programs, etc. This usually results in an increase in effective drive capacity.

#79 12 02 Jan 87 13:11:11 (RECV'D)  
From: Lloyd Leblanc  
To: Franz Hirner  
Subj: HD INFO THANKS

Hmmmm, well, I THINK I understand what you've stated about the differences in disk sizing and FATs. But if there are "software problems" associated with the larger disks (30 Mb and larger), how does VICTOR set them up? Someone did SOMETHING down there at Scotts Valley after setting the drive label -- to give me the one big DOS "partition" of 29.5 Mb.....if they are using the distributed DOS 2.11 HDSETUP file, why wouldn't it work for me too? I'll look at those ROMs too.

By the way, for any WORDPERFECT/V9000 fans here: I've tested the new WP 4.2 IBM version and it runs perfectly on the Victor under the "emulator". This is good if we don't get a new version this time, or if you want to use "windows" or other features not available on the last V9000 versions. However, though 4.2 has some good new features, it also has a few quirks, mainly in screen/cursor addressing.

#119 10 05 Jan 87 17:53:20 (RECV'D)  
From: Clint Chaplin  
To: Lloyd Leblanc  
Subj: HDisk

Sounds like you have a version of HDSETUP which does not support the two sector bad track list, which was put on your 30M by Victor. You need the latest version. Also, Victor DOS 3.1 for the V9 uses 12 bit FATs for everything. You should have to only transfer the system over (use the DOS 2.11 SYS; 3.1 has little buggies). Then everything should work without reformatting the disk.

#123 9 05 Jan 87 21:58:45  
From: Franz Hirner  
To: Clint Chaplin  
Subj: HDisk

2.15 SYS should work also on Lloyd's hard disk.

#130 4 06 Jan 87 17:18:42  
From: Franz Hirner  
To: Lloyd Leblanc  
Subj: AU's

AU's can only be changed to a point. It depends on the size of the partition that you have set up and also the size of the FAT. The FAT only allows so many entries so there is no point in setting up too small an AU if the limit will then be the FAT size. Sometimes it is best to set up a separate, small directory with small AU's to contain .BAT files and other small programs and utilities.

NOTE: The following subject is 2400 and 1200 baud modems

#53 14 24 May 87 15:04:29  
From: Boyd Naron  
To: All  
Subj: 2400 BAUD MODEM SALE

If you are tired of waiting for files to transfer at 300 or 1200 baud, now you can upgrade to 2400 baud and not break your budget.

Hayes Compatible (external) 2400 baud modem  
full AT command set  
full stack register manipulation  
full complement of modem lights  
\$199.00 + \$7.00 Shipping and Handling  
Send check or money order to:  
TCL Subsystems  
500 'N' Street #705  
Sacramento, CA 95814 (916) 448-3836  
Allow 2 to 3 weeks for delivery...  
CALIFORNIA RESIDENTS ADD \$11.94 SALES TAX

#57 16 25 May 87 06:40:31  
From: John Navas  
To: All

Subj: Inexpensive 2400 bps modems

The USR 2400 bps internal (for the IBM PC and compatibles) is widely available by mail order for \$189 (or sometimes even less). Most users seem to be quite satisfied.



#61 14 25 May 87 13:18:07 (RECV'D)  
From: Franz Hirner  
To: John Navas  
Subj: 2400 bps modems

Omnitel also makes a solid 2400 internal 2400 baud modem that is priced below \$200.00. It is compatible with Hayes and will run Fido without a hitch. Fido is very particular about compatibility.

The USR is also very solid. I use a external USR 2400 on BAVUG and am quite happy with it.

#65 14 26 May 87 07:24:48 (RECV'D)  
From: Lloyd Leblanc  
To: Franz Hirner  
Subj: MODEMS

You might add to the 2400 baud modem list the ZIPPER 1200/2400 sold by Priority Electronics for just under \$300 complete: this is their version of the Prometheus 1200/2400 modem that receives pretty good reviews. As far as I can tell the ZIPPER is identical, though quite a bit less costly. Seems totally Hayes compat to me and there are some other very useful functions available, plus options. After six months' use of one, I would recommend.

#66 16 26 May 87 15:20:16 (RECV'D)  
From: Franz Hirner  
To: Lloyd Leblanc  
Subj: Prometheus/ZIPPER

I didn't know that they were one and the same. With all the options the Prometheus becomes a stand-alone system.

I remember when I paid \$495.00 for my Hayes 1200 and that was a "bargain". Incidentally, the AVATEK (spelling??) is a nice 1200 baud modem for about \$75.00 depending on the source. There's really no reason for anyone to not hit the "boards" anymore.

#74 13 23 Jul 87 16:41:03 (RECV'D)  
From: Franz Hirner  
To: Bert Happel  
Subj: AVATEX 1200

I played with an Avatex 1200 for a while. The local group was thinking of offering a package to members complete with cables, documentation, batch files, Vicmini, etc.

I can't remember using the Avatex for mail, but it worked fine in place of my Hayes 1200 for Vicmini, Xmodem and Crosstalk. I did run Fido off the Avatex, but only for regular calls - not mail.

When you borrow the Hayes, you might be able to set two machines up with your own one line nodelist and have "local" mail until you get the kinks worked out.

#23 13 29 Aug 87 20:35:26 (RECV'D)  
From: Franz Hirner  
To: Dan Mardesich  
Subj: Zoom Modems

There aren't too many 2400 modems that are clean. Omni, U. S. Robotics and Hayes seem to be about the best. Byte did a review of modems six or eight months ago. Interesting enough, the cheap 2400's generally outperformed the expensive 1200 baud modem's when running at 1200 baud but there was a major drop off in reliability of the cheap modems at 2400 baud. Perhaps there have been some changes since then.

To my way of thinking, a modem should be external and the best you can afford. You can then use it with whatever computer you may acquire in the years ahead.

NOTE: The following subject is about making a public domain program that would duplicate Victor Technology's FACTORYF and SUP\_BIOS.211

#46 16 24 May 87 07:38:46 (RECV'D)  
From: Franz Hirner  
To: John Navas  
Subj: Victor Hard Disk

Victor is still shrinking. They're down to 39 employees now. The new import duty on Japanese machines has just about knocked them out of the clone business.

Victor has never released their hard disk set up program FACTORYF. This is the program that creates the Disk ID Label which HDFORMAT and other programs in the HD Tool Kit use. Most of the data is contained in DISK\_DAT in Area 7. Victor uses their own SUpEr-Bios routines which are also documented in a file in Area 7. How much of a job would it be to write a program to create the disk label that could be released to the PD?

#48 14 24 May 87 12:11:49 (RECV'D)  
From: John Navas  
To: Franz Hirner  
Subj: Victor FACTORYF  
Re:

Victor has never released their hard disk set up program FACTORYF. This is the program that creates the Disk ID Label which HDFORMAT and other programs in the HD Tool Kit use. Most of the data is contained in DISK\_DAT

in Area 7. Victor uses their own Super-Bios routines which are also documented in a file in Area 7. How much of a job would it be to write a program to create the disk label that could be released to the PD?

Pretty easy I'd guess! I converted VIC\_DISK.DAT to C and uploaded it as VIC\_HD.ARC. I tried to check out SUP\_BIOS.211, the file has a LOT of non-ASCII garbage and looks corrupted. Nevertheless, if there are well documented interrupts to read and write absolute disk sectors, it should be a piece of cake to use the int86x() function in MSC to tinker with the disk any way that you want!

#50 13 24 May 87 12:33:27 (RECV'D)  
From: Franz Hirner  
To: John Navas  
Subj: SUP\_BIOS.211

Thanks for letting me know that the file is corrupt. I'll take a look at it.

Victor does everything through INT 223 (Super-Bios) including a lot of their diagnostic's, initialization, etc.

I've never "tweaked" a disk directly with anything I've written. I should set up a test machine to "crash" for the good of science.

#51 14 24 May 87 16:13:06 (RECV'D)  
From: Franz Hirner  
To: John Navas  
Subj: SUP\_BIOS

I just finished entering the previous message to you when the power supply went out. One new power supply and a full back-up of the files (which was overdue) and we're back on line.

I put a fresh copy of SUP\_BIOS.211 in the Victor Tech Area. I don't know if the power supply had anything to do with the corrupted file or not. It's been downloaded several times before and no one has commented that it was corrupted (which it was). This doesn't tell you much in the way of text, but I thought with your background with hard disks that it might make sense to you.

#68 17 27 May 87 11:52:20  
From: Franz Hirner  
To: Clint Chaplin  
Subj: Super-Bios

Is there a description of Victor's Super-Bios routines for the hard disk operating systems?

NOTE: The following topics are discussed:  
Downloading files and archive files  
PC Pursuit and downloading  
CrossTalk updates  
Communication software  
MultiPlan, Lotus 123 and the 8087 math chip.

#19 13 29 Aug 87 16:07:54 (RECV'D)  
From: Dan Mardesich  
To: Franz Hirner  
Subj: ARC Files

I have down-loaded two .ARC files from the dBase area then tried to Un-ARC them with PKXARC.COM from PD disk 83. I get the message that they are not archived files. Is there another ARC routine I should be using? Thanks for the help.

#21 15 29 Aug 87 20:16:24 (RECV'D)  
From: Franz Hirner  
To: Dan Mardesich  
Subj: Misc.

Looks like you are moving right along. Most of the files on the board should be ARC'ed with ARC521 which you downloaded. Sometimes a file may be uploaded that was ARC'ed with PKARC and ARC ARC may not work with some PKARC'ed files. PKARC has a option that is not ARC compatible and if that option is used there will be a problem. I believe both the ARC and PKARC files here are the latest. There is also a file that tells you how to modify PKARC so that compatibility with ARC is the default. You might want to download that one also.

I've heard of IBM clones but never a COMPAQ clone. That's a clone of a clone. That's going the wrong way from a Victor.

#43 19 01 Sep 87 19:31:06 (RECV'D)  
From: Dan Mardesich  
To: Franz Hirner  
Subj: ARC512

I found ARC512 after I left my message, downloaded it and it does work on the dBase files I had downloaded. Thanks for the follow-up.

From: John Navas  
To: Roger Folsom  
Subj: PC Pursuit and downloading

Your problem is not your modem, but the constraints of PC Pursuit when using a file transfer protocol. To get really good results you need a newer advanced protocol such as

YMODEM, ZMODEM, WXMODEM, or Sealink. Unfortunately, this board currently supports none of them so you have to limp along with XMODEM. If you are patient -- after all, the price is right -- you can get by with comm programs that allow for the long time delays that occur in PCP with XMODEM. Otherwise, browse with PCP and use AT&T for your XMODEM downloads.

#42 16 01 Sep 87 17:11:49

From: Ken Fiske

To: Sysop

Subj: Crosstalk update

If you operate Crosstalk XVI, they will update it by converting it to Crosstalk Mk.4 which has lots of extra goodies. Price is return of the original registered XVI disk and \$50.00 plus handling. It is available thru Crosstalk Communications, Crosstalk Mk.4 Conversions, 1000 Holcolmb Woods Parkway, Suite 440, Roswell, Georgia 30076. Can't guarantee it is in 9000 format yet unless enuf people want it.

#44 14 02 Sep 87 04:40:18 (RECV'D)

From: Roger Folsom

To: Lloyd Leblanc

Subj: Kermit

No, I still have not made Kermit work. Problem is simple: after I have connected, then if I use an escape sequence to return to local control (i.e. to return to MS Kermit prompt), I get the MS Kermit prompt as I am supposed to, but the phone hangs up. I've about decided that my Taiwan cheapie modem is doing me in--although it works fine with Vicmini, and in fact I have patched Vicmini so that it uses the same escape right bracket to toggle to and from local control that Kermit uses (that way I can use the same keyboard for both) I have NOT modified Kermit in any way, and the problem occurs regardless of whether I use escape right bracket (ordinarily not accessible on the original Victor keyboard, but it is accessible on my telecomm.kb), or use some substitute such as a tilde or anything else.

On another, but perhaps related, subject: My modem does fine with ASCII captures, but it is doing badly in true downloading using either Kermit protocol (via Vicmini) or Xmodem with either Checksum or CRC--but I tried direct connection the other night to BAVUG instead of going through PC Pursuit, and downloading worked fine. So maybe I need a better quality modem to work with PC Pursuit.

If you have any ideas, before I borrow or buy a better modem, I'd like to know them. Thanks for the ideas you have already offered--they have been appreciated.

#45 16 02 Sep 87 04:49:09 (RECV'D)

From: Roger Folsom

To: John Navas

Subj: Downloading

Thanks for the encouragement of my penurious (PCPursuit and Taiwan cheapie modem) ways. You warm the cockles of my economist heart. Maybe the next version of FIDO, which I think I saw is about to come out, will add more sophisticated protocols. Note, however, that my downloads didn't get garbaged in midstream--they never even started. That is, the first block never made it. So I do wonder whether the modem is inadequate, somehow. Or maybe I did things right when I tried a direct connection, and messed up using PCPursuit--but I don't think so, I think I used the same (correct) procedures in each case. Pride goes before a fall--i.e. a crash. See my reply to Sam Martin's Message #58 (Is there same way to link your message and my reply to his message to me and my reply to him?)

#46 18 02 Sep 87 04:56:35 (RECV'D)

From: Roger Folsom

To: Ken Fiske

Subj: Crosstalk latest version XVI

Does the latest Crosstalk support Kermit, or the other protocols mentioned by John Navas in his message to me? I think I will give them a call and indicate my interest in a Victor 9000 version, in case the answer to these questions is yes.

#47 15 02 Sep 87 05:43:31 (RECV'D)

From: Roger Folsom

To: Ken Fiske

Subj: Crosstalk XVI

Microstuff, developer of Crosstalk, has 800 number, 241-6393. Crosstalk XVI now DOES support kermit. \$195 list. Crosstalk Mark IV, which despite having a lower roman numeral is a superset of XVI, lets you deal with 15 terminals (e.g. 15 bulletin boards, or a bull board plus Dow Jones plus Source plus . . . etc.) simultaneously, using a single modem, and an x. (dot) PC protocol supported by tymnet. But Mark IV is not available on the Victor. Cost is \$245. However, Crosstalk XVI does support Victor, still, and now is at version 3.61, in Victor format. Presumably, it takes care of the problems that the various patches for version 3.5 dealt with, as well as supporting Kermit.

#48 13 02 Sep 87 07:08:52 (RECV'D)

From: John Navas

To: Roger Folsom

Subj: File transfers over PC Pursuit

Before you blame your modem, consider that the problem could well be your comm software. In fact, since your modem works OK over direct long distance connections, I'd say it's probably OK.

Because of the nature of packet switching, it's \*very\* difficult to transfer files over PCP using an old protocol like XMODEM. (And yes, the symptom is that you often cannot even get the transfer started.) Some comm programs (like ProComm 2.4.2) really have problems with PCP, even if "relaxed" XMODEM timing is used. So try another program -- and remember that transfers will be very slow.

As I said, the secret to fast file transfers over PCP is to use more advanced protocols. YMODEM is much better than XMODEM, but the real winners are ZMODEM, WXMODEM and Sealink. Unfortunately, Fido currently doesn't support any of them, so you'll have to find both a comm program and a board that supports one of these more advanced protocols to try it out.

For some good insights into the special problems of PC Pursuit, read the messages in Message area 7 on PDSE San Mateo (415-571-6911).

#49 11 02 Sep 87 12:49:02 (RECV'D)  
From: Franz Hirner  
To: Roger Folsom  
Subj: Crosstalk 3.61

Did Microstuff indicate what changes have been made to Crosstalk XVI 3.61 for the Victor?

When downloading with Crosstalk try plain XMODEM rather than XMODEM CRC. You might try Vicmini available from the file area. It supports xmodem, xmodem CRC, telink, kermi, etc. All it won't do is auto dial, but the ARC file contains ways to get around that also.

#50 10 02 Sep 87 12:54:48  
From: Franz Hirner  
To: Dan Mardesich  
Subj: ARC512

Glad everything worked out OK.

#52 12 02 Sep 87 12:57:20  
From: Franz Hirner  
To: Roger Folsom  
Subj: Downloads

Your modem shouldn't make too much difference at 1200 baud or lower. I've started some callers with \$5.00 300 baud modems from the electronics surplus store at the Oakland Airport with no problems. What software are you running? You have been referring to Kermit but are you

using any other software also? Try Vicmini, available here or on the P. D. Disks. It works well although it does not support auto-dial. At least if your modem works with Vicmini you can figure your problem is due to your other software, parameters, or whatever.

#58 11 02 Sep 87 19:27:30  
From: Ken Fiske  
To: Roger Folsom  
Subj: Kermit

Info booklet says yte,,yes it does support Kermit, X.PC, Xmodem and Crosstalk protocols.

#63 12 03 Sep 87 07:32:35 (RECV'D)  
From: John Navas  
To: Franz Hirner  
Subj: MP and the 8087

A friend of mine has extensively tested MP v3 both with and without an 8087 installed. It does make a difference, but it's not huge -- on the order of 20% or so for a common business spreadsheet. Of course the advantage would tend to be greater if complex math functions are being used.

The difference may be greater for Lotus -- which wouldn't be surprising, since MP is significantly more efficient (recalculates faster) than Lotus without an 8087.

#49 13 03 Sep 87 07:32:35 (RECV'D)  
From: John Navas  
To: Franz Hirner  
Subj: MP and the 8087

A friend of mine has extensively tested MP v3 both with and without an 8087 installed. It does make a difference, but it's not huge -- on the order of 20% or so for a common business spreadsheet. Of course the advantage would tend to be greater if complex math functions are being used.

The difference may be greater for Lotus -- which wouldn't be surprising, since MP is significantly more efficient (recalculates faster) than Lotus without an 8087.

#50 11 03 Sep 87 15:38:07 (RECV'D)  
From: Franz Hirner  
To: John Navas  
Subj: LOTUS/8087

I have not done any accurate testing, but I'd estimate that a large moving average type spreadsheet (12 by 500) will calculate three or more times faster with an 8087 than without.

CAD/CAM type applications are on the order of 8 to 10 times faster on a screen redraw.

#107 7 15 Sep 87 16:36:05

From: Bert Happel  
To: All  
Subj: MP Templates?

Anyone know of a BBS with a collection of MultiPlan templates? I recently got v3.0 on an upgrade and would like a jumpstart on using it.

#111 6 16 Sep 87 07:20:18

From: John Navas  
To: Bert Happel  
Subj: MP templates

Get a copy of the new edition of "Managing Your Business with Multiplan" (Microsoft Press) -- there are a lot of good examples, and a disk is available if you don't want to key them all in. Best book I know of on Multiplan!

In the meantime, I'll upload my CHECKREG.ARC -- which is a simple checkbook register that demonstrates macros and customizable menus.

#117 2 10 Sep 87 09:09:47

From: Franz Hirner  
To: Lloyd Leblanc  
Subj: Messages

Thanks for your comments on the message arrangements. There are pluses and minuses no matter what you do.

When BAVUG was run from a 5 Meg corner of my machine at work I tried to have specific file areas for different subjects. Many callers would leave a message in the proper area, but come back and look for a reply in another area. Others entered messages in the wrong area. This was before echos were added.

Until message traffic reaches very high volume levels I think that just one local General Message Area has some advantages. The Echos, of course, are a different matter.

The shape of the board is really directed by the callers. Note the CP/M file area. Two callers asked for it and it was set up, but nothing ever came of it afterwards.

I would like to keep BAVUG somewhat loose to the extent possible. Some boards will not allow 300 baud modems or require minimum upload/download ratios be maintained for example. I'd rather see the 300 baud modems allowed access. Let's face it, there are a lot of people out there who can be enticed into buying a surplus 300 baud modem for \$10 or \$15. If they like what they see, they'll upgrade soon enough to 1200 or even 2400 baud. As far as upload ratios, there are many beginning users who don't really have anything yet

that is of general value anyway. Yet, they can not only download useful programs and utilities but learn by studying the dBASE, BASIC, "C" or whatever code. Eventually they'll come across, either write or find a program or utility worth uploading.

Meantime, any and all callers are welcome to make any suggestions regarding the content and the structure of BAVUG. As long as Victor Users have a place to hang their hats and gain some benefit, either directly or indirectly, I feel we've contributed something.

As time goes on, Victor users are going to have to get more into programming if they want to expand the value of their hardware. Let's face it, software is written for the machine of the day. That's where the market is. Unfortunately, the machine of the day is not Victor, IMSAI, SOL or many other fine machines that have faded into history. That doesn't mean that those machines are obsolete. They'll still do anything the newer machines will, if the software is written. I'm happy if BAVUG contributes in some small way to making that possible.

#119 2 10 Sep 87 13:50:47

From: Franz Hirner  
To: Anyone  
Subj: Ray Duncan

Ray Duncan will soon have a regular column in PC Magazine every month on assembly and "C" programming. Even Victor owners would be well advised to at least purchase a few editions of PCM to see how the articles look. Ray Duncan is one of the better known authors of programming articles for MS-DOS and PC-DOS. He had a regular column in Softalk (an excellent magazine in the early days of the PC but now defunct) and is the author of "Advanced MS-DOS". If you are at all interested in programming, even just a ten line printer control routine, you might find his articles very interesting.

I'm not sure when his column will begin but I'd guess in the next one or two issues. If the content is the same quality as his series was in Softalk I'd think you'd want to save them for future reference.

\* \* \*

---

#### VICTOR COMPUTERS AND REPAIRS

David Koshinz and Yorgos Hatzis (both formerly of Victor Technologies, Inc.) are starting their own computer store. They offer, among other things, used and reconditioned Victor 9000 and VI computers, and general repairs and services. If you have a problem, give them a call: (408) 336-3240, 190 Coon Heights Road, Ben Lomond, CA 95005.

\* \* \*

# THE LIBRARY REPORT / John Knobelsdorf

Two new public domain disks, PD #91 and #92, have been released. Copies are now on file in the Chapter set which is maintained by John Kimber, and will be available for copying at the monthly meetings. They also may be ordered thru the mail in the usual manner.

NOTE: Some files on these disks are in compressed form.

Use NJSQ (on PD #73) to "Unsqueeze"  
Use LUE " " to "Unlibrary"  
Use PKXARC (on PD #87) to "Unarchive"

The contents of disk PD #91 are:

DISKID91.TXT 4096 Information about the files on this disk PD #91.

Note: The following files on this disk should work O.K. on the V9000 in standard format, and without using an emulator.

FINDS101	ARC	16512	Find strings in files; V9 OK.
KEYFONT	ARC	11264	Helpful notes on KEYGEN, EFONT and MODCON
SHOWCHAR	ARC	10240	Enter 0 - 225; see character in current charset
COUNTEM	ARC	12672	Several word/line counting programs together
COMPARE	ARC	34560	Utilities comparing lines of similar files

Note: The following files on this disk will require the use of a Victor VI, Plus-PC, a PC-Compatible or one of the various emulators. Try the files and programs, and let us know what works and what doesn't work. We gather files like this for your potential benefit. Help us do a better job by advising of your experiences.

QMODEM31	ARC	142336	QMODEM SST Ver.3.1; the "BMW" of comm packages
TELIX212	ARC	112640	Great comm package; fast, slim, powerful!
TELXSORT	ARC	15360	Telnet .FON directory sort utility; Ver. 2.0
NYWORD22	ARC	208896	Marc Adler's terrific shareware wordprocessor
NYWRDPRT	ARC	43008	New York Word printer configuration files

605568 Bytes in 10 Files. 10240 Bytes free.

The contents of disk PD #92 are:

DISKID92.TXT 4992 Information about the files on this disk PD #92.

Note: The following files on this disk should work O.K. on the V9000 in standard format, and without using an emulator.

ANYWHERE	.ARC	21504	Filesearch designated drives in specifiable order
BMGREP	.ARC	27917	Boyan-Moore string search program.
CMPTRUSE	.RNF	4924	Comparison of alternative tax log programs
CONTROL	.ARC	6552	Sam helps us use both CNTL & ALT key functions

\*\*\*\*\*

## SPECIAL NOTE:

All of the files and programs on these disks were downloaded from the PUB, a private user's board in Boston, owned and controlled by Michael Wishnietsky and operated by Marty Moleski, the Sysop. Our special thanks to Michael and Marty and all of the members and users of the PUB who contribute so generously of their time, talents and telephone expense.

\*\*\*\*\*

The contents of the DISKID files for these disks follow.

CRACKIBM.ARC	5782	How to decrypt protection coding etc. (text).
DELDUP22.ARC	16384	Buerg's utility to delete duplicate files; w/ASM
DOSINTRO.ARC	31753	Mary Jo Knobelsdorf's 3-part intro to MS-DOS 2.11
FASTLAST.EXE	11392	Display n lines from end of file (or redirect).
FGREP158.ARC	9472	Chris Dunford's [FASTEST] GREP utility
FUNDS .ARC	87936	BASIC/GRAFIX fund tracking program; w/source
KTG-MOD .ARC	43233	Jim Hastey's printer fix for Knee Toe Graphics
K7/IC .ARC	21504	Finds keyword in context (size specifiable).
MGREP .ARC	14427	Searches text strings; works when FGREP won't.
MODEM .ARC	29184	C.H. Carlisle on Modem Madness; interesting stuff
PARK .ARC	3712	Guy Gordon's PARK utilities; w/source (by request)
PKARCBUG.ARC	1408	Report from Pkware of Pkarc Bug and Patch
SCAN2 .ARC	6144	Displays ASCII text within binary files; useful.
SORTF225.ARC	6400	Vern Buerg's latest fast sort; feature-rich
TT .COM	8192	Gives Xmodem transfer time for file(s); V9 OK.
WPMACS11.ARC	10240	WordPerfect macros; worth the download time?

Note: The following files on this disk will require the use of a Victor VI, Plus-PC, a PC-Compatible or one of the various emulators. Try the files and programs, and let us know what works and what doesn't work. We gather files like this for your potential benefit. Help us do a better job by advising of your experiences.

WSSI320A.ARC 109824 Part 1 of Babcock's newest disk index/catalog utility.  
WSSI320B.ARC 107648 Part 2 of Babcock's newest disk index/catalog utility.

591104 Bytes in 23 Files. 10240 Bytes free.

\* \* \*

#### FURTHER VICTOR TECHNOLOGIES SALE

The following remaining items are available at \$10.00 each from Rosemary Abell of Victor Technologies, Inc.:

VI KEYBOARD, International (German, French, Swedish)  
VI FLOPPY DRIVE  
VICI KEYBOARD, French

Phone orders preferred, with payment by VISA or MASTERCARD; call Rosemary at (408) 426-6194; after December 18, leave message at the warehouse (408) 263-4777.

If ordering by mail, address:

Rosemary Abell  
897 Wrigley Way  
Milpitas, CA 95035

\* \* \*

#### VICTOR COMPUTERS FOR SALE

V9000, hard disk (20MB internal) with single DS floppy drive; IBM PC compatible. Model 721 99-key programming keyboard (keys dual labeled for Victor and IBM program users); Victor Model 602 high resolution monochrome display.

V9000, 128 KB memory, two single or double sided floppy disk drives; 97-key word-processing keyboard; Model 602 display; CP/M-86 and MS-DOS operating systems. Software includes WordStar 2000, Lotus, Multiplan, BOOKS! (book-keeping system) and more.

For further information contact:

Jim Logan  
Logan Productions, Inc.  
8035 N. Port Washington Road  
Milwaukee, Wisconsin 53217  
(414) 352-9691

\* \* \*

## MORE V9000 COMPUTERS FOR SALE

V9000 with single-sided floppies, 256K RAM, and old ROMs.

George Marianes, (415) 865-3042  
3417 Soloman Ln.,  
Alameda, CA 94501

V9000 with 10MB hard disk, 256K RAM, 10 function key keyboard, Wordstar, Multiplan, Real World (bookkeeping).

Hari Bachan, (415) 841-0571  
1640 Fransisco St.,  
Berkeley, CA 94703

V9000 with 512K RAM, old keyboard, old ROMs, Lotus 1-2-3. \$250.

David Wheeler (415) 488-9781  
7 Crescent Dr.  
Woodacre, CA 94973

V9000 with 640K RAM (CPI expansion board), 2 single-sided drives, WP keyboard (7 function keys, plus HELP), system dust-cover. Software includes: CPM/86 with manual; MS-DOS 1.25 and 2.11, including MS-BASIC, with manuals; Magic Memory; White Crane Systems RAM Drive; Knowledge Manager (aka K-man, KnowledgeMan) 1.07 with manual and textbook; Spellbinder 5.4 with manual; Crosstalk XVI, version 3.5 with manual; Lotus 1A with manual; complete set of VictorTalk, Vol. 1, No. 1 through Vol. 5, No. 5; Victor Technologies Hardware & Software Catalog; miscellaneous articles on the above. Asking \$800; will consider any reasonable offer.

Kenneth P. Ong  
276A Connecticut St.,  
San Francisco, CA 94107  
(415) 953-4772 (work)  
(415) 626-1970 (home, before 10PM)

\* \* \*

## VictorTalk

Address inquiries to: P. O. Box 50893  
Palo Alto, CA 94303

Subscriptions: Nancy Weaver

Editor: Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

Contributors: Al Bullock, John Knobelsdorf, Mary Jo Knobelsdorf, Sam Martin, Jack Prichard, Nancy Weaver.

VictorTalk is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in VictorTalk. Copyright 1987 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

VictorTalk is an independent publication not affiliated in any way with Victor Technologies, Inc. Victor is a registered trademark of Victor Technologies, Inc.

Submit Articles on MSDOS disks in ASCII or Spellbinder format, with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

## Yes, I want to join.

Enclosed is a check/money order for \$35.00  
(foreign membership, \$40.00 in U.S. dollars)  
for my 1988 Victor\*Group membership.

Name \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

Make checks payable to:

## VictorTalk

P. O. Box 50893  
Palo Alto, CA 94303

City \_\_\_\_\_

State \_\_\_\_\_ ZIP \_\_\_\_\_

Country \_\_\_\_\_

Phone \_\_\_\_\_



PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
P.O. Box 50893  
Palo Alto, CA 94303

Date \_\_\_\_\_ Office use only:  
 Name \_\_\_\_\_ Order rec'd \_\_\_\_\_  
 Company \_\_\_\_\_ Member check \_\_\_\_\_  
 Address \_\_\_\_\_ Comments \_\_\_\_\_  
 City \_\_\_\_\_ State & Zip \_\_\_\_\_  
 Country \_\_\_\_\_ Phone \_\_\_\_\_ Order shipped \_\_\_\_\_

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No. Disks	Vol. No. Disks	Vol. No. Disks	Vol. No. Disks
4 2 61-64	4 6 72-77	5 2 82-84	5 5 87-90
4 3 65-71	5 1 78-81	5 4 85-86	5 6 91-92

- 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
 21 22 23 24 25 26 27 28 29 30 31 32 33 34\* 35\* 36 37 38 39 40  
 \* (34 & 35 are a two-disk set)  
 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56\* 57\* -- 59 60  
 \* (56 & 57 are a two-disk set)  
 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80  
 81 82 83 84 85 86 87 88 89 90 91 92 \_ \_ \_ \_ \_

1987 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_  
 Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_  
 Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_  
 Postage/handling (applies to disk orders only)..... \$ 4.00  
 Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_  
 VictorTalk back issues: 1986, 1987, sets of 6, \$12.00 /set ..... \$ \_\_\_\_\_  
 1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_  
 1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_  
 1988 membership fee \$35 (receive all 1988 issues of VictorTalk) .. \$ \_\_\_\_\_  
 (Foreign membership, \$40 in U.S. funds.)  
 TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_



---

**VICTOR\*GROUP**

The San Francisco Bay Area Victor Users Group

---

1 9 8 8

**MEMBERSHIP RENEWAL NOTICE**

This issue of VictorTalk, which contains our first cumulative index, is being mailed to all who were members of Victor\*Group in 1986 or 1987. We hope you find it helpful in locating past VictorTalk articles.

If you haven't already sent in your annual membership fee for 1988, please do so NOW! To continue receiving VictorTalk and the other benefits of membership, simply tear off this cover sheet (with your mailing label on the reverse side), make any necessary address corrections, and return to us (in an envelope) with your enclosed payment.

Additionally, we would be delighted to receive:

- o Your comments on how we are doing,
- o Suggestions for future articles, and,
- o Especially, your help in providing material of general interest for inclusion in future issues of VictorTalk. (For example, see "Help Wanted", VictorTalk, Vol. 5, No. 6, Nov-Dec 1987, page 11. Articles on subjects other than Public Domain software are, of course, equally desired.)

---

January 1988 through December 1988  
Annual Membership Fee for the San Francisco  
Bay Area Victor\*Group:

**Please Remit To:**

VICTOR\*GROUP  
PO Box 50893  
Palo Alto CA 94303

U.S. Residents: \$35.00

All Other Countries: \$40.00

U.S. Dollars Only

Membership includes all issues of VictorTalk published during this membership period...Access by mail to our Software Library at 1/2 price... Group purchase discounts offered through Affiliated Computer Groups, Inc...Participation in our Bay Area monthly meetings to exchange ideas and information...Access to BAVUG (Bay Area Victor Users Group) Bulletin Board...

---

Is mailing label on reverse side correct?      No      Yes

Please tear off this page and return it with your check to ensure that we maintain the proper listing for your membership.

# VICTOR<sup>®</sup> TALK

Bi-Monthly Magazine for Victor 9000 users

VictorTalk, Vol. 6, No. 1, Jan-Feb 1988

## VICTORTALK CUMMULATIVE INDEX

### Volumes 1-5, 1983-1987

This issue of VictorTalk marks the start of the sixth volume, an appropriate time for the publication of a cumulative index. Hence, we are happy to present this index, which consists of three tables: Table 1, the chronological listing of each published article with its category or categories; Table 2, a cross-index by category; and Table 3, a cross-index by author.

In each table, articles are referenced by a four-digit number (V/N/P) giving the volume, number, and page (as the 1st, 2nd, and 3rd and 4th digits, respectively). (In keeping with the four-digit format, zero is used for the third digit with pages 1 through 9. Although this system gives ambiguous references when two or more articles start on the same page, these ambiguities are easily resolved. For ease of reference, categories are listed alphabetically.

In Table 1, supplementary title-like information is given

in square brackets in a few instances. Where no author is shown, the article has either been written by the editor or assembled from (usually unidentified) contributors. In Table 2, the categories and their descriptions are identically those used in "Public Domain Library Index", VictorTalk, Vol. 5, No. 5, Sep-Oct 1987, page 1 (5501).

VictorTalk has been published since September, 1983. Volume 1, Number 1 was identified as "Premier Issue, September, 1983". Succeeding issues were identified by Volume and Number (except Volume 1, Number 2, which was not identified). In most cases, issues were additionally identified by date. Initially, publication was about once a month; since the beginning of 1987, publication has been bimonthly. The number of issues published each year is as follows: 1983, 3 issues; 1984, 9 issues; 1985, 9 issues; 1986, 6 issues; 1987, 6 issues. (For the foreseeable future, it is planned to continue publishing bimonthly.)

TABLE 1. CUMMULATIVE INDEX (Page 1 of 7)

V/N/P	Title / Author	Cat.
1101	Welcome	NEW
1102	Chuck Peddle Talks about Victor	VTI
1104	Chapter News	NEW
1105	Is There Life Beyond WordStar?	WP:WS
1105	The Ease of MemoPlan / Mark R. Majhor	WP
1106	The Many Features of Spellbinder / Steve Barnes	WP:SB
1106	Everything's WordPerfect / Alex Shafer	WP:WP
1107	Making Victor Fly / Jack Skalicky	MEM
1107	132 Column Mode	SCR
1108	Calendar	NEW
1201	The Ball is Rolling / Jack Skalicky	NEW
1201	Extra! Extal	VTI
1202	dBASE II - Some Introductory Thoughts / Douglas Keachie	DB:dB2
1203	TIM [Total Information Management / Debora Levy	DB
1204	Database with a Spreadsheet [KnowledgeMan]	DB SPR
1204	"a Nice Little Package" [Super List Manager]	DB

V/N/P	Title / Author	Cat.
1205	Chapter News	NEW
1206	Typesetting with a Victor 9000 / David Minehart	WP
1206	TEX Typesetting	WP
1207	How Victor Markets Software / Greg Boo	VTI
1208	Calendar	NEW
1301	Now for the Latest News	NEW
1302	Real Estate Applications On the Victor / R. John Kimber	DB:dB2
1303	Questions and Answers	DB:dB2 SPR WP:WS
1304	Operating Systems on the Victor 9000 / James Leshner	DOS
1305	Chapter News	NEW
1306	Announcing... [Vicki and Apricot]	HAR
1306	Loan Amortization Program / Lawrence Akutagawa	BAS FIN
1306	\$100 Upgrade for Your VictorCalc	SPR:VC
1307	VictorTalk Classified Ads	SAL
1308	Calendar	NEW

TABLE 1. CUMMULATIVE INDEX (Page 2 of 7)

V/N/P	Title / Author	Cat.
2101	Happy 1984 to All Victor 9000 Users / R. John Kimber	NEW
2102	The Truth About Floppy Disks / Marie K. Falcon	MEM
2102	Questions and Answers	DOS SPR:123 WP:WS DB:DB2
2104	Tired of Ugly Mailing Labels? / R. John Kimber	NEW
2105	Chapter News	SAL
2106	Where to go to get Victor Software	BAS
2106	Help and Hints Regarding BASIC / Lawrence Akutagawa	NEW
2108	Calendar	NEW
2201	Used or User - One Person's Opinion / Darrell D. Bross	NEW
2202	How to Buy a Modem / Mark Majhor	COM
2202	Victor Plus GMBASIC Produces Muscial "Carnival of Venice" / Andrew Rudiak	BAS MUS
2203	Questions and Answers	COM:XMO CPM HAR CPM DOS
2204	MS-DOS and CP/M Command Comparison	WP:WS DOS
2206	New WordStar Enhancement	NEW
2206	When to Use DCOPY and COPY *.* / Lawrence Akutagawa	NEW
2207	Chapter News	NEW
2208	Calendar	NEW
2301	The Amazing Victor Story	NEW
2301	The British Are Coming? [ACT]	NEW
2302	Victor Products Becoming Available	VTI
2302	Some Thoughts on a National User Group	NEW
2303	Questions and Answers	DIR DOS MEM SPR:VC VTI
2304	Victor's "Vicki" - The Latest News from Germany / Michael Stuhr	DB
2305	An Alternative Data Base System - MAG/base / Joe Hawk	LST
2306	Victor User Groups	NEW
2307	[Chapter News]	COM
2310	VICTMINI - Communications Software for the Victor / David Minehart	

V/N/P	Title / Author	Cat.
2310	Setting Up Your Own BBS System	COM
2311	The Need to Standardize Formats / David Minehart	BAS WP
2312	Display Driver Commands in Victor's MSBASIC / Bob Marcil	BAS SCR
2314	Setting Up Batch Files / Ron Trenowski	BAT
2315	Resource 9000 Stops Publishing	NEW
2315	Public Domain Library Growing / Darrell D. Bross	PD
2316	Calendar	NEW
2401	Victor News Includes ACT and Peddle	VTI
2402	Questions and Answers	DIR DOS MEM SPR:VC
2403	IBM Scrambles to Become Victor Compatible / Joe Hawk	HLM
2404	12 Public Doamain Disks Available for the Victor / Darrell Bross	PD
2404	Concurrent CP/M to Be Announced Soon	CPM
2406	From VictorCalc to Report Manager 2.0 - A User's Critique / Richard Jones	SPR SPR:VC
2410	Three Statistical Packages for the Victor 9000/Sirius Computer / Chris D'Elia	MAT
2412	Victor Users Group	LST
2413	[Chapter News]	NEW
2414	Making Music on the Victor / Andrew Rudiak	BAS MUS
2501	Victor Booth at Comdex To Display Latest Products	LST VTI
2502	Victor Promises New Software Policy for V9000 Owners	VTI
2503	Victor Continues Its Profitable Ways	VTI
2503	Questions and Answers	BAS COM DOS EDU HAR SCR HAR NEW
2505	Apricot Computer Arrives in U.S.	BAS SCR
2505	BASIC Program Enables 25th Line / Bob Marcil	
2507	Technical Word Processors for the Victor 9000 - State of the Art / Chris D'Elia	WP

TABLE 1. CUMMULATIVE INDEX (Page 3 of 7)

V/N/P	Title / Author	Cat.	V/N/P	Title / Author	Cat.
2514	Public Domain Library Adds Videotape Feature / Darrell Bross	PD	2802	Questions and Answers	BAS DOS GRA LNG:C SCR UTL WP:WP HAR
2516	Victor User Groups	LST	2805	A Personal View - Why I Chose the Victor 9000 / Chris D'Elia	HAR
2517	[Chapter News]	NEW	2805	Special Features of the Victor 9000 / Chris D'Elia	PD
2519	StarPolish is "Amazing to Watch"	WP:WS	2806	Disk Demand Greater Than Expected	LST
2520	Display Driver Escape Routines and Their GMBASIC Equivalents / Andrew Rudiak	BAS SCR	2808	Victor User Groups	NEW
2522	New Idea for Batch Files: Create a Selection Menu / Ron Trenowski	BAT	2809	[Chapter News]	BAS MUS
2523	Modem Madness / Jack W. Dukes, Sr.	HUM	2810	"Bumblebee" Takes Flight on the Victor / Andrew Rudiak	DB HAR MIS SPR
2601	ACT Steps Aside: Beta and Datatronic Bid for Victor	VTI	2812	Product Announcements	HAR VTI BAS DB SCR WP NEW
2602	Questions and Answers	CVM GRA MEM WP:WS VTI	2901	Victor Begins Marketing Push	PD
2603	Comdex Successful for Victor and Victor People	LST	2902	Questions and Answers	BAS
2612	Victor User Groups	NEW	2903	Farewell and Hello / Jack Skalicky	PD
2613	[Chapter News]	NEW PD	2906	Some Comments on Public Domain Disks / Mary Jo Knobelsdorf	BAS
2615	User Group Info	BUS DB GRA HAR LNG	2906	A "Random Numbers" Method / Dan Morrow	LST
2616	Product Announcements		2908	Victor User Groups	NEW
2701	Victor's MS-DOS 2.1 Has Superior Features / David Minehart	DOS	2909	[Chapter News]	NEW
2702	Victor to Sign Agreement with Beta "Any Day Now"	VTI	2912	Notice to our Readers / Andrew Rudiak	NEW
2703	Questions and Answers	CVM DOS GAM HAR MUS SPR	3101	Act II, Scene I	NEW
2704	New Mailing Address	NEW	3102	Victor News(?)	HAR VTI
2708	Victor User groups	LST	3102	Stop! Save a Page! / Mary Jo Knobelsdorf	PRN
2709	[Chapter News]	NEW	3103	BASIC in the nude / Guy Gordon	BAS PAT
2710	Copy Your Ramdisk Files - Automatically / Lauren R. Pacini	BAT WP:WS	3104	WORDSTAR - STRIP = SPELLBINDER / Joseph Zmuda	DOS UTL WP:WS PD
2712	Product Announcements	DB HAR LNG:PAS WP	3105	Public Domain Catalog / Mary Jo Knobelsdorf	MIS
2801	Swedish Software Company Buys 90% of Victor	VTI	3108	Addenda & Errata	BAS CHR MEM PAT
			3109	Questions and Answers	PD PRN SCR UTL NEW
			3110	User Groups	

TABLE 1. CUMMULATIVE INDEX (Page 4 of 7)

V/N/P	Title / Author	Cat.
3201	Today's Docket	NEW WP:WP
3202	OMNIFILE / John Pietro	DB
3203	Using EDLIN / Lawrence Akutagawa	EDI
3209	The Bill Johnson Show / R. W. Johnson	BAS PAT
3211	.C Centering Using .Y Settings with P Control / Boardman Moore	PRN WP:SB
3212	["CAKE" bug fix]	PAT
3213	Public Domain / John Knobelsdorf	PD
3214	Alphadates / R. W. Johnson	BAS D/T
3214	Bug of the Month [TIMEUP]	PAT WP:SB
3216	BATCHing Backups / R. W. Johnson	BAT
3301	Bits & PC's	NEW LST VTI
3302	Word(almost)Perfect / Andrew Rudiak	WP:WP
3304	Additional DISKID [Bugs and patches]	BAS BUS DOS KB LNG:ASM PAT PRN SCR TEC WP:WS
3309	The Carlisle Report / C. H. Carlisle	PD
3311	Garfinkle at large / Bob Garfinkle	DOS MEM PAT PD
3311	[Query on LAN experiece]	LAN
3312	Public Domain / John Knobelsdorf	PD
3401	ShortStuff	DIR NEW SAL TEC
3402	Top Secret / Lirpa Loof	HUM
3403	More Additional DISKID / Sheri Mullen & Robert Stephen	CPM EDI EMU SPR:123 SPR:MP TEC
3405	Left-handed Lithuanians... / Andrew Rudiak	PRN WP:SB
3408	MAG/base, Files & Folders / Lawrence Akutagawa	DB

V/N/P	Title / Author	Cat.
3411	Down the Primerose \Path / Wm. Wilgus	BAT DOS WP:WP WP:WS
3413	Public Domain / John Knobelsdorf	PD
3501	Mea Culpa	KB MUS NEW SAL VTI
3502	Now Presenting... [VI & VPC]	HAR VTI
3504	Miss May	D/T GRA PRN
3505	Compiling BASIC / R. W. Johnson	BAS
3508	A Visit with Al Bullock / Boardman Moore	MIS
3509	An ELECTRIC WEBSTER Patchwork / Joe Zmuda	PAT WP:SB
3509	Another Report From C.H.Carlisle / C. H. Carlisle	DB:dB2 PD PRN SCR
3510	Fixing ALIEN	BAS GAM PD SCR
3511	Public Domain / John Knobelsdorf	PD
3512	More Material for the SPELLBINDER Quilt	PAT WP:SB
3512	[Patch for NUKE-LA.BAS] / Bob Garfinkle	BAS GAM PAT PD SCR
3601	Ten-Hut!	DB:dB3 NEW SAL VTI
3602	DS or not DS [Doub.-sided drive] / David Rensin	HAR MEM
3603	MSBASIC Animation / David Ortman	BAS GAM
3604	Sharing Data / Dan Mardesich	DB:dB2 PRN SPR
3605	UG! [User group formation]	MIS
3606	Linking FABS and TURBO / Bob Marcil	DB UTL LNG:ASM LNG:PAS
3609	Public Domain / John Knobelsdorf	PD
3610	Music(?) Column [Ticotico] / [Andrew Rudiak]	BAS MUS
3701	Burnout	NEW SAL VTI

TABLE 1. CUMMULATIVE INDEX (Page 5 of 7)

V/N/P	Title / Author	Cat.
3702	Stuffing DIR / John Kimber	CPM MEM
3703	Another Left-handed Lithuanian [Speech synthesis] / [Andrew Rudiak]	BAS DOS GAM HAR MUS PAT
3704	Spellbinder Revisited [Printer control] / Boardman Moore	PRN WP:SB
3707	Bridging FABS with TURBO / C. H. Carisle	DB LNG:PAS
3710	A Menu Subroutine / R. W. Johnson	BAS SCR
3801	Easter Issue	GRA MUS NEW SAL SPR:MP VTI
3802	Report Manager AV / John Kimber	SPR SPR:VC
3803	FINCALC	FIN
3806	STARWARS Deglitched / David Ortman	BAS GAM PAT
3807	Raising VICTMINI's IQ / Bert Happel	COM KB
3808	Garfinkle at Large / Robert Garfinkle	BAS BAT WP:WS
3809	Fire Sale II - The Sequel	BAS BAT PRN SAL WP
3810	Public Domain / John Knobelsdorf	PD
3812	A Patch for FINANCE.BAS (V*G#17) / Chuck Guittari	FIN LNG:PAS PAT
3812	Bridging FABS Errata / C. H. Carisle	DB PAT
3901	Stop the Presses!!	BUS SAL VTI
3902	Hints / David Ortman	DIR DOS UTL WP:SB
3903	The Victor Support Memo	VTI
3908	[Screen Driver User's Guide]	SCR
3908	New SPELLBINDER Upgrade	WP:SB
3909	Public Domain / John Knobelsdorf	PD
3910	Software Tip of the Month	PAT
3910	Anyone Have PC-FILE That Runs on Victor?	DB
3912	BEEP - it yourself / Bert Happel	LNG:ASM MUS PAT SCR

V/N/P	Title / Author	Cat.
4101	Hail and Farewell / Andrew Rudiak	HAR NEW SAL VTI
4102	The Moleski Reports / Marty Moleski, SJ	NEW VTI
4103	Warnings [Files that destroy data]	COM PD
4105	DISKCOPY Switches	DOS UTL
4106	Helps - Provided & Requested / John Knobelsdorf	EMU MEM
4201	From Andrew / Andrew Rudiak	HAR KB MEM PRN SAL SCR SPR VTI WP
4203	U.S./Disks / John Pietro	MEM SAL
4203	Fix ELIZA / Clifford Cloonan	GAM PAT
4204	MS-DOS 2.11 / Mary Jo Knobelsdorf	DOS
4208	V9000 Users Communicate / Dale Copps	COM
4210	Will Writer / Andrew Rudiak	MIS WP
4212	Victor Circle / John Knobelsdorf	MIS
4213	Share Your Public Domain	PD
4214	Victor's Sale Continues / Mary Jo Knobelsdorf	SAL
4216	Publisher's Page	NEW SAL SPR
4217	Public Domain	PD
4218	Winchester Disk Drive	MEM
4301	V9000 Software	SAL
4302	PMATE / Frederick Wilson	EDI WP:PMA QUS
4305	Question V Responses	MIS PD
4307	FOGFIND [Measures writing complexity]	
4308	MS-DOS 2.11, Part 2 / Mary Jo Knobelsdorf	DOS
4314	Telecommunications / Joe and Ann Reid	COM
4316	Public Domain / John Knobelsdorf	PD
4401	Welcome a Board	COM DOS SAL
4404	So You Want to go Double Sided / David Elesh	MEM
4405	Screen Dump Found	PRN SCR
4406	More on DS Drives / Michael Wishnietsky	MEM



TABLE 1. CUMMULATIVE INDEX (Page 6 of 7)

V/N/P	Title / Author	Cat.
4407	What Do We Mean When We Say Boot?	MIS
4407	Brooklyn Bridge / G. Gordon	OMM DOS
4408	Odds and Ends / Bert Happel and others	OMM:VMI DB MUS SPR:MP TEC
4410	A Guide to Knee Toe Graphics / Jan Ewing	GRA KTG
4414	What We Use and What We Think	BAS OMM DB DOS FIN GRA LNG QUS SPR UTL WP
4501	Get on Board / Franz Hirner	OMM
4504	Victor Circle Gets Us Deals / Dale Copps	MEM SAL
4506	VI Review / Don Henderson	DOS HAR
4508	Raise Your Batch Quotient / Joe Zmuda	BAT KB PRN SPR WP
4510	How to Live with Your VDT / Bert Happel	SCR
4513	From the Boards	BAT OMM GRA HAR MEM PRN SPR
4518	Compressed Files / Bert Happel & John Knobelsdorf	ARC FIL LBR SQZ
4522	Publisher's Page	OMM DOS GRA HAR KTG MIS
4601	Time to Renew	IND
4602	Software - Latest Version / B. Chase & E. Schrader	OMM CPM DB DOS EDI GAM GRA LAN LNG SPR UTL WP
4604	Finish the Questions	QUS
4607	V9000 Vendors and Sources / Marti Moleski	LST
4614	MS-DOS 2.11, Part 3 / Mary Jo Knobelsdorf	DOS
4619	Nested Batch Files / Wayne Crouch	BAT MEM

V/N/P	Title / Author	Cat.
4620	Public Domain / John Knobelsdorf	PD
4621	Warning Tones / Joe Zmuda	DB MUS WP:SB
5101	Changing Resident MS-DOS Character Sets & Keyboards / J. S. Hastey	CHR DOS KB
5104	MS-DOS 2.11 Command Formats By Function Category / Jack Prichard	DOS
5106	From the Boards / Mary Jo Knobelsdorf	DOS UTL
5108	Fastrax / Franz Hirner	MEM
5110	Conversion to MS-DOS 3.1	DOS
5113	Ram Size Limits	MEM
5113	Victor 9000 Service	VTI
5113	Victor Plus PC	HAR MEM
5113	Re: Vendors and Sources	LST
5114	MS-BASIC, Part 1 / Jack Prichard	BAS
5121	Questions, Questions, Questions / John Knobelsdorf	NEW PD
5124	More on Warning Tones / R. W. Johnson	BAS MUS
5201	BAVUG RBBS / Franz Hirner, Sysop	ARC OMM LBR LNG:C MEM SQZ DB:dB2 WP:SB
5205	Spellbinder/dBASE II Data Exchange / Lexisoft	DB:dB2 WP:SB
5205	MS-BASIC, Part 2 / Jack Prichard	BAS
5214	Upgrading the Victor 9000 / Roger Habeck	DOS HAR MEM WP:WS
5216	Technically Yours Membership	MIS
5217	Plus PC With Hard Disk in Victor Format / Wayne Crouch	EMU MEM TEC
5218	Sheet Label Printing with Spellbinder / Boardman Moore	WP:SB
5219	Victor 9000/Vicki BIOS Listing / Victor Technologies	TEC
5220	From the Boards	ARC OMM:XMO DB:dB3 WP:WS PD
5222	Library Report / John Knobelsdorf	PD

TABLE 1. CUMMULATIVE INDEX (Page 7 of 7)

V/N/P	Title / Author	Cat.	V/N/P	Title / Author	Cat.
5301	A Look Back / John and Mary Jo Knobelsdorf	DOS MEM MIS	5416	From the Boards	DB EMU MEM SPR:MP WP:WP
5304	Victor Circle Newsdisk	LST	5417	MS-DOS 3.1 - Change from 2.11	DOS
5305	MB-BASIC, Part 3 / Jack Prichard	BAS	5422	The Library Report / John Knobelsdorf	PD
5314	Re: Upgrading the Victor 9000 / Marty Moleski	HAR	5501	Public Domain Library Index	IND
5314	Valid Memory Board Troubles in a Plus PC / Wayne Crouch	MEM	5510	MS-BASIC, Part 5 / Jack Prichard	BAS
5315	Accessing System Time in dBASE II / Steve Venitsky	D/T DB:DB2	5520	Victor Computers for Sale	SAL
5316	Plus PC and VI Upgrades	HAR SAL	5520	Victor Technologies Relocating to East / Don Clark	VTI
5317	From the Boards	MEM WP	5521	Fix for Bug in Print Routine of Knee Toe Graphics / J. P. Hastey	KTG PAT
5318	Some Thoughts on MS-DOS 3.1 / Oliver Deex	ARC CHR DOS KB	5522	More on Knee Toe Graphics	KTG
5322	BAVUG RBBS / Franz Hirner	ARC OMM SPR:MP	5523	From the Boards	EMU IND WP:WP PD
5401	CONFIG.SYS and AUTOEXEC.BAT / Franz Hirner	AUT BAT DOS	5529	The Library Report / John Knobelsdorf	PD
5406	V9000 Computers for Sale	SAL	5601	Time to Renew!	MIS
5406	Format Change	MIS	5601	Users Group Update	LST
5407	Final Sale on Software for Victor 9000 Systems	SAL VTI	5603	VT Cache	MEM SAL
5408	MS-BASIC, Part 4 / Jack Prichard	BAS	5604	Directories / Jack Prichard	DIR
5412	New Eyes for Old / Roger Habeck	SCR	5611	Help Wanted	MIS
5413	How to Clean the Victor 9000 Screen / Hermann Maseler	SCR	5612	Questions and Answers for the Victor 9000 Novice / Sam Martin	TEC
5413	Re: Valid Memory Board Troubles in a Plus PC / Wendy Cykana	MEM	5615	From the Boards	ARC OMM CMM:KER CMM:XMO MEM
5414	Messing with MULTO / Bud Popken	DOS UTL	5621	Victor Computers and Repairs	SAL VTI
5415	V9000 and VI Hardware and Software	SAL	5622	The Library Report / John Knobelsdorf	PD
			5623	Further Victor Technologies Sale	SAL VTI
			5623	Victor Computers for Sale	SAL

TABLE 2. CUMMULATIVE INDEX BY CATEGORY (Page 1 of 4)

Category	Description	V/N/P (Volume/Number/Page)
ARC	ARChiving utilities and information about Backup and Restore, and file compression and expansion techniques for communications purposes. See also LBR and SQZ.	4518 5201 5220 5318 5322 5615
AUT	AUTOexec.bat programs	5401
BAS	BASIC programs (including MS-BASIC, GW-BASIC, VBASICA, etc.) and tutorials on writing BASIC programs.	1306 2106 2202 2311 2312 2414 2503 2520 2802 2810 2902 2906 3103 3109 3209 3214 3304 3505 3510 3512 3603 3610 3703 3710 3806 3808 3809 4414 5114 5124 5305 5401 5408 5510

TABLE 2. CUMMULATIVE INDEX BY CATEGORY (Page 2 of 4)

Category	Description	V/N/P	(Volume/Number/Page)
BAT	BATch files and tips on writing BATch files.	2314 2522 2710 3216 3411 3808 3809	4508 4513 4619
BUS	BUSiness and accounting software including accounts receivable, accounts payable, checkbook accounting, etc.	2616 3304 3901	
CHR	CHaRacter files and means of displaying and printing different character sets (fonts).	3109 5101 5318	
GM	CoMMunications, including transfer protocol, modems, bulletin board systems (example, FIDO) and communication software (examples, Vicmini, Kermit, etc.). Some GM software is further identified:	2202 2310 2503 2602 2703 3807 4103	4208 4314 4401 4407 4414 4501 4513
GM:FDO	FIDO BBS	-	
GM:KER	KErmit	5615	
GM:VMI	ViCMini	4408	
GM:XMD	XMDem	2203 5220 5615	
CPM	CP/M utilities	2203 2204 2404 3403 3702 4602	
CVS	ConVerSion of programs (IBM to Victor), DOS2 TO DOS3, and programs to convert output (Wordstar to ASCII, ASCII to hex, etc.)	-	
D/T	Date/Time. Programs for displaying, modifying, etc. date/time information (both at system level, and within various application programs), keeping diaries, calendars, and time records.	3214 3504 5315	
DB	DataBase programs. Most DB software is further identified:	1203 1204 2305 2616 2712 2812 2902	3202 3408 3606 3707 3812 3910 4408
DB:db2	dBASE II	1202 1302 1303 2104 3509 3604 5205	5315
DB:db3	dBASE III	3601 5220	
DB:FEX	File Express	-	
DB:PC3	PC-FILE III	-	
DIA	DIAGnostic programs to test and find hardware and software problems.	-	
DIR	DIRectory programs for displaying or printing lists of files or otherwise manipulating (searching, finding, modifying, sorting, re-dating, deleting, etc.) files.	2303 2402 3401 3902 5604	
DOS	Disk Operating Systems such as MS-DOS, PC-DOS, CP/M, XENIX, etc. and shells such as Marti Moleski's MULTO.	1304 2102 2204 2206 2303 2402 2503	2701 2703 2802 3104 3304 3311 3411
		3703 3902 4105 4204 4308 4401 4407	4414 4506 4522 4602 4614 5101 5104
		5106 5108 5214 5301 5318 5401 5414	5417
DTP	Desk Top Publishing, primarily what has become to be known as desk top publishing but includes information of a general nature related to publishing.	-	
EDI	EDItors, programs for line editing (such as EDLIN and similar shareware programs) but less than complete word processing programs (such as Wordstar, Spellbinder, etc.)	3203 3403 4302 4602	
EDU	EDUcational programs, including typing tutorials, math problems, spelling flash cards, etc.	2503	
EMU	EMUlators, such as those emulating CP/M in MS-DOS, or emulating an IBM PC on a Victor 9000 computer.	3403 4106 5217 5416 5523	
FIL	FILE manipulation programs.	4518	

TABLE 2. CUMMULATIVE INDEX BY CATEGORY (Page 3 of 4)

Category	Description	V/N/P (Volume/Number/Page)
FIN	FINAncial programs for making business calculations (amortization schedules, present value, future value, interest, etc.), forecasting, planning and budgeting and investment programs (such as Wall Street Weekly).	1306 3803 3812 4414
GAM	GAMES, including card games, board games, action games, narrative type games, ball games, crosswords, etc. (many of which include some graphics).	2703 3510 3512 3603 3703 3806 4203 4602
GEN	GENealogy programs, some use BASIC, some need dBASE II.	-
GRA	GRAphics, programs in which graphics are an important element (such as certain games and financial programs, or programs to facilitate the production of graphics other than Knee Toe Graphics.) All programs and information related to Tony Casano's Knee-Toe Graphics are identified with KTG.	2602 2616 2802 3504 3801 4410 4414 4513 4522 4602
HAR	HARdware, including V9000, +PC, and VI computers and computer system components such as CPU's, keyboards, monitors, disk drives, printers, plotters, the NEC 20 chip, etc.; and fixing hardware.	1306 2203 2503 2505 2616 2703 2712 2805 2812 2901 3102 3502 3602 3703 4101 4201 4506 4513 4522 5113 5214 5314 5316
FLM	FLM or related to computers and computer users.	2403 2523 3402
IND	INDexes, including cross-indexes, tables of contents, disk indexing utilities, etc.	4601 5501 5523
KB	KeyBoards, including means of changing the characters or functions associated with the keyboard keys and .KB files containing these keyboard assignments.	3304 3501 3807 4201 4508 5101 5318
KTG	Knee Toe Graphics by Tony Casano.	4410 4522 5521 5522
LAN	Local Area Network software and related information.	4602
LBR	LiBRary utilities for file compression and expansion.	4518 5201
LNG	LaNGuages excluding BASIC (.BAS) (which includes MS-BASIC, GW-BASIC, and VBASICA), whose programs and articles are sufficiently numerous to justify a separate category. Widely used and well represented languages are further identified:	2616 4414 4602
LNG:ASM	ASseMbly language (.ASM)	3304 3606 3912
LNG:C	C language (.C)	2802 5201
LNG:FOR	FORtran (.FOR)	-
LNG:PAS	PAScal (.PAS)	2712 3606 3707 3812
LST	LiSTs of user groups, bulletin boards, vendors, software, etc.	2306 2412 2501 2516 2612 2708 2808 2908 3301 4607 5113 5304 5601
MAT	MAThematics, including mathematical and statistical computations (including the use of the 8087 series math chip).	2410
MEM	MEMory and off-line storage, including RAM, ROM, floppy disks, hard disks, ram disks, the V9000 memory map, etc.	1107 2102 2303 2402 2602 3109 3311 3602 3702 4106 4201 4203 4218 4404 4406 4504 4513 4619 5108 5113 5201 5214 5217 5301 5314 5317 5413 5416 5603 5615
MIS	MIScellaneous, including BBS chit chat and any subject that does not reasonably fit in any other category.	2812 3108 3508 3605 4210 4212 4307 4407 4522 5216 5301 5406 5601 5611
MUS	MUsic and Sound, including audio output from the computer in any form (beeps, voice, noise effects, as well as music.)	2202 2414 2703 2810 3501 3610 3703 3801 3912 4408 4621 5124
NEW	NEWs of a general nature, as opposed to news specific to another category (such as VTI, news specific to Victor Technologies, Inc.)	1101 1104 1108 1201 1205 1208 1301 1305 1308 2101 2105 2108 2201 2207 2208 2301 2302 2307 2315 2316 2413 2505 2517 2613 2615 2704 2709 2809 2903 2909 2912 3101 3110 3201 3301 3401 3501 3601 3801 4101 4102 4216 5121

TABLE 2. CUMMULATIVE INDEX BY CATEGORY (Page 4 of 4)

Category	Description	V/N/P (Volume/Number/Page)
PAT	PATches and fixes, including information on the use of Disk Tools and DEBUG, etc, for restoring files and patching programs and the patch information itself.	3103 3109 3209 3212 3214 3304 3311 3512 3701 3806 3812 3910 3912 4203 5521
PD	Public Domain program information, library catalogs, alphabetical listings of public domain files, etc.	2315 2404 2514 2615 2806 2906 3105 3109 3213 3309 3311 3312 3413 3509 3510 3511 3512 3609 3703 3909 4103 4213 4217 4307 4316 4620 5121 5220 5422 5529 5622
PRN	PRINters, including dot matrix, daisy wheel, ink jet, laser types, etc. and including plotters, as well.	3102 3109 3211 3304 3405 3504 3509 3604 3704 3809 3810 4201 4405 4508 4513
QJS	QJestionnaires.	4305 4414 4604
SAL	SALes of both hardware and software, including listing of materials for sale, generally with price and other specifics.	1307 2106 3401 3501 3601 3701 3801 3809 3901 4101 4201 4203 4214 4216 4301 4401 4504 5316 5406 5407 5415 5520 5603 5621 5623 5624
SCR	SCReen utilities, including programs relating to the display of computer output and "screen dumps" (print screen on the IBM PC).	1107 2312 2503 2520 2802 2902 3109 3304 3509 3510 3512 3710 3908 3912 4201 4405 4510 5412 5413
SPR	SPRead sheets, including shareware spreadsheets programs such as ASEASY, version 2.8 (on PD #90). Widely used and well represented spreadsheets are further identified:	1204 1303 2406 2703 2812 3604 3802 4201 4216 4414 4508 4513 4602
SPR:123	Lotus 123	2102 3403
SPR:MP	Microsoft's Multiplan	3403 3801 4408 5322 5416
SPR:MTH	WordPerfect's Mathplan	-
SPR:VC	VictorCalc	1306 2303 2402 2406 3802
SQZ	SQeeZe and unsqueeze utilities for file compression and expansion.	4518 5201
TEC	TECHnical notes, Victor's Additional Diskid's, tips and "how-to's" of all sorts that are not better suited for listing under a more specific category.	3304 3401 3403 4408 5217 5219 5612
UTL	UTILity programs of all sorts (examples: reset the computer from the keyboard, make the cursor blink, set and clear the 25th line, make menus for programs, etc.) other than those listed under a more specific category like OMM, CVS, DIA, DIR, PRN, SCR, etc. (CP/M utilities are listed under CP/M.)	2802 3104 3109 3606 3902 4105 4414 4602 5106 5414
VTI	Victor Technologies, Inc. news, including information on general subjects not better listed under a more specific category.	1102 1201 1207 2302 2304 2401 2501 2502 2503 2601 2603 2702 2801 2901 3102 3301 3501 3502 3601 3701 3801 3901 3903 4101 4102 4201 5113 5407 5520 5621 5623
WP	Word Processing and utilities, including use of application programs such as Wordstar, Word Perfect, Spellbinder, etc. and utilities such as making mailing labels. Widely used and well represented application programs are further identified:	1105 1206 2311 2507 2712 2902 3809 4201 4210 4414 4508 4602 5317
WP:FMA	FMAte	4302
WP:SB	SpellBinder	1106 3211 3214 3405 3509 3512 3704 3902 3908 4621 5205 5218
WP:WP	Word Perfect	1106 2802 3201 3302 3411 5416 5523
WP:WS	WordStar	1105 1303 2102 2206 2519 2602 2710 3104 3304 3411 3808 3908 5214 5220

TABLE 3. CUMMULATIVE INDEX BY AUTHOR

Author	V/N/P (Vol/No/Pg)
Akutagawa, Lawrence	1306 2106 2206 3203 3408
Barnes, Steve	1106
Booi, Greg	1207
Bross, Darrell D.	2201 2315 2404 2514
Carlisle, C. H.	3309 3509 3707 3812
Chase, B.	4602
Clark, Don	5520
Cloonan, Clifford	4203
Coops, Dale	4208 4504
Crouch, Wayne	4619 5217 5314
Cykana, Wendy	5413
Deex, Oliver	5318
D'Elia, Chris	2410 2507 2805
Dukes, Jack W., Sr.	2523
Elesh, David	4404
Ewing, Jan	4410
Falcon, Marie K.	2102
Garfinkle, Bob	3311 3512 3808
Gordon, Guy	3103 4407
Guittari, Chuck	3812
Habeck, Roger	5214 5412
Happel, Bert	3807 3912 4408 4510 4518
Hastey, J. P.	5101 5521
Hawk, Joe	2305 2403
Henderson, Don	4506
Hirner, Franz	4501 5108 5201 5322 5401
Johnson, R. W.	3209 3214 3216 3505 3710 5124
Jones, Richard	2406
Keachie, Douglas	1202
Kimber, R. John	1302 2101 2104 3702 3802
Knobelsdorf, John	3213 3312 3413 3511 3609 3810 3909 4106 4212 4316 4518 4620 5121 5222 5301 5422 5529 5622
Knobelsdorf, Mary Jo	2906 3102 3105 4204 4214 4308 4614 5106

Author	V/N/P (Vol/No/Pg)
Leshner, James	1304
Levy, Debora	1203
Lexisoft	5205
Loof, Lirpa	3402
Majhor, Mark R.	1105 2202
Marcil, Bob	2312 2505 3606
Mardesich, Dan	3604
Martin, Sam	5612
Maseler, Hermann	5413
Minehart, David	1206 2310 2311 2701
Moleski, Marty, SJ	4102 4607 5314
Moore, Boardman	3211 3508 3704 5218
Morrow, Dan	2906
Mullen, Sheri	3403
Ortman, David	3603 3806 3902
Pacini, Lauren R.	2710
Pietro, John	3202 4203
Popken, Bud	5414
Prichard, Jack	5104 5114 5205 5305 5408 5510 5604
Reid, Ann	4314
Reid, Joe	4314
Rensin, David	3602
Rudiak, Andrew	2202 2414 2520 2810 2912 3302 3405 3610 3703 4101 4201 4210
Schrader, E.	4602
Shafer, Alex	1106
Skalicky, Jack	1107 1201 2903
Stephen, Robert	3403
Stuhr, Michael	2304
Trenowski, Ron	2314 2522
Venitsky, Steve	5315
Victor Technologies	5219
Wilgus, William	3411
Wilson, Frederick	4302
Wishnietsky, Michael	4406
Zmuda, Joe	3104 3509 4508 4621

\* \* \*

# PRUFREAD / Jan Ewing

[Jan Ewing has been a prolific contributor to Victor Circle's NewDisks; his definitive article "A Guide to Knee Toe Graphics" was "reprinted" in VictorTalk, Vol. 4, No. 4, page 10. This article on PRUFREAD is composed of excerpts from his recent letter and a verbatim copy of his flyer. It would appear to be of interest and use to many users of WordPerfect. - Ed.]

I'm contacting Victor groups about a very useful WordPerfect utility I've developed called PRUFREAD. PRUFREAD is available for the Victor 9000 as well as the IBM-PC family, and I know (since I am a Victor user myself) that announcements of useful new Victor products are usually appreciated.

...

PRUFREAD is currently being used by (among others) Random House, the University of Pennsylvania, MacMillan Publishing, In-House Computer Services (a Victor shop), and the Washington State Internal Revenue Service. It was also recommended on the last Word Perfectionist Disk Service\* disk (an "on-disk" service and on the Word Perfectionist online computer bulletin board.

The program is presently available for the V9 and the IBM-PC. The retail price is \$69.95. For VUG members, however with one of these flyers (or a xerox copy of same), either version (yes, I'm afraid many of our staunch supporters are actually using that other machine) is \$49.95.

---

[The flyer follows. (I would think that a copy of this page would be acceptable for the above offer.) - Ed.]

## PRUFREAD

A proof reading utility for WORD PERFECT  
Secondary Merge Files

WORD PERFECT offers what is certainly one of the most powerful database MERGE utilities of any word processor on the market. But how many of us have laboriously typed in our lists with all those ^R and ^E codes, only to have 500 envelopes ruined because somewhere near the beginning an ^R or ^E was inadvertently left out?

---

\* The Word Perfectionist is a newsletter published by an independent group of WordPerfect users/supporters. It's recommended by WordPerfect Corp., however, and is widely recognized as the definitive WP user interface.

## THOSE DAYS ARE GONE FOREVER!!!

PRUFREAD automatically performs some of the most labor intensive chores associated with the creation of WORD PERFECT Secondary Merge Files. Specifically,

- it lists all records which do not contain the correct number of fields;
- it automatically removes EXACT duplicates;
- it detects and reports on missing End of Record (^E) marks, and
- it flags records within a user-specified percentage of being possible duplicates.

PRUFREAD quickly scans and proofs any WORD PERFECT Secondary Merge File with records containing as many as sixty fields. AND IT'S FAST!! PRUFREAD will scan a Secondary Merge File containing 2,300 records (with seven fields each) in approximately two and a half minutes\* (think how long it would take to do this manually??). The resulting error analysis can be sent to a WORD PERFECT text file or to a printer. This analysis tells you exactly where to look, so errors can be corrected in seconds.

PRUFREAD is completely accurate and EXTREMELY easy to use. Just answer the few simple questions on the single screen. That's all there is to it. There are no complicated instructions or manuals to master, although the program does come complete with clearly written documentation which can be printed on any printer.

PRUFREAD is currently available for the IBM-PC/XT/AT (and compatibles) and for the Victor 9000. The price is \$49.95 (New York residents add \$4.12 for sales tax). Contact:

MR. JAN EWING  
CUSTOM COMPUTER SERVICES  
409 W. 24th Street, Suite #14  
New York, NY 10011  
(212) 675-1974

Make checks or money orders payable to JAN EWING.

---

\* Speed varies with computer configuration and mailing list density. Figures are based upon an IBM-AT harddisk scanning a professional mailing list at 6mz.

# USEFUL INFORMATION FOR NEW VICTOR 9000 USERS

/ Roger Folsom

[The following article is based on a letter the author wrote to a new Victor 9000 user and his father after talking to them at the BAVUG meeting. It contains much information helpful to new Victor 9000 users (and likely to some older ones, as well). - Ed]

I greatly enjoyed talking with you and your father at the Bay Area Victor Group meeting. Congratulations on having obtained the best 8088 microprocessor computer around. It isn't as fast as the latest 80286 or 80386 microprocessor machines, but it is still the best, in terms of keyboard quality, screen resolution (400 x 800 pixel count), flexibility (easily loaded alternate keyboards and character sets), and floppy disk capacity (1.2 megabytes, assuming your machine reads and writes double-sided floppies; if it doesn't, you can upgrade).

Here is some follow-up on the things we talked about, plus a bunch of other stuff that comes to mind as I write this.

## HARDWARE

### Random Access Memory

Apparently your machine has 640k of memory. A000 in hexadecimal (hex for short) converts to 640k in decimal. Thus, you do have room for more memory, if you decide you want or need it (e.g. for a ramdrive.) You can go up to 896k (more with a Western Automation expansion board and software, discussed below.) It is possible to add more memory by replacing 64k chips in your expansion boards with 256k chips, but I don't know if your boards are capable of this (they certainly are not, if the chips are soldered in!, as they are on the motherboard), or what the dipswitch settings would be. You could get help on this by leaving a message on a bulletin board. Alternatively, there is an active market (on computer bulletin boards) for used expansion boards, if you decide to replace yours with one from Valid Technologies, which could also contain a clock.

An even bigger expansion board comes from Western Automation Laboratories (memory in excess of 896k is used for a ramdrive, I think), but the clock version of this board is awfully expensive, and it doesn't work with the Valid clock. Al Bullock [39522 Benavente Place, Fremont, California 94538, (415) 651-6183] can supply either Valid Technologies or Western Automation expansion boards, probably at a discount if you remind him you are a BAVUG member; Technically Yours [P.O. Box 1414, Aptos, California 95001, (408) 688-5201] could also; either of them might give you a

trade in on your existing boards. Western Automation [P.O. Box 3438, Boulder, Colorado 80307, (303) 449-6400] might give you an even better deal than Bullock; try them if you're considering a different expansion board.

Incidentally, a clock is a luxury, but it is awfully convenient and you sure get used to it; I wouldn't buy a board without one. If you don't have a clock, ALWAYS set time and date when you boot up, so that all files will be correctly timed and dated.

### Processor Chip Upgrade

You might want to consider upgrading to an NEC V-20 microprocessor chip, to replace your Intel 8088 chip; the NEC is a little (10-15%) faster. All you need for the Victor is an NEC V-20 5 megahertz chip, although the quality of an 8 megahertz chip may be a little bit higher. The megahertz rating of the chip simply is what it is capable of; the actual speed it runs at is governed by the machine. Bullock will charge you \$40 or so for the chip, but I think you can get one direct for \$10 to \$20 from a mailorder house, e.g. one listed in the back pages of BYTE. For my two Victors, I got NEC V-20 chips from Bullock (and had him install them at the same time he installed new bootroms, discussed below), but I have ordered (spare) NEC-V-20 chips (and also memory chips) from Microprocessors Unlimited, 24000 South Peoria Avenue, Beggs, Oklahoma 74421, (918) 267-4961. Very competitive prices; good service. If you buy memory chips, Bullock or Valid or Western Automation could advise you what to buy.

I got spare NEC V-20 chips for fear that mine might die (they never have) and that Intel would win its copyright suit against NEC and that the NEC chips would no longer be available in the U.S.

The NEC V-20 chips work fine except that, with them installed, the FORMAT program that comes with your Victor won't work; a patch is needed. Bullock's price for these chips includes a patched Format program; he got the public-domain patch from a bulletin board. If you got the NEC V-20 chip other than from Bullock (e.g. from Microprocessors Unlimited), you would need to get the patched format program from somewhere else.

### Boot ROMs

The latest bootrom chips work better with some software and have some other advantages; Bullock could give you details. These are soldered chips, so you would need him



or somebody else to put them in unless you know what you are doing soldering on your motherboard.

### Other Suppliers

Other hardware (and software) suppliers are Technically Yours (see above), and Cosmos Engineering, 2538 South Robertson Boulevard, Los Angeles, California 90034, (213) 558-3905. For repairs and perhaps some hardware or software, the latest VictorTalk mentioned former Victor employees David Koshinz and Yorgos Hatzis, 190 Coon Heights Road, Ben Lomond, California 95005, (408) 336-3240; I have heard Koshinz mentioned favorably, but never dealt with him. It is possible that Koshinz and Hatzis have more technical expertise--I don't know; Bullock has a lot.

### Printers

Check back issues of Infoworld in any good library, for blunt Consumer Reports style product reviews. Technically Yours might be a good source for a printer; Bullock doesn't handle them (I think). Cosmos does, but I'd prefer a more local supplier. And Cosmos doesn't price very aggressively (although they're worth calling).

### Modems

If you decide to get a modem, the U.S. Robotics Courier is very highly recommended by Franz Hirner. A 2400 baud (fast) model is available from the Online Store, 3037 Golf Course Drive, Ventura, California 93003, (805) 650-0188 for \$159. This is a steal-price for a quality 2400 baud modem. (I am considering upgrading my EasyData 1200 baud (slow) cheapie, which is giving me some trouble in downloading some files, to this U.S. Robotics model.) Incidentally, you need an external modem; the internal ones don't work with the Victor.

If you get a modem, the telephone bills will kill you unless Hirner's BAVUG bulletin board [(415) 574-7914] is a local call for you, and you refrain from calling any other board. However, there is a packet switching service called PC Pursuit from Telenet that lets you make unlimited modem phone calls nationwide (sorry, no voice) for a flat \$25 per month, provided you call on weekends or between 6 p.m. and 7 a.m. weekdays. You often have to wait to get on, and transmissions are slow using this service (much slower than just calling direct). [PC Pursuit sends each packet to its destination using whatever route happens to be free at the moment, so much time is used during the transmission while PC Pursuit looks for an empty route on which to send the next packet of signals, but the price is right.] In addition to the \$25 per month, there's a \$50 sign-up fee. However, I think this is waived for bona fide user group members; check with Boardman Moore for the appropriate form.

### BOOKS

For an informative and very funny introduction to microcomputers, see:

Peter McWilliams, *The Personal Computer Book*, paperback. Probably not needed for a reference, but worth getting from a library. This book discusses specific machines; you'll see in it why I bought a Victor.

In my judgment, the best introduction to MS-DOS 2.11 (and 3.1) is:

Chris Devoney, *MS-DOS User's Guide*, Que Corporation [Suite 202, 7999 Knue Road, Indianapolis Indiana 46250], c1984.

Another good introduction is:

Richard Allen King, *The MS-DOS Handbook*, Sybex, Inc. [#100, 2021 Challenger Drive, Alameda, California 94501], c1985. With this book be sure to start with chapter 9, which is where the introductory material begins: why this weird organization, I don't know.

Personally, I like the Devoney book better, but both are useful supplements to the MS-DOS manual. I have given the publisher's addresses, but both of these should be available from any computer oriented bookstore.

With MS-DOS 2.11 you should have received from Victor a looseleaf User's Guide in a binder, and also an Operator's Reference Guide II in paperback, not looseleaf. (Incidentally, I have never used MS-DOS 3.1 on my dual floppy system, although some people think it has some advantages over 2.11 on a hard disk system. On the other hand, Franz Hirner claims that 3.1 has some printing bugs; then again, I think I read on some bulletin board that the bugs were fixed in the latest version of 3.1.)

A good elementary introduction to microcomputers and to the Victor is:

Dennis Jarrett, *The Sirius and Victor 9000 Computer Book*, Hutchinson, c1985.

Jarrett is with TP Group, P.O. Box 509, London N1 1YL, England, and he probably can supply this book (and possibly also the Lane book below). When mentioning operating systems, this book assumes an early MS-DOS 1.25, but I still found it very useful background.

A good Victor oriented introduction to Basic programming is:

John B. Lane, *The ACT Sirius 1 Computer*, Prentice-Hall International Personal Computer Group, Englewood Cliffs, New Jersey, and London, England, c1985.

(Sirius is the name the Victor was marketed under in Europe; ACT was the distributor in England. ACT also made the Apricot, which--although it used 3.5 instead of 5.25 inch floppies--was otherwise very similar to the Victor, and in fact

ACT had to get permission to use Victor's design patents and/or BIOS [Basic Input Output System] and other software copyrights). The title is somewhat misleading, because this book is more an introduction to Basic programming than it is an introduction to the Victor. Nevertheless, because this book assumes one is working on the Victor, it is nice to have. Prentice-Hall in New Jersey told me it was out of print, but Prentice-Hall in England might be more on the ball. I got my copy from TP Group, mentioned above.

If you get interested in assembly language programming, a good reference (despite the title) is: Murray Sargent III and Richard L. Shoemaker, *The IBM Computer from the Inside Out*, Micro-Computer Books, Addison-Wesley Publishing Company, c1984. The book contains some references to the Victor 9000 (and in fact was written on a Victor 9000), and in any case the 8088 central microprocessor chip used in the Victor is identical to the 8088 chip used in the IBM PC. You can get the book from Scroll Systems, address below.

If you get a modem, a good introduction to computer telecommunications is: Peter W. Gofton, *Mastering Serial Communications*, Sybex, c1986.

## MAGAZINES AND NEWSLETTERS

Jarrett and the TP Group published an excellent Victor-oriented magazine, *16-Bit Computing*, and tried to come out with a U.S. version of same. The U.S. version died after a few issues (not enough advertisers), and the British version died a year or so later (same reason I think), but both were fascinating. I didn't keep mine whole, but tore them up and keep the articles that were most useful to me. Jarrett and the TP Group then came out with a newsletter, the *Victor/Sirius File*, which lasted for a year or so, and contained much useful information. My recommendation would be to write Jarrett, and ask him not only for the Jarrett and Lane books, but also find out if back issues of either the British or U.S. versions of *16-Bit Computing* and of the newsletter are available. Mention my name, if you wish, because Jarrett might remember me. If the newsletter back issues are unavailable, I could loan you mine--but I want them back! As an academician, I take copyright laws very seriously, so I wouldn't want to lend you my back issues for copying unless Jarrett said that he couldn't supply them to you.

Whatever you buy from Jarrett, I'd pay for it by sending him a blank check (which you can limit by writing "not to exceed \$xxx.xx U.S. Dollars" on the face), and let him fill in what he needs to cover the current exchange rate and postage. If you are not in a hurry, tell him to send stuff surface mail (boat), and it will show up in a few months;

otherwise he will assume you want it right away and he will charge you airmail postage. My recollection is that Jarrett's *Sirius/Victor Book* was 18 pounds or so. The Lane book was a little more expensive, I think. I have no idea what back issues of the magazines or newsletter would cost, but the newsletter subscription (including airmail to the U.S.) was 50 pounds. If you can send him a check written in pounds he would love it. I never could get my bank to write a check in pounds at a reasonable fee, so I just sent him a check in dollars and let him charge me for the conversion.

In the United States, for several years Ann and Joe Reid ran a Victor User Group, PSU (Problem Solvers Unanimous [sic; "United" would have made more sense]). They didn't get a big enough subscriber list (a bit more than 500 as I recall) to make this venture profitable enough for them to continue. They got tired and stopped, but the back issues of the PSU newsletter are full of very useful information. For a while, back issues were available for purchase; if they still are, they would be very worthwhile reading, even though they are not cheap. The last two years were indexed. Joe and Ann Reid are at P.O. Box 3244, Scottsdale, Arizona 85257, (602) 946-5948. They are professional accountants, and tax season is approaching, so they might not respond until after April 15, but they will respond eventually. They definitely do know me, so feel free to mention my name. Do NOT phone them until after April 15.

[VictorTalk (published since September, 1983 by the Bay Area Victor Users Group) contains considerable useful information. Most issues (since April 1984) contain summary information on BAVUG's Public Domain Software Library as new disks are released. A cumulative index of VictorTalk appears in this issue. - Ed.]

## SOFTWARE

### Word Processors

In my opinion, the best wordprocessing program is the PS Technical Wordprocessor. It is not a popular program, but I think this is only because it is very poorly marketed. PS is written by Professor Murray Sargent III, Ph.D., a physicist and computer scientist at the University of Arizona, doing business as Scroll Systems, Inc., 5530 North Camino Escuela, Tucson, Arizona 85718, (602) 299-8087. Even as a regular wordprocessor, this program is excellent, but as you can tell from the name, it also does mathematics: Greek and math symbols, summation and integral signs and square root signs drawn automatically as large as needed, fractions with the numerator centered over denominator, underlining, italics, line and box drawing, some graphs, automatic footnote placement (crude but getting better), proportional spacing (although not yet on my TI-855 printer), etc. You can preview--on

screen--what will appear on paper before you print it. I can't think of anything you can't do with PS.

I think PS is about \$400 or so -- expensive, but worth it, especially if you have any need for its capabilities. Originally designed on the Victor, it now runs on both the Victor and IBM machines. If you inquire about PS, mention my name (I don't get a commission, but I've bugged Sargent unmercifully with suggestions and requested modifications, and would like him to know I've recommended his product).

My second choice for a wordprocessing program is now the most popular wordprocessing program: Word Perfect, version 4.2, available on the Victor (if you insist--some of the company's marketing people don't know they have a Victor 9000 version), from Word Perfect Corp., 288 West Center Street, Orem, Utah 84057, (801) 225-5000. This is for sale from Affiliated Computer Groups, mentioned above, and maybe from Technically Yours or Al Bullock, and there is a company sanctioned huge discount for academics and students; my recollection of a flyer from Affiliated is that the cost drops to \$125 or so! Well worth looking into, if the math capabilities of PS don't appeal.

Awhile back Victor distributed free copies of EasyWriter II to various BAVUG members, which was supposed to be pretty good; if you can find a copy, grab it. Some people (particularly Al Bullock and Andrew Rudiak) like Spellbinder, but from what I have read about it, I don't think it measures up, although it does have excellent support. [VictorTalk is produced entirely with Spellbinder (except for headlines, which are produced by Fancy Font). - Ed.] Wordstar, by all reports, is a disaster to learn. My only personal experience is with Sargent's PS.

### Spreadsheets

Technically Yours has an excellent three-dimensional spreadsheet program, Report Manager, for a very low price, I think \$59 or so; the latest version is 2.20 Advanced Version. I have it and think it is great. Originally this was a \$300 or so (maybe \$500) product. Its only disadvantage is that it doesn't read Lotus files, and Lotus 1-2-3 is sort of an industry-standard spreadsheet program. But at the last BAVUG meeting I learned that there are public domain programs to convert Lotus 1-2-3 files into ASCII files (ordinary files), which then could be read into Report Manager. The Lotus 1-2-3 spreadsheet, version 1.x, may still be available from Affiliated Computer Groups, Inc., 4605 Cedar Lake Road, Minneapolis, Minnesota 55416. But unless you need to read Lotus files, I think Report Manager is better; I find the three-dimensional box very useful. Lotus is now well beyond version 1.x anyway, and I have heard that later Lotus programs don't necessarily read early Lotus files,

and maybe vice versa, so owning Lotus 1.x (the latest version for the Victor, I think) might not get you the ability to read all Lotus files that might come your way.

I have an early, primitive, but still useful version of Report Manager called VictorCalc that you would be welcome to have if you want to see what a three dimensional spreadsheet is like. I recall seeing that somebody who frequents the BAVUG bulletin board, and who may attend BAVUG meetings, is enthusiastic about the Multiplan spreadsheet. This now is IBM, but it can be run on the Victor (after you "port it over" to Victor type disks), using some type of not-too-complex installation procedure that has been discussed on the BAVUG bulletin board.

### Utilities

Since you have a hard disk, you may not want to bother with RamDisk (aka RamDrive) software which sets aside some amount (whatever amount you want) of RAM memory to use as an additional disk. I much prefer to work my programs out of RamDrive, because it saves wear and tear on my disk drives and is much faster. Even with a hard disk, I think setting up a RamDrive would be useful, especially if you upgrade your memory to 896k (but I have no experience with a hard disk). Hard disks are fast, but RamDrive is faster. The Western Automation expansion board comes with its own RamDrive software, but with any other expansion board(s), I would recommend the RamDrive software from Dr. Guy Gordon, White Crane Systems, Suite 151, 6889 Peachtree Ind. Blvd., Norcross, Georgia 30092, (404) 454-7911, or (residence) 3194 Friar Tuck Way, Doraville, Georgia 30340. Gordon also has a set of proprietary utilities that are awfully handy for backing up disks, etc. If you order directly from him he might have a package deal, or throw in other public domain utilities that he has checked out or modified. Bullock and Technically Yours also carry some of his stuff, but they may not include the utilities. Incidentally, utilities are small programs that do "housekeeping" chores like copying disks, backing up disks, removing files, copying selected files, etc.

A DiskCache program could be useful, although more to someone such as myself with a floppy system than to someone such as you with a hard disk; I intend to get one. VT Cache (by Valid Technologies) is available, at discount, from both Bullock and Technically Yours.

### Communications

If you get a modem, a public domain communications program to run the modem is Vicmini. Its only deficiency is that it doesn't dial automatically, but you can work around that by using a special telecommunications software keyboard (created with the program Keygen). Alternatively, use

CrossTalk, for about \$100 or so from Technically Yours. (Despite talk of a version 3.61, it will not be available; version 3.5 is the latest.) I use Vicmini but probably will get Crosstalk for the dialing feature.

#### Public Domain and Shareware

You ought to join Victor Circle, a group of volunteers that creates and distributes NewsDisks. These contain tons of information, public domain programs, and "shareware" (if you like it, you send the author a check). They include, not only games, but also useful utilities. [See article at end of this page for current distributor for our area. - Ed.] There is no charge, except for the cost of the disks (cheap ones will do, because you'll read a backup anyway), mailers, and postage--you either send him disks and stamps, or money will do. He can probably supply you with copies of all of the back disks; if not, I can supply the earlier ones. VictorCircle always distributes Newsdisks on single-sided floppies, because not all machines are double-sided. I have Newsdisks 1-20 and also 22 (21 never got done, I think), and then my pre-payment for disks and postage ran out; I need to renew, because Victor Circle is now close to Newsdisk 30.

[Much Public Domain software is available from the Bay Area Victor Users Group. You can make copies on your own disks at the meetings for \$2 per disk. Alternatively, members may order disks by mail for \$5 per disk, plus postage and handling. (See last page of recent copies of VictorTalk for mail order form; see VictorTalk, Vol. 5, No. 5, Sep-Oct 1987 for index; a free Public Domain Library Catalog is available from BAVUG.) - Ed.]

I hope all this stuff is useful. Good luck. If I can answer any questions, don't hesitate to call.

Roger Nils Folsom	Residence office:
Professor of Economics	128 Seafoam Avenue
San Jose State University	Monterey, CA 93940
(408) 924-5418 or -5400	(408) 649-6383

P.S.: See Byte magazine, Vol. 7, No. 11 (Nov. 1982), for an excellent product review of the Victor 9000.

\* \* \*

#### VICTOR COMPUTERS FOR SALE

Victor 9000, single-sided, double drives, new CRT.

Donna Hamlin  
1740 Technology Drive  
San Jose, CA 95110  
(408) 279-4340

Victor 9000, dual floppy disks, 128K RAM, MS-DOS, CP/M, 86 diskettes, plus Victor 9000 128K memory board. Includes palmrest keyboard, Victor green monitor, Multimate Version 3.20.

Bennett Kerns  
Suite 200  
1541 Ocean Ave.  
Santa Monica, CA 90401  
(213) 452-5977

Victor 9000, two double-sided drives, 256K RAM, programmer's keyboard, monochrome (green) monitor; Hayes Smartmodem 300 w/ cable; Hayes Chronograph w/o cable; Daisywriter 2000 daisy-wheel printer w/ tractor feed, six extra printwheels, 32K memory board, and cable; Epson MX-80 printer w/Grafrax w/ cable. Software (w/doc.): MS-DOS Ver. 2.1; CP/M Ver. 2.2/2.4; Crosstalk XVI Ver. 3.5; Multiplan Ver. 1.06; Wordstar w/Mailmerge Ver. 3.30;

All equipment in excellent condition and guaranteed to work properly upon installation. All documentation and software diskettes are complete. Make offer.

Larry B. Foiles  
Foiles Farms, Inc.  
Route 2, Box 21  
Merino, CO 80741  
(303) 522-7249

VI with non-operative floppy drive. Everything else fine. 20MB hard disk. Cheap!

Lesley Alexander  
605 San Roque  
Santa Barbara, CA 93105  
(805) 682-5381  
\* \* \*

#### VICTOR CIRCLE NEWSDISK DISTRIBUTOR CHANGE

NewsDisks are no longer being distributed to Victor Circle members living in California by David Rensin. This responsibility has now been taken over, for not only California, but for Hawaii, Alaska, Oregon, Washington, Nevada, and Idaho, by Michael Wishnietsky. To receive your Newsdisks as they are released, send him a supply of self-addressed, stamped (\$0.39) diskette mailers and diskettes. Alternatively, send \$1.50 per disk and your mailing address; he will provide the diskettes, mailers and postage.

Michael Wishnietsky  
41B Scott Circle  
Bedford, MA 01730  
(617) 274-6689  
\* \* \*

## CIPHERS FOR FUN AND PROFIT / Andrew Rudiak

Ever since I got my Victor, I wanted to run Infocom's games [ZORK (I, II and III), Hitchhiker's Guide to the Galaxy, Leather Goddesses of Phobos and the like] on it. Even though Infocom has supported some very obscure formats (DEC Rainbow, NEC APC, TI 99/4A), it never got around to Victor. Then I got Plus-PC and, wah-lah (as we say in the French quarter of Lafayette, California) - the rest is history.

These games - text simulations with no graphics - run just fine on Victors. Once the programs have been moved to a Victor-formatted disk, you need only do the following:

1. Put ANSI.EXE on the disk.
2. Run the SETUP program. It asks for screen width and lines. Set yours to 79 and 24, respectively.
3. Run ANSI (ANSI <CR>)
4. Run the Infocom program (e.g., LEATHER <CR>)

If you want to run these games but have no way of downloading them, I would be happy to do it for you. Just buy the Infocom game of your choice, mail me the game disk and a blank floppy, include enough stamps or cash to cover return postage, and I'll have it back to you, Victor-ready, in about one week. Be sure you get the generic MSDOS version.

**Leather Goddesses of Phobos** is a "soft-porn" simulation game which parodies pulp science fiction of the 1930s. If you ever saw "Flash Gordon" at a Saturday morning matinee, Leather Goddesses will induce *deja vu*. Steve Meretzky, who wrote this program, loves putting a double whammy on things. An example: At one point - if you get that far - you enter a maze of sewers. A map comes with the game. However, it maps the "lower level" and the "very lower level." You enter at the upper level, for which there is no map. The rooms are indistinguishable, because they are shrouded in mist. Going east after going west does not necessarily return you to the same point. Furthermore, you have to say "KWEIPA", hop or clap at specific intervals or you suffer losses (your location, your possessions, or your life, depending). I have yet to escape the upper level.

Another one of Meretzky's cute tricks is coded messages. You are given a clue in a simple transposition code - shift everything three characters to the left. The sneaky part is

that the message is also backwards. I had better luck with this one. I worked out a BASIC routine which did the decoding for me.

On the next page is that BASIC routine, and another, cipher-key decoding routine. If uncoded language is entered, these routines will encode it. Victor\*Group members in the Bay Area have involuntarily experienced this phenomenon these last few months. The postcards I send out to announce our meetings have been filtered through these routines, much to everyone's consternation (and my fiendish glee).

**Listing #1** translates the coded message(s) in **Leather Goddesses**. It is a no-frills translation algorithm. Your keyboard **must** be shift-locked in UPPER CASE as you input. Only letters, numbers and spaces are translated. Punctuation comes out as gobbledygook. The only frill this routine provides is the slash ("/") character which you use to quit. You might want to add lines to print out the results, or to turn them into a disk file. I'll leave that up to you.

**Listing #2** is also a deciphering routine (or enciphering, if such is your pleasure). It works for simple substitution ciphers. Infocom now prints "VisiClues" in **The Status Line** (which was **The New York Times** until the blue meanies in the NY Times legal department complained). **The Status Line** is Infocom's quarterly newsletter to registered owners of their programs. These VisiClues are - you guessed it - encoded clues to particularly thorny problems.

Infocom thoughtfully provided the substitution key. I will provide the deciphering routine. This program is more user-friendly than Listing #1. Numbers and punctuation are unchanged. Lower-case input is converted to upper case input (all output is UPPER case). You input the cipher, the routine outputs the translation. Enter the exponentiation character (^) to exit the routine.

Infocom may change the substitution key in future VisiClues.

All you will need to do is enter the new key DATA on line #150. To encode text, swap variables C\$(n) and D\$(n) in lines 20 and 30. Swap them back to translate.

Incidentally, these routines output to your monitor only. I leave you the puzzle of how to send it to your printer.

## LISTING 1: LEATHER GODDESSES CODE TRANSLATOR

```

1 REM *** DECIPHER.BAS ***
10 CLS:
20 PRINT"INPUT '/' <CR> TO QUIT":
30 WHILE A$<>" /":
40 LINE INPUT"INPUT CIPHER :";A$:
50 FOR A=LEN(A$) TO 1 STEP -1:
60 CY$=MID$(A$,A,1):
70 IF CY$=" "THEN GOTO 110:
80 IF ASC(CY$)>47 AND ASC(CY$)<58 THEN GOTO 110:' print numbers as-is
90 CY$=CHR$(ASC(CY$)-3):
100 IF ASC(CY$)<65 AND ASC(CY$)>61 THEN CY$=CHR$(ASC(CY$)+26):' ABC wraparound
110 PRINT CY$;;
120 NEXT:PRINT:
130 WEND:

```

PROGRAM BY  
 ANDREW RUDIAK  
 132 VILLA COURT  
 LAFAYETTE, CA 94549  
 (415) 932-5580

## LISTING 2: VISICLUES TRANSLATOR

```

1 REM *** VISICLUE.BAS *** BASIC translation routine for VisiClues
10 CLS:KEY OFF:DIM D$(26),C$(26):'
20 FOR N=1 TO 26:READ D$(N):NEXT:'store decode
30 FOR N=1 TO 26:READ C$(N):NEXT:'store code
40 WHILE A$<>"^":'^ is the break-out character
50 LINE INPUT"INPUT CIPHER: ";A$:
60 FOR N=1 TO LEN(A$):'get letters one at a time
70 CY$=MID$(A$,N,1):'temporary designation of each letter
80 IF ASC(CY$)>31 AND ASC(CY$)<65 THEN GOTO 120:'print symbols as-is
90 IF ASC(CY$)>90 AND ASC(CY$)<97 THEN GOTO 120:'print symbols as-is
95 REM next line converts lower to UPPER case
100 IF ASC(CY$)>96 AND ASC(CY$)<123 THEN CY$=CHR$(ASC(CY$)-32)
110 M=1:WHILE D$(M)<>CY$:M=M+1:WEND:PRINT C$(M);:GOTO 130:'translate cipher
120 PRINT CY$;:'print non-letter characters as-is.
130 NEXT:PRINT:WEND:END
140 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
150 DATA Y,D,T,U,H,S,A,Q,P,B,J,X,V,O,I,G,M,C,E,N,L,Z,K,F,R,W

```

PROGRAM BY  
 ANDREW RUDIAK  
 132 VILLA COURT  
 LAFAYETTE, CA 94549-2042  
 (415) 932-5580

\* \* \*

## V9 FLOPPY TO HD CONVERSION

A do-it-yourself conversion kit for Victor 9000 users wishing to convert their computers to an internal hard-disk machine is now available. All major parts are included: ready-to-boot hard disk (in various sizes) with MS-DOS 3.1 installed (2.11 if equipped with Plus PC card). Owner must have some knowledge of the V9 hardware and furnish minor hardware. Installation is also available for a fee of \$100.00.

10MB Kit	\$495
20MB Kit	595
40MB Kit	795
F3F7 ROMs (Required)	40
Aztec PS (Optional)	150
66MB Kit	Call

Bullock & Associates  
 39522 Benavente Pl.  
 Fremont, CA 94538  
 (415) 651-6183

\* \* \*

## VICTOR 9000 HARDWARE AND SOFTWARE

[A note from Clay Claiborne, Jr., Manager of Cosmos Engineering Company says: "The Victor 9000 is proving to be a real survivor. . . we are an authorized Victor

dealer, and we continue to supply both sales and service of old and new Victor computers." The following is the Victor 9000 portion of their recent flyer. - Ed.]

### 20 MB EXTERNAL HARD DISK

Our 20 MB External Hard Disk for the Victor 9000 is electronically and functionally the same as the internal hard disks that Victor provided. We use exactly the same Direct Memory Access (DMA) Card and Hard Disk Controller Card that Victor did. The Seagate ST-225 we use is 50% faster and much more reliable than the Victor original equipment.

You can boot directly from the hard disk, provided you have the updated boot roms (see below). It will operate under MSDOS 1.25, 2.11, or 3.1. It can be configured with HDSETUP, and will work with the PlusPC kit.

The Hard Disk Unit is contained in a small 3.5" w x 6" h x 14" d metal cabinet, color matched to the Victor, it is connected to the Victor 9000 by a four foot cable, so you can put the unit anywhere that is out of the way. Best of all, it is user installable. If you can install a memory card, you can install this hard drive subsystem.

**We sold lots of these at our catalog price of \$1295, But now it will cost you only \$895**  
**40MB EXTERNAL HARD DISK. \$1095**

**VALID TECH MEMORY BOARD . . . \$395**  
 With 768KB Ram, Clock/Calender and Software. Other configuration available starting at \$300.

**128KB RAM BOARD . . . . . \$98**

**UNIVERSAL BOOT ROMS . . . . . \$30**

You must have these for MSDOS 3.1 or a hard disk larger than 10MB. Recommended for MSDOS 2.11

**PLUS PC KIT . . . . . \$695**

Make your Victor 9000 IBM PC compatible and still run all your Victor programs.

**NUMERICAL DATA PROCESSOR . . . \$275**

**40MB HARD DISK UPGRADE . . . . \$595**

Replace your crowded 10MB hard disk and gain twice the speed and three times the space. It comes to you formatted and loaded with DOS.

**1.2MB DRIVE UPGRADE KIT . . . . \$225**

Replace your single-sided drives. Complete with instructions.

**MONITOR EXTENSION CABLE . . . \$50**

**KEYBOARD EXTENSION CABLE . . \$25**

#### UP GRADE YOUR 128KB VICTOR 9000 TO AN UP-DATED 512KB MACHINE

Exchange your old 128Kb Motherboard for a completely rebuilt unit with Universal Boot Rom, all the latest factory revisions and 16 256Kb 150 nsec. Ram chips for 512KB on the motherboard. This is the cheapest way to increase your Ram for running MSDOS 3.1, WordPerfect, Dbase III, Vbasic and other memory intensive software.

**\$195**

### SOFTWARE

FOR THE VICTOR 9000

- |   |        |
|---|--------|
| MSDOS 3.1 . . . . .   | \$95   |
| WORDPERFECT 4.2 . . . . .   | \$299  |
| MATHPLAN 2.1 . . . . .  | \$175  |
| WCS DISK TOOLS DISK PACK . . .  | \$95   |
| White Crane Systems' Ram Drive, Disk Tools, and Dos Utilities all in package, a \$185 value   |        |
| WCS' BROOKLYN BRIDGE . . . . .  | \$199  |
| Use your PC's Hard Disk from your Victor and your Victor's drives from your PC, transfer files, even use the printer or modem attached to the remote machine. |        |
| CROSS TALK XVI . . . . .  | \$150. |
| THE ACCOUNTING PARTNER . . . . .  | \$329  |

#### ONE OF A KIND SOFTWARE BARGAINS FOR THE VICTOR

- |  |       |
|--|-------|
| LOTUS 123 Release 1.0A . . . . .           | \$195 |
| VictorCALC . . . . .                       | \$50  |
| Bisync 86/3780 . . . . .                   | \$50  |
| Cobol Level 2 . . . . .                    | \$95  |
| Aladin Relational Problem Solver . . . . . | \$50  |

*A neat little database manager that contain a copy of the UCSD P-System operating system from Germany, but manual and program are in English*

**COMPLETE BUSINESS SYSTEM . . . \$100**  
 GL, AR, AP/IO, Inventory/PO  
 Six manuals - Four Disks


**OPEN SYSTEMS ACCOUNTING VER. 2.0**  
 8 Modules, 8 Full Manual and Disks for One Low Price:  
 General Ledger - Accounts Receivable - Accounts Payable - Sales Order Entry - Payroll - Inventory - Team Manager - Business Basic

Originally over \$4800 now only . . . . **\$195**  
 No Support available at these prices.

The West L.A. Victor User Group now meets in our shop the on the third Wednesday of every month at 7:30PM. Come join us and learn more about your Victor.

**REPAIR SERVICE FOR THE VICTOR**

As always we run a complete Victor repair shop, with a 48 hrs. turn-around on all 9000 repairs. With UPS red label shipping we can have you back up and running in 3 days, from anywhere in the continental United States. In the Greater Los Angeles Area we also offer on-site and carry-in service.

 **COSMOS ENGINEERING COMPANY**  
 5317 WEST VENICE BLVD.  
 LOS ANGELES, CA 90019

(213) 930-2540

# FROM THE BOARDS

[FROM THE BOARDS is assembled from messages appearing on various (generally Victor-specific) bulletin boards, mainly The PUB and BAVUG, by John and Mary Jo Knobelsdorf. Only minor editing is done to correct obvious typographical errors and to eliminate some superfluous material. An effort has been made toward a degree of standardized formatting and the elimination of excess space to the extent practical without degrading legibility. Garbles that cannot be readily rectified are allowed to remain. Believing that considerable worthwhile material is contained in these excerpts, our intention is to include this feature in each issue of VictorTalk, as space will allow. - Ed.]

NOTE: The following messages concerning modems, downloads, communication, call waiting, and Spellbinder were downloaded from BAVUG.

#68 12 07 Sep 87 21:27:09 (RECV'D)

From: Roger Folsom

To: Franz Hirner

Subj: DOWNLOADS MODEMS AND PROTOCOLS ETC.

I have been using Vicmini; I don't have Crosstalk because the patches put me off and I figured they would get fixed someday, as I suppose they now have with version 3.61. I have been using Vicmini with Xmodem and Kermit protocols. Over PC Pursuit, Xmodem works rarely (and now Navas has explained why), but over the past few days I have had good luck with Vicmini's Kermit protocol. Probably my earlier failures were due to some dumb mistake on my part, or else due to bad lines. Even Vicmini's Kermit protocol, however, is SLOW over PC Pursuit; a file that Vicmini says should take 5 minutes may take three or four times as long, not only between Monterey and the PUB but also between Monterey and your board.

Until some program available on the Victor uses the high powered protocols mentioned by John Navas, I probably will stick with Vicmini--although the lack of a dialer is a pain. All of the public domain dialers I have seen (other than just buying the commercial PTEL program from which Vicmini comes) operate outside of Vicmini, which creates two problems: first, Vicmini hangs up the modem when you access it, unless you forced it on with the appropriate dipswitch (in which case you can't hang up). (I know you think I'm nuts in saying this, but the CALL.DOC file I got somewhere says the same thing.) Second, and more serious, when you call from outside Vicmini using an echo to seriala, I can't figure out how to know what is going on. For an

ordinary telephone call, this is no problem because you can see your modem lights or listen to its speaker, but for PCPursuit, you need to know what Telenet is doing in order to take the next step, and once you have your connection to your LOCAL PCPursuit number the modem stops paying attention to what is happening, so if you're outside your comm program you miss the news that your destination is busy or not busy, etc.

My problems in using the Kermit PROGRAM --it hangs up when I go local--are outlined somewhere on this board in a message to Lloyd LeBlanc. This is unfortunate because I need the Kermit Program to access the San Jose State mainframe: once I am connected to the mainframe, I need to be able to talk directly to it (using mainframe commands) to tell it what to do. Kermit lets me do that (if it would stop hanging up!); Vicmini apparently does not. For now, I can prevent Kermit from hanging up by setting the dipswitch on.

Thanks--very much--for the concern and help. It was good to learn your opinion that my modem probably is ok, since it works with Vicmini.

Is the X.pc protocol mentioned by John Fiske as supported by Crosstalk 3.61 a "modern" protocol, or one of the old ones that probably wouldn't work too well over PCPursuit?

P.S.: Does anyone know how to disable Call Waiting, so downloads don't crash from an incoming call? Phone co said hit an asterisk followed by 7 0 , but I don't know if the ASCII asterisk

#69 12 08 Sep 87 REC'D. 6:53:45

From: John Navas

To: Roger Folsom

Subj: Comm

Is the X.pc protocol mentioned by John Fiske as supported by Crosstalk 3.61 a "modern" protocol, or one of the old ones that probably wouldn't work too well over PCPursuit?

Yes -- but since it's not directly supported by most BBS systems, it won't help you speed up your file transfers. You need a comm program that supports one of the more advanced file transfer protocols that are widely implemented.



Does anyone know how to disable Call Waiting, so downloads don't crash from an incoming call? Phone co said hit an asterisk followed by 7 0, but I don't know if the ASCII asterisk "\*" preceding the telephone number (as in "ATDT\*705551212") will indeed disable call waiting for that call alone, and the "\*" should be handled correctly by fully Hayes-compatible modems.

It works fine for me.

#80 9 09 Sep 87 07:56:59  
From: Lloyd Leblanc  
To: Roger Folsom  
Subj: Call Waiting

Well, well. Fifteen calls once to the (various) phone companies and it was impossible to learn how to disable CW via the modem. We finally dropped that stuff, since I find it a very irritating intrusion to most people (wait a sec, huh, while I see who that is.....) but it's great to know how to do it!!!! I'm going to send a mailer on that to all the CW fans I deal with.

#81 9 09 Sep 87 08:12:07  
From: John Navas  
To: Lloyd Leblanc  
Subj: Call Waiting

The problem with the disable call waiting code is that it only works for the call originator. Thus, a connection can still be broken by call waiting at the answering end. As far as I know, there is no way to disable call waiting when you answer.

#113 4 10 Sep 87 08:47:13  
From: Lloyd Leblanc  
To: John Navas  
Subj: Call Waiting

Hmm, can't disable the remote CALL WAITING, Hm? I was getting a whole directory of people ready and was going on a disable CW binge. Makes sense, though. Still, nice to know that CW on your end can be disabled via modem commands.

#75 10 13 Sep 87 11:09:49  
From: Roger Folsom  
To: Sysop Franz Hirner And Anyone Else:  
Subj: Comm protocols

Franz: Thanks for the reply. I don't yet know what x.pc is; somebody on this board (probably John Navas) mentioned it as an advanced protocol supported by Crosstalk

3.61. I called Microstuff's 800 number a week or so ago, and they promised to send me the specs on version 3.61, but nothing has shown up in my mailbox yet.

Perhaps you could get Tom Jennings to implement any more sophisticated protocols contained in Fido version 12 in the next version of Ptel, from which Minitel and, hence, Vicmini are derived, since my understanding is that he wrote all of them (Ptel, Minitel, and Vicmini) for Phoenix software. Why Minitel and Vicmini were let into the Public Domain, I can't imagine, unless it was an attempt to attract people into wanting the complete Ptel program.

Your Crosstalk info was very helpful: apparently if you wanted to use a protocol not supported by Crosstalk but supported by some other program (e.g. Vicmini's Telink or Kermit, although the latter is now supported by Crosstalk 3.61), you could use Crosstalk to dial and log on and then exit to dos and run the other program to get the Protocol you wanted. Neat! Does any one know if that is possible with Ptel?

Final comment: The Victor version of the Kermit program, from Columbia University, is a refinement of a generic MSDOS Kermit. The Kermit program manual invites people to write modules which will include a dialer, and also more advanced versions of the Kermit protocol (e.g. sliding windows Kermit). And apparently work is proceeding on redoing all of the various Kermit implementations for different mainframe and micro machines in C. When that gets done, the more advanced Kermit protocols--which are already available in some other Kermit implementations--should become easily and readily available for MSDOS kermit, and hopefully for the Victor version of the latter. My info in this comes from the documentation files included on the Victor Kermit disk I got from Columbia University (but not included in the shortened Kermit program documentation available from the PUB). So with luck, Victor users will get access at least to advanced Kermit protocols, via the Kermit program, in the next year or so.

#78 10 13 Sep 87 12:52:59  
From: Franz Hirner  
To: Roger Folsom  
Subj: PROCOMM

PROCOMM will not run under emulation on the V9. Emulation, at this point, does not handle calls to port addresses. There is something in the works though that might just handle these. Meantime, Crosstalk or Vicmini are your two best bets unless you like KERMIT. There is a new Kermit (Version 229) out. Keep on me and we may have it posted next week.

#93 9 14 Sep 87 05:51:17 (RECV'D)

From: John Navas  
To: Franz Hirner  
Subj: Crosstalk

ProComm can only handle a single WHEN at a time -- I wouldn't be surprised if Crosstalk is the same.

In ProComm, RGET can be used to capture the next incoming line into a string variable. Then FIND can be used to search the string variable for a text pattern. Multiple FIND commands can be used, so it's possible to search for multiple patterns in each line. I had thought I remembered that Crosstalk has a similar capability, but it has been a long time since I gratefully retired Crosstalk and switched to ProComm.

#100 6 09 Sep 87 19:05:36

From: John Knobelsdorf  
To: Lloyd Leblanc  
Subj: Spellbinder

SPELLBINDER DESKTOP PUBLISHER, Version 6.2 was released in late 1986 by Lexissoft, Inc., P.O. Box 1950, Davis, CA 95617, phone 916/ 758-3630 and runs on IBM compatible computers. To my knowledge there is no version for the V9000 and Lexissoft's sales manager, Beth Haynes, said that there would probably never be a V9000 version written.

V9000 users of SPELLBINDER can update from their current version to the IBM Version 6.2 for about \$280.00. I have updated to version 6.2 but have not used it yet. I have seen Beth Hanes demonstrate it and it was impressive.

Beth Hanes said that Al Bullock was the beta test site for the IBM version 6.2 and it worked well on the V9000 with a +PC installed and it worked well on the VI. Al Bullock would be a good contact for information concerning this program. When I learn to use it, I will put something on the boards and, hopefully, write an article for VictorTalk.

Mary Jo.

#112 4 10 Sep 87 08:45:14

From: Lloyd Leblanc  
To: Mary Jo/john Knobelsdorf  
Subj: SB INFO

Well, thanks very much John and Mary Jo!!! I was a little behind although I still get Lexissoft's brochures and releases. I know Al B. well and will talk to him about the 6.2 release and PlusPC/VI usage. That would be nice to test at least. We do have SB 5.4 for the Victor but it is still lacking a few mandatory features, in our view. Might try this new one on the PlusPC, though.

\* \* \*

## RE: USERS GROUP UPDATE

A note from Don Wacker states that he is still contact for the Puget Sound Victor Users Group, but "we moved from Mercer Island". Address for PSVUG is now:

Puget Sound Victor Users Group

Don Wacker  
12201 206th Place SE  
Issaquah, WA 98027  
(206) 255-0533

\* \* \*

## VICTOR 9000 PARTS

256K RAM Expansion Board	(101690-01)	\$35
384K RAM Expansion Board	(101070-01)	60
DMA Boards	(101160-01)	45
10MB External Hard Disk	with DMA board	225
" " " "	without DMA board	185

- All parts guaranteed for 30 days.
- Prices and inventory limited to stock on hand.
- All orders shipped C.O.D.
- Free shipping on pre-paid orders.
- Call or write if you have requests for other V9000 parts.

Gary Tyler  
P.O. Box 3014  
Santa Cruz, CA 95063  
(408)-427-1063

\* \* \*

## V9000 AND VI HARDWARE AND SOFTWARE

Al Bullock, of Bullock and Associates [39522 Benavente Place, Fremont, CA 94539 (415) 651-6183], currently has available the following hardware and software, for Victor 9000 and Victor VI computers, to user group members at the prices indicated.

VI	Floppy Test Software	\$50
VI	Technical Reference Manual	75
VT-CACHE	Cache Software	100
V9000	10MB External Hard Disk (Graymatter)	300
V9000	20MB " " " " "	400
V9000	15MB " " " (Sunol)	400
V9000	Plus PC Kits with .26 ROMs	495
	Brooklyn Bridge with Cable	200
	FireFly File Transfer with Cable	75

\* \* \*

## CLEANING THE VICTOR 9000 SCREEN

/ R. W. Johnson

[The author provides his brief, but perhaps quite adequate, recipe for cleaning the Victor screen. For two other versions, see VictorTalk Vol. 5, No. 5, pages 12 and 13. - Ed.]

I recently tried a Texwipe free sample on my Victor 9000 anti-glare screen and was dismayed to find it gooped up the screen and made it very hard to read. I disconnected the CRT box from the CPU box, and removed the two Phillips screws underneath in the front, which released the main plastic case. Upon loosening the four screws on the side panels, the front bezel cover came loose and disclosed that the anti-glare screen is actually a very fine plastic screen formed against the CRT face by the bezel. I carefully removed the screen and used a suction vacuum on it to get out a lot of the 4-year accumulation of dust, then washed it carefully in clean water and blotted it dry. There was still trapped water which I dislodged by blowing it gently with warm (not hot) air from a hair dryer.

Holding up the screen to the light now disclosed it to be completely clear. My attention then went to the CRT tube surface, which was also dirty and stained. A good rubbing with Kleenex and Windex got it very clean once more. Putting the whole thing back together took only a few minutes.

It is hard to describe the difference. The screen looks like a brand new one now and it is clear and sharp. This kind of cleaning is well worth the trouble, because the Sun-Flex oiled cloth furnished by Victor does not do a very good job. But be very careful of the anti-glare screen - it appears to be very fragile!!

\* \* \*

### VictorTalk

**Address inquiries to:** P. O. Box 50893  
Palo Alto, CA 94303

**Subscriptions:** Nancy Weaver

**Editor:** Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

**Contributors:** Al Bullock, Jan Ewing, Roger Folsom, R. W. Johnson, Mary Jo Knobelsdorf, Jack Prichard, Andrew Rudiak.

VictorTalk is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in VictorTalk. Copyright 1988 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

VictorTalk is an independent publication not affiliated in any way with Victor Technologies, Inc. Victor is a registered trademark of Victor Technologies, Inc.

**Submit Articles** on MSDOS disks in ASCII or Spellbinder format, with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

## Yes, I want to join.

Enclosed is a check/money order for \$35.00  
(foreign membership, \$40.00 in U.S. dollars)  
for my 1988 Victor\*Group membership.

Name \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

Make checks payable to:

## VictorTalk

P. O. Box 50893  
Palo Alto, CA 94303

City \_\_\_\_\_

State \_\_\_\_\_ ZIP \_\_\_\_\_

Country \_\_\_\_\_

Phone \_\_\_\_\_

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
P.O. Box 50893  
Palo Alto, CA 94303

Date \_\_\_\_\_ Office use only:  
 Name \_\_\_\_\_ Order rec'd \_\_\_\_\_  
 Company \_\_\_\_\_ Member check \_\_\_\_\_  
 Address \_\_\_\_\_ Comments \_\_\_\_\_  
 City \_\_\_\_\_ State \_\_\_\_\_ & Zip \_\_\_\_\_  
 Country \_\_\_\_\_ Phone \_\_\_\_\_ Order shipped \_\_\_\_\_

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No. Disks	Vol. No. Disks	Vol. No. Disks	Vol. No. Disks
4 2 61-64	4 6 72-77	5 2 82-84	5 5 87-90
4 3 65-71	5 1 78-81	5 4 85-86	5 6 91-92

- 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
 21 22 23 24 25 26 27 28 29 30 31 32 33 34\* 35\* 36 37 38 39 40  
 \*(34 & 35 are a two-disk set)  
 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56\* 57\* -- 59 60  
 \*(56 & 57 are a two-disk set)  
 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80  
 81 82 83 84 85 86 87 88 89 90 91 92 \_ \_ \_ \_ \_

1988 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_  
 Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_  
 Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_  
 Postage/handling (applies to disk orders only)..... \$ 4.00  
 Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_  
 VictorTalk back issues: 1986, 1987, sets of 6, \$12.00 /set ..... \$ \_\_\_\_\_  
 1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_  
 1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_  
 1988 membership fee \$35 (receive all 1988 issues of VictorTalk) .. \$ \_\_\_\_\_  
 (Foreign membership, \$40 in U.S. funds.)  
 TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders).. \$ \_\_\_\_\_

# VICTOR\*GROUP

NOTE OUR NEW ADDRESS:  
VICTOR \* GROUP  
1493 BEACH PARK BLVD., #180  
FOSTER CITY, CA 94404



## FIRST CLASS MAIL

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
State City _____ & Zip _____	_____
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No. Disks	Vol. No. Disks	Vol. No. Disks
4 2 61-64	5 1 78-81	5 5 87-90
4 3 65-71	5 2 82-84	5 6 91-92
4 6 72-77	5 4 85-86	6 2 93-95

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40
														*(34 & 35 are a two-disk set)					
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56*	57*	--	59	60
														*(56 & 57 are a two-disk set)					
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	_____	_____	_____	_____	_____	_____	_____	_____

1988 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, sets of 6, \$12.00 /set ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_

1988 membership fee \$35 (receive all 1988 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR<sup>®</sup> TALK

Bi-Monthly Magazine for Victor 9000 users

VictorTalk, Vol. 6, No. 2, Mar-Apr 1988

## EXPLORING MS-DOS / George Anderson

Any computer to which a disk is attached must have a program or set of programs to define and control the interface between that computer and the disk, whether a hard disk or a floppy. Those programs, collectively, are called a disk operating system, or DOS. This series of articles discusses MS-DOS, the most popular DOS program to run on machines using the Intel 80xx series CPU's (such as the V9, IBM PC's and their clones). In the first article of the series, four programs will be discussed:

CONFIG.SYS  
CONFIG.BAT  
AUTOEXEC.BAT  
ANSI.EXE

These programs are somewhat special, because they provide a means of defining how your computer is organized when first booted up. Because these programs are modifiable by the user, there is an opportunity to modify its operating personality to better suit your tastes. The second article in the series will discuss COMMAND.COM and the internal commands included in MS-DOS. The third article will cover the external commands in DOS -- the .COM and .EXE programs that are contained on your DOS disk. The fourth article will discuss batch files, and what you can do with them to create your own 'integrated' programs from separate programs you already have, and to turn over to the computer tasks you always felt a computer should be doing anyway. You might even have had a sneaking expectation that you were buying the capability to do the 'boring' stuff when you first bought your computer. Now, granted that a V9 can do more than other computers - and better, to boot - still, batch files can make even a V9 more productive and fun to work with, as unlikely as that may sound! Let's start by taking a fresh look at setting up our computer all over again, and examine the possibilities we might invoke on it. First, a few comments on how computers and DOS relate.

MS-DOS provides the interface and control between the computer and the various peripherals which may be attached to it. Examples of the peripherals it controls are: keyboards, monitors, a mouse, disks (both hard and floppy), digitizing pad, printers, tape drives and other less often used devices.

DOS also provides two other services for the user: it enables the user to load and to execute application programs, and it provides the format and maintenance facilities for placing files on disks and for maintaining them.

Although it is well known that different types of disks store different amounts of information, you may not know that all storage devices store data in the same basic way. Data is sent to the storage peripheral as a stream of groups or bytes, with each byte containing eight bits. Each bit is coded to represent either a one or a zero, much like a light switch being alternately turned on (a one) or off (a zero).

Now, it turns out that eight bits contains enough coding to represent a character which we, as users, want to be able to call from the computer. Thus, the computer stores information by character, but does so in bit form; i.e., a group of eight ones and zeros forms a byte, and this is the basic stored element, representing a single character. A group of bytes, or characters, forms the content of a file, which may be a word processing document, a spreadsheet file, a drawing or other type of document.

For example, an application program is stored in a program file, but the data generated through use of the application program is stored in a data file. The difference in their behavior when in the computer, lies in the different significance which has been preassigned to certain character codes or strings of character codes. It is DOS's responsibility to interpret these codes according to their preassigned values, and to set the computer to react to COMMANDS, but to store DATA in whatever way is specified by the COMMANDS you feed the computer.

Not only does DOS store the generated data on disks or other storage media, it also controls the organization of that storage on the disk, so that quick and efficient transfers of the stored data may be accomplished when called for.

If you are satisfied with the way your computer is working for you, you desire nothing different, and you never are faced with a problem such as loss of data for some reason, then you might consider DOS a necessary evil. If, however, something does occasionally go wrong, or there are certain things which you would like to change for your convenience and efficiency, then DOS can be a great friend. Let's start by exploring CONFIG.SYS.

### CONFIG.SYS

The CONFIG.SYS file is a file that specifies hardware connections and definitions for use, some initial screen formatting information, and allows specification of device drivers to give you further flexibility in arranging the way your computer will respond to you. When your computer is

booted, there is a small program in ROM BIOS that gets the whole operation going. Almost immediately it turns control over to DOS, which is stored in program files on disk. If you use floppy disks only, the ROM BIOS program looks first on the A: drive for DOS, then on the B: drive if DOS is not found on the A: drive. A similar search is performed if a hard disk is involved, and if DOS is contained on the hard disk. However, in this instance, the B: drive is not searched. Once DOS files are found, the program searches for a CONFIG.SYS file; and, if found, proceeds to read the instructions there that modify the default startup configuration predefined by DOS. The V9 uses this file to specify the drivers for two serial (RS-232) ports, and for a single parallel port. These must be specified in the CONFIG.SYS file on the V9, because these ports are addressed through software, and not through hardwired address lines, as in the IBM PC.

EXAMPLE: To insert a port A serial driver in the CONFIG.SYS file, add this line:

```
DEVICE=PORTA.EXE baud(1200) bits(8) stop(1)
parity(none)
```

where:

- o PORTA is the filename for the driver file to set up serial port A;
- o BAUD(1200) specifies the speed of transmission (or number of characters per second) through the port - this could range from 10 to 19,200 baud, depending on the equipment connected to the port;
- o BITS(8) refers to the number of bits that represents a character. This could be either 7 (if the character is ASCII) or 8 (if the character is binary) today, most connections call for 8 bits (binary), though some services like Compuserve may require 7;
- o STOP(1) refers to the number of bits that are used to determine unambiguously when a stop occurs. There are three conventions - 1, 1.5 or 2. All three numbers may be used with ASCII (7 data bits) though 2 is the most common, but 1 is used with binary (8 data bits);
- o PARITY(none) is one of three conventions used. They are ODD, EVEN or NONE. All three may be used on ASCII 7-bit data transmissions, but none is the only choice for 8-bit binary transmissions. Whether to use odd or even parity is an agreed-on convention between the sender and the receiver, using the ASCII data convention. Odd means that the 7 binary character digits (1's and 0's)

PLUS an eighth bit will always add up to an odd value, with the 8th bit used to adjust to an odd value. Even uses the same rules, except that the 8 bits are made to add up to an even value. When transmitting binary 8-bit characters, there is not an extra bit in a byte to provide the make up value. These conventions are used for serial printers and for modems mostly, but apply to other serial devices as well.

The SWITCH character (used in option switches, e.g., /S or /V) is defined and modifiable in CONFIG.SYS, but only in DOS 2.1. This function was deleted in DOS 3.1. In DOS 2.1 you may elect to use either a ^-1 or a !/. DOS's vary on this, but either one or the other is set as the default switch character in your version of DOS. No matter which character it defaults to, the other character is optional, and may be changed to by specifying which one you prefer in the CONFIG.SYS file. An example will show how to do this:

EXAMPLE: To add a switch character to the CONFIG.SYS file, add this line:

```
SWITCHAR=/ (or ^-1 if you picked it)
```

NOTE: Once you select a switch character, it becomes reserved, and must not be used elsewhere, e.g., in filenames, etc.

FILES is a specification in the CONFIG.SYS file that specifies how many files may be open at any one time. And if you are wondering why anyone would want more than one at a time, here's why. DOS defaults to eight files which can be open concurrently. DOS itself takes up two of these (IO.SYS and COMMAND.COM) on the V9, and a third on the IBM PC. Every application program requires an open file, as does every overlay file in a program. With more and more programs large enough to require one or more overlay files, help files, etc. plus use of resident programs such as a RAM disk or even Sidekick, eight files open are soon taken up. To extend the number of files that can be open concurrently, a new number can be specified in the CONFIG.SYS file, as shown in the example file below. The trade off in specifying a large number is that DOS sets aside a small block of memory (48 bytes per open file specified) in which to store and track the status of each open file, so if you specify a larger than necessary number, you will be causing DOS to reserve memory for this status data which will never be used, and that memory cannot be used for anything else. The price is small, but could, in certain instances, become critical. If you should ever run out of memory, and think that a few more bytes



might do the trick, by all means, modify the Files statement (and the next one, Buffers) and try again.

For most uses, it is recommended that you set Files to 15 or 20, since that is large enough to meet the requirements of some of the larger commercial programs. It should be noted, however, that some programs install their own values in the CONFIG.SYS file when they are installed. You can check this by typing the CONFIG.SYS file to the screen and checking its value (TYPE CONFIG.SYS <RETURN>).

EXAMPLE: To add FILES to the CONFIG.SYS file, add this line:

FILES=20 (or a number you pick)

BUFFERS is another specification option you can select. Buffers refers to internal DOS buffers, and each buffer is an assigned block of RAM (512 bytes) into which and out of which data are transferred between computer and disk (or other peripheral). Each block serves as a common point of exchange of data coming in to or going out from the computer. And, since RAM is a form of temporary storage, such a block helps to adjust the differences in speed between the computer and the disk(s) in transferring data. The number of buffers may be selected from 2 to 255, but after a certain point, enlarging the number of buffers specified will only slow down the program being run. The degree of slowing-down will be related to the specific program run, and will occur sooner if the program is one which is memory intensive. Typically, if you are running a word processor program, 10 to 20 buffers provides the best performance. Some rules of thumb are given below, to serve as a starting point:

Basic system	- allocate 2 buffers
Disk space over 360K	- add 3 buffers
128K to 255K bytes of RAM	- add 5 buffers
256K to 511K bytes of RAM	- add 10 buffers
512K bytes or more of RAM	- add 15-20 buffers

Note that, if one of the last three possibilities is selected, then add it to the appropriate first and second choices. You should not feel tied to these numbers, but should vary them experimentally to determine which seems best for you.

EXAMPLE: To add BUFFERS to the CONFIG.SYS file, add this line:

BUFFERS=20 (or the number you pick)

COUNTRY specifies, by a number code, the format for dates and time which will be presented by DOS.

EXAMPLE: To add COUNTRY to the CONFIG.SYS file, add this line:

COUNTRY=1 (for the US - or pick one you want to use, from the codes listed in the V9 DOS Manual)

The CONFIG.SYS file also provides for calling software drivers. Drivers are software programs whose purpose is to handle and direct the operation of specific peripheral devices. DOS defines a peripheral as any entity operating through the I/O of the computer. Therefore, it may be a real hardware device, such as a printer, a disk, or even the CRT; or it may be a software 'virtual' device, such as a RAM disk. We spoke of the special port drivers earlier, which set up the operating parameters of the hardware ports (PORTA, and PORTB for the two serial ports, and PPORT for the parallel port). These are true device drivers, but they are by no means the only ones we might use. For example, we might decide to create a RAM disk in RAM memory to speed up computer operations by storing disk based information in the RAM memory. RAM is much faster to access, so we do not have to wait as long before we can use called data on the screen. To create a RAM disk requires a RAMDISK driver or software program. That program may be referenced in the CONFIG.SYS file, and the RAMDISK will then be available automatically on boot up.

EXAMPLE: To invoke a RAMDISK on boot up, add this line to the CONFIG.SYS file:

DEVICE=VDISK.SYS 128  
or  
DEVICE=RAMDISK.SYS 16

where: 'filename.SYS' is the name of your RAMDISK, and the number given designates the size of the RAMDISK in Kbytes (1 Kbyte = 1024 bytes).

NOTE: These file names are typical of programs included in MS-DOS for the IBM PC and its clones. However, you would need to use a commercial program, and should then follow the instructions given there for installing the program.

BREAK is a toggle function that may be specified in CONFIG.SYS. Its default setting is OFF, but if turned ON, DOS will check more often to see if BREAK had been invoked, and stop the computer. If BREAK is set ON, you will only benefit. Here's how to add it:

EXAMPLE: To set BREAK ON, add the following line to the CONFIG.SYS file:

BREAK=ON

SHELL is a specifier that determines what you choose to use as a COMMAND program. The one furnished with MS-DOS is COMMAND.COM, but others are available that could be used in its place. One reason you might choose another shell would be to have more (or more efficient) resident utility programs you could draw on. However, most of us will choose to use the COMMAND.COM program which came with MS-DOS. The SHELL command may be added this way:

EXAMPLE: To add SHELL to the CONFIG.SYS file, add this line:

```
SHELL=COMMAND.COM /P
```

NOTE: The /P means that this is the default SHELL, and programs, when returning to DOS control, will automatically access this SHELL (i.e. COMMAND.COM). The /P must be separated from COMMAND.COM by at least one space, for the computer to accept it. If COMMAND.COM is not on the current drive, or if there is not a specified path to COMMAND.COM, you will see a message on the screen asking that you insert the disk containing it in the current drive.

Now let us put a CONFIG.SYS file together from the lines we have defined. CONFIG.SYS would then look like this for MS-DOS 2.11:

```
BREAK=ON
FILES=20
BUFFERS=20
COUNTRY=1
SWITCHAR=/
DEVICE=PORTA.EXE BAUD(1200) BITS(8) STOP(1)
    PARITY(NONE)
DEVICE=PORTB.EXE BAUD(1200) BITS(8) STOP(1)
    PARITY(NONE)
DEVICE=PPORT.EXE
SHELL=COMMAND.COM /P
```

NOTE: SWITCHAR=/ would be deleted in a CONFIG.SYS file for MS-DOS 3.1, since it is not supported in that version of DOS.

CONFIG.SYS is an ASCII file used by DOS to store the operating parameters we have chosen to work with at start up. To augment the .SYS file we will be looking at .BAT files, which can give us great flexibility in defining how we want our computer to run. Two special .BAT files are used in the setup of your computer. They are the CONFIG.BAT and AUTOEXEC.BAT files. Both are true batch files, but special treatment is accorded them by COMMAND.COM.

COMMAND.COM looks for these .BAT files automatically. First, CONFIG.SYS is read, then CONFIG.BAT. Finally, AUTOEXEC.BAT is read.

The primary purpose of a CONFIG.BAT FILE is to store our selections for such things as a custom prompt definition, port selection default (for printer connection) and other hardware related selections.

The AUTOEXEC.BAT file differs only in that COMMAND.COM skips the time and date requests on bootup if an AUTOEXEC.BAT file is present. Otherwise it operates just like any other batch file. The next step is to get CONFIG.BAT to select for us the particular hardware we want at start up.

### CONFIG.BAT

CONFIG.BAT is an ASCII file used by DOS to store our selections for the hardware we want to use at start up. For example, we can store a specification for what we want as a prompt here. In addition, we will want to specify the printer port connection we intend to use. These may be the major items we want to store in CONFIG.BAT. To do so, we would proceed as follows:

EXAMPLE: To add a prompt specification which will display both the current drive, and the current directory, add this line:

```
PROMPT $p$g
```

where: \$p produces the current drive and directory, and \$g follows them with a '>' symbol. More will be said about how you may customize your prompt in the section covering ANSI.EXE.

EXAMPLE: To add a parallel printer port specification to the CONFIG.BAT file, add this line:

```
CLST\DEV\PARALLEL
```

where: CLST\DEV states that the default listed device port is specified next. The \PARALLEL specifies that the default port will be the parallel port. If you wish to connect a serial printer to PORT A, the line should read:

```
CLST\DEV\SERIALA
```

where: \SERIALA designates serial port A.

If the first line in any batch file is 'echo off', the subsequent commands do not print on the screen. This is

less confusing to most people, since there is less distraction on the screen. Our example CONFIG.BAT file now should look like this:

EXAMPLE: A sample CONFIG.BAT file:

```
ECHO OFF
CLST\DEV\PARALLEL
VT-ICLK
PROMPT $P$G
```

Where: First, second and fourth lines perform as described above; VT-ICLK invokes the operational availability of the Valid Technology clock module.

ANSI.EXE is a program which allows greater flexibility in the appearance of your screen and the key assignments (a la modcon). A further description is given below. If you want to use ANSI.EXE on a regular basis, it can be included in the CONFIG.BAT file as shown:

EXAMPLE: To use ANSI.EXE, enter this line in the CONFIG.BAT file:

```
ANSI <RETURN>
```

#### AUTOEXEC.BAT

AUTOEXEC.BAT files may profitably contain such a wide variety of commands, that I will defer much of their potential until we discuss the .BAT file generally. However, a sample file might be used to clear the screen, and to provide initial instructions to the operator (no more remembering how to get into that favorite program!). This approach might work very well for you if you are operating with floppy disks exclusively. Here is a sample AUTOEXEC .BAT file:

EXAMPLE: An AUTOEXEC.BAT file:

```
echo off
cls
type AUTO.TXT
```

EXAMPLE: An AUTO.TXT file:

When All Else Fails, Grab a Pencil!

At present, you have two programs to choose from:

1. dBASE II
2. WordStar

To enter dBASE II, insert the program disk in Drive

A:, and insert a working disk in Drive B:. Type 'DBASE' and your program will load and present the prompt '!'

To enter WordStar, insert the program disk in Drive A:, and insert a working disk in Drive B:. Type 'WS' and your program will load and present the Edit No File screen.

BE HAPPY IN YOUR WORK!

In this sample, all the text to be presented on the screen is contained in an ASCII text file called AUTO.TXT. The text file is called from the last line of the AUTOEXEC.BAT file, and prints its message on the screen. We will look at many other command strings that might be used in an AUTOEXEC.BAT file when we look at .BAT files generally. With the knowledge you have gained, you are armed and ready to modify your CONFIG.SYS, CONFIG.BAT, and AUTOEXEC.BAT files to your liking, and to do it with confidence.

#### ANSI.EXE

ANSI.EXE may be considered a device driver, but on the V9, it emulates a device driver and must, therefore, be treated somewhat differently than a true device driver such as PORTA.EXE. (ANSI stands for American National Standards Institute) which emulates the IBM file ANSI.SYS. You may want to use it to modify your screen's appearance, or to change the keyboard. If you have both 2.11 and 3.1 and want to use ANSI.EXE with 2.11, all that is necessary is to copy ANSI.EXE onto the DOS 2.11 disk, and install it as shown:

EXAMPLE: To use ANSI.EXE enter this line at the prompt:

```
ANSI <RETURN>
```

NOTE: ANSI is a terminate-and-stay-resident program. Entering it as shown will make it available until you shut off power on the V9. To invoke it automatically, place the same command (ANSI) in your CONFIG.BAT file. The procedure for this was described earlier in this article, in exploring CONFIG.BAT files.

The ANSI emulator driver allows you to change the default values for both screen and keyboard. For example, you may customize the function keys for use as part of a custom menu selection system you create. Some application programs depend on ANSI.EXE being available to enable them to customize certain keys for that specific program. Often, these programs make ANSI.EXE available by

rewriting the CONFIG.SYS file to include it. It can be instructive to start with a 'bare bones' version of the CONFIG.SYS file, and then to install a program. By looking at the CONFIG.SYS file both before and after installing the program, the changes that were performed by the program installation can be seen. To look at the file, type:

```
TYPE CONFIG.SYS
```

If you want to study it, you can print it out by typing:

```
TYPE CONFIG.SYS>PRN
```

which will send the file to the printer (using DOS 2.1 or higher). You may find that the number of FILES open, or the number of BUFFERS allocated has been changed; and this can help to give you a feel of what a good setting is for your machine when running the type of program you plan to run.

If you use DOS 3.1, screen attributes are easily specified with ANSI.EXE. For example, bold text, blinking text, or text in reverse video are easily accessed, providing you with a variety of ways to emphasize the information you place on the screen.

The attributes DOS offers are:

- 0 All attributes off
- 1 Bold
- 4 Underline
- 5 Blinking
- 7 Reverse Video

Underline only occurs with certain mono monitors, due to the limitations inherent in color monitors.

EXAMPLE: To print your PROMPT in reverse video, precede the message with:

```
PROMPT $e[7m (then the message).
```

To switch back, add:

```
$e[0m
```

The line should then look like this:

```
PROMPT $e[7m(Your message or prompt codes)$e[0m
```

where: PROMPT precedes a string of options defining the DOS prompt; a mandatory space; \$ begins the string; e[ gives the ASCII equivalent of ESCAPE;

7 is the attribute code for reverse video; and 'm' signifies the end of the string. The 'm' must be lower case to be interpreted properly. Similarly, the '0' in the second PROMPT line resets all attributes to off. Note that spaces will show up in the end result exactly as they do in the command line you form. If you leave a space after the \$e[7m code, that space will show up in the final effect on screen. I use a prompt formed as follows:

```
PROMPT $e[7m $p$g $e[0m
```

which gives a prompt in reverse video showing the current drive and the current directory. There is a reverse video space both before and after the drive and directory, which doesn't look quite so chopped off as it would if those spaces were not there.

Meta Symbols are symbols that replace symbols. They 'stand in place of' default symbols, and are used to specify the elements included in the PROMPT command line. This can be very useful in customizing your screen, and allows phrases to be incorporated in your prompt, such as:

EXAMPLE: To incorporate a reverse video phrase in your prompt, type the following:

```
PROMPT $e[7mYour Wish is My Command!$g$e[0m
```

The prompt then becomes:

```
Your Wish is My Command!> (in reverse video)
```

The Meta Symbols that DOS offers are as follows:

SYMBOL	MEANING
=====	=====
e	The ESC key
p	Current directory
g	The > character
n	Default drive name
d	System date
t	System time
v	Version of DOS
l	The < character
b	The ; character
q	The = character
h	A backspace
—	The carriage return/line feed sequence

These are the symbols you use (preceded by a '!') to modify your screen, as shown in the examples above.

Another version of a prompt might use some more of the symbols above, to include the time and the date. It might look like this:

EXAMPLE: To design a prompt in reverse video, with current drive and directory, date and time, use the following:

```
PROMPT $e[7m $p $d $t $e[0m$ $g
```

The prompt might then read (in reverse video):

```
A:\1DIRPLUS Sun 3-20-1988 18:14:46.32
```

where the drive and directory are separated from the date and time by the spaces included in the command line.

If you want to include your new prompt on a permanent basis, you may place the following in your CONFIG.BAT or AUTOEXEC.BAT file:

EXAMPLE: To place the prompt giving the current drive and directory in either the CONFIG.BAT or AUTOEXEC.BAT file, add the following line:

```
PROMPT $P$G
```

This will change the prompt from the default prompt automatically.

Next time we will look at internal commands in MS-DOS, those that are resident in RAM when DOS is running.

\* \* \*

---

## SPREADSHEET NEWS

/ Dan Mardesich

For those Victor users who do not have a spread sheet and want one, they can now have one of the best for a very low price. Report Manager is now available in either V9000 or IBM format for less than fifty dollars.

Back in the old days when the 9000 was new, Victor packaged a program for their new computer called "Victor-Calc." It came in one of those silver binders with the tilting black slip case and matched the other software in the Victor line. This was a 3-D spread sheet that had pages in addition to the rows and columns of other spread sheets, and having 256 each of pages, rows, and columns, was probably the largest available at the time. It was an OEM product, meaning that Victor bought it from another manufacturer and put their own label on it. They did not do that with WordStar or dBase II, but they did with other

products, including the Diablo 630 printer which they called the Victor 6040.

When Datamension, the maker of Victor-Calc, upgraded their product, they offered it under their own label: Report Manager. They subsequently upgraded the upgrade until the final version was 2.5av (standing for "Advanced Version"). Victor-Calc originally sold for \$295. The upgrade to Report Manager v.2.0 was \$200 (bringing it to the retail price for Lotus 1-2-3). The subsequent upgrade to 2.5av, as I recall, was another \$125.

It was a wonderful product. It won an award from one of the magazines as the best software of 1984. It was more flexible than 1-2-3, permitted larger spread sheets and allowed the easy consolidation of spread sheets; it had an extensive MACRO language and could even do rudimentary charts. In spite of all its good qualities, like the V9000, it was supplanted by an inferior product which became the "industry standard."

Now that we are generations beyond that time and software has more and more bells and louder and louder whistles, prices for the "old" stuff have come down.

Those who have registered copies of Report Manager have probably received a catalog from something called "Share House." This is a marketing venture by Datamension that has a kind of pyramid club or Amway-type structure in which you "join" when you buy something and you receive "commissions" on things that you convince others to buy. This gimmick may or may not interest you, but the important event is that you can now purchase Report Manager v.2.5av for \$49.75. That is for the software alone. If you want the manual it will cost you another \$49.75 for "The Report Manager Book", which includes "a disk of sample spread sheets."

This is an excellent deal for Victor users looking for a spread sheet. For those, like me, who have been forced to downgrade to the "industry standard" computer, it allows you to use the same templates you have worked out without having to reinvent them. (This assumes you have the means to transfer the files to the other format.) The IBM version has color and on-line help of sorts and seems to support all the things I did with it on my Victor.

For more information, or to order, call:  
Share House  
800-843-2823  
(In Illinois, call 312-564-5625)

If you decide to buy anything, my number is 001062!

\* \* \*

# CHARACTER SETS / Jack Prichard

One of the many desirable features of the Victor 9000 computer is the ability to easily change to any of a large number of character sets for display on the monitor screen. The character sets reside in files with the extension .CHR, a large number of which are included in the Public Domain Library. The purpose of this article is serve as an index to these files. Fortunately, all such files have a common extension, .CHR. This is necessary for them to be properly processed by MODCON, the program that installs them into the operating system. Hence, they are easily identified.

When the Victor 9000 computer is booted, the character set normally installed is the one identified as "International Character Set 01" on the legend appearing during bootup and in the various users' guides and reference manuals. (VINTL01.CHR and STANDARD.CHR in Table 1.) Other character sets are contained in the SYSELECT disk included in the Programmer's Tool Kit, in the graphics character set provided with CHARGRAF in the Graphics Tool Kit, and in many applications programs. These other character sets are not covered in this article.

Several different versions of MODCON are available, although they all appear to perform the same functions. In MS-DOS 1.25, MODCON was not available with the original set of programs supplied with the Victor 9000 computer; but, it was available both as part of the Programmer's Tool Kit and of the Graphics Tool Kit. MODCON.EXE (14,738 bytes 2/7/84 09:12) is included as part of both MS-DOS 2.11 and 3.1. Two earlier and longer versions of MODCON.EXE are available in the Public Domain Library as follows:

```
MODCON.EXE 23424 10/4/82 10:11 PD 11, 12, 28, 48.  
MODCON.EXE 24832 1/21/83 10:53 PD 5, 55, 59.
```

Some of the details of .CHR files are described in the Graphics Tool Kit under the program EFONT. This program provides for viewing the .CHR file header and for viewing a character set in detail to edit it or create a new set. Briefly, a .CHR file consists of a header (with either 128 or 256 bytes allocated) and a character file of either 128 characters (character set 0) or 256 characters (character sets 0 and 1). Each character is allocated a 16 x 16 array of pixels (picture elements or dots); thus, each character requires 32 bytes to represent it. The total characters sets, thus, occupy either 4,096 bytes or 8,192 bytes. Hence, the total .CHR files are of one of four sizes: 4224, 4352, 8320, or 8448 bytes, depending both on the size of the header and of the character array.

The .CHR files in the Public Domain Library are listed alphabetically and described in Table 1. The procedure used to elicit the information given is as follows:

1. All .CHR files were gathered together on two disks for ease of handling. Files with identical filenames from different public domain disks were separated by prefacing their filenames with the public domain disk number from which they came (and abbreviating as necessary).

2. Table 1 was started as an alphabetical listing of all .CHR files as they appear on the various Public Domain disks (separating the public domain disk number from the filename).

3. Files with identical filenames and date/time were listed in a single entry. The belief that the files were identical was checked using FILCOM (a MS-DOS 2.1 utility). In all cases the files proved identical.

4. Printouts were made of the header of each .CHR file. [If this is attempted using the internal command TYPE or the external command LIST (from MS-DOS 3.1 or from the Public Domain Library), the header is usually clobbered by the data representing the character sets.] As EFONT was not available for this purpose, the following short MS-BASIC program was used:

```
10 REM "CHRHDRP.BAS" 4/4/88  
12 REM Prints header of .CHR files  
20 INPUT "FILE: ", F$  
30 F$=F$+".CHR"  
40 OPEN F$ FOR INPUT AS #1  
50 IF EOF(1) THEN 110  
60 LPRINT F$  
70 FOR N=1 TO 128 'Make 256 when needed  
80 I$=INPUT$(1,#1)  
90 LPRINT I$;  
100 NEXT N  
110 CLOSE: LPRINT: LPRINT  
120 GOTO 20
```

5. The appearance of each possibly unique character set was checked by graphic printout of the complete character set (using the MS-DOS 3.1 utility SCRNMX80 with the printer set to 8/72-inch line spacing). Files having like appearances were then checked for identity, again using FILCOM. It was found that many files, though having different date/times (and in some cases, different filenames) were actually identical. In at least one case, the character

sets were identical, but the header information was slightly different. (These files are called identical for our purposes.)

6. Table 1 was then modified to reflect these discoveries and to quantify the appearance of the character sets, using both a set of symbols and additional remarks to characterize them. The symbols used (in their order of appearance in the column headed "Characteristics" in Table 1) are as follows:

- U - Upper case alphabet (A-Z)
- L - Lower case alphabet (a-z)
- N - Numbers (0-9)
- P - Punctuation marks (at least , ; : . ! ? ' " )
- M - Mathematical symbols (at least + - \* / =)
- F - Foreign letters added
- G - Miscellaneous symbols (such as ( ) [ ] < > @ # \$ % & \) and/or graphics
- V - Vertical (letters running up the page, rather than horizontally, left to right)
- I - Inverse (white on black printout)
- T - Thin (single dot line width)
- X - Flawed characters (some missing from normal set or missing parts)
- H - Height (given in dots)
- \* - Suitable for general use (complete, unflawed, upper and lower case, American alphanumeric set with normal punctuation marks and math. symbols, etc., displaying and printing horizontally).

7. Consideration was given to illustrating each unique character set in Table 1 with a printed array of the 128 or 256 characters; however, this was deemed impractical. (More than 12 pages would be required.) Four such character sets (identified as International, British Standard, French Standard, and German Standard) are illustrated in appendices of various of the Victor documentation.

#### OBSERVATIONS

Some observations about the various character sets:

1. Many files are duplicated, some with different date/times of latest revision, some with different filenames.
2. Most files contain only character set 0 (4224 or 4352 bytes in length); a small number contain both character sets 0 and 1 (8320 or 8448 bytes).
3. Most files have an essentially complete set of both upper and lower case alphabetic characters, numbers, punctuation marks, mathematical and various other symbols.

4. Other than an expanded (but generally not very useful) name for the character set, the headings provide little useful information. They do indicate that: Most files were originated by Victor or Sirius Systems, although a few bear the names of other originators. Most files were originated on 7/16/82 or 7/17/82, although a few were originated more recently; some bear the suspect date of 1/1/82. [Many files incorrectly show the latest revision date as 1/1/80 (because the computer date was not set)].

5. While most file names give some clue to the type of character set, some are cryptic and others misleading. Some clarifications follow:

- o SAVE.CHR 11 71 (though bearing different date/times of latest revision) are identical to VINTL01.CHR 05 55 and STANDARD.CHR 48. [SAVE.CHR is actually the (temporary) filename suggested in the reference manual for saving a .CHR file that is to be later restored to use by MODCON.]
- o Many files have the same, or essentially the same, character set 0 as VINTL01.CHR, etc.
- o SMALL or SM, MEDIUM or MED, and LARGE or LG in the filename refers to the character height (normally 7, 9, or 12 dots high, respectively). However, character heights of 7, 9, 10, 11, 12, 15, and 16 dots appear in the H column under "Characteristics" in Table 1.
- o All files including VERT or a final V in their filename (with the exception of VERT3.CHR 71) are vertical character sets, as is VPROP.CHR 71. (The V in VINTL01.CHR apparently stands for Victor.)
- o While all character sets can be displayed in inverted video (white on black print), several files are already inverted, or partially inverted (indicated by I or (I), respectively, in Characteristics column of Table 1).
- o Files including THIN or THN in their filename have character lines only one dot wide, as do some files without this clue. (See T in Characteristics column of Table 1.)
- o Files including PROP or PRP in their filename have supposedly proportionally sized widths, although many appear the same as the fixed width characters. (Some files are actually identical.)
- o BOLD.CHR 71 does not display or print bold characters; it is blank except for one character!
- o VERT3.CHR 71 is similarly blank, except for one character!
- o VICI.CHR 71 is only a means of displaying a "vicki" logo and is otherwise blank.
- o PROP.CHR 71, THINPROP.CHR 80, THINVERT 80, and THNPROPV 80 have been identified as missing lower case characters x, y, and z.

TABLE 1. CHARACTER FILES (Page 1 of 4)

Filename (.CHR)	Size	Last Change	PD No.	Identical File Similar File (S)	Characteristics ULNPMFGVITX H *	Remarks (NOTES at end of Table)
AUDIO	8320	1/01/80 00:01	85	-	ULNPM G(1) 7 *	00-1F appears ident. STD; set 0 alphanumerics appear ident. STRETCH 71; set 1 inverted video UC LC & shaddowed alphanumeric sets
BOLD	4352	1/01/80 00:29	71	-	X -	All blanks but 1 char.; hdg. blank
CENTI	8320	1/01/80 00:01	82 84	-	U N G I 10 -	Inverted video; set 0 includes spc. symbols; set 1 spc. legends
CONTROL	8320	8/02/86 13:16	92	-	ULNPM G *	Includes italics in 4 UC & 2 LC sets
DECO	8448	6/11/85 18:37	66	-	ULNPMFG 10 *	Set 1 like STD
DEF	4224	1/01/80 00:00	12 48	-	UL FG 11 -	00-39 symbols
EG	4352	11/01/83 18:41	71	GAMES 23 24 S	ULNPM 11 -	Mixed font in upper case
FUTURE	4224	10/18/82 17:07	71 80	-	ULNPM G 11 -	Legability poor; UC & LC same mixed font
FUTUREV	4224	7/06/84 12:02	80	-	ULNPM GV -	Legability poor; UC & LC same mixed font
GAMES	8320	1/01/80 00:03	23 24	EG 71 S SAVE 24	ULNPMFG 11 *	Sim. to STD, but 00-0F & 80-86 reshuffled; ident. SAVE 24 except hdg.
GAMESBAS	4224	10/03/84 15:43	48	-	ULNPM G 11 *	Omits some symbols
GERMAN	4224	10/12/82 17:03	71	-	ULNPMFG T 12 *	00-1F small UC with caret
GRAFPK	8320	1/01/80 00:01	80	-	ULNPMFG 10 *	00-9F & E0-FF sim. to STD, except height; A0-B2 & C0-DE spec. graphics
GRAPHICS	8320	7/06/84 12:06	80	-	FG 16 -	Set 1 entirely blank; 00-1F are STD 80-9F; remaining char. spec. graphics
GREEK	4352	3/01/85 13:49	71	-	- F -	Mostly Greek alphabet, superscript numbers; 20 not blank
GREEK	4352	10/13/82 13:31	80	-	- F -	Mostly Greek alphabet, superscript & subscript numbers; 20 not blank
GREEKV	4224	7/06/84 11:58	80	-	F V -	Mostly Greek alphabet, superscript numbers; 20 not blank
GW	8320	6/21/83 10:52	11 29	-	ULNPMFG X11 *	Only F0-FF dif. from STD
GW	4224	1/01/80 14:11	71	-	ULNPM G 7 *	
HEBREW	8320	4/18/83 08:08	87	-	ULNPMFG 11 *	00-7F same as STD; set 1 Hebrew & other
INVEXEC	8320	1/01/80 00:02	05 55	-	ULNPMFG I 11 *	Same as STD but inverted video



TABLE 1. CHARACTER FILES (Page 2 of 4)

Filename (.CHR)	Size	Last Change	PD No.	Identical File Similar File (S)	Characteristics ULNPMFGVITX H *	Remarks (NOTES at end of Table)
KTG	8320	1/01/80 00:02	71	-	ULNPM G 10 *	00-1F & set 1 spec. graphics for Knee Toe Graphics; 20-7F sim. STD
LARGE	4352	9/30/84 09:57	80	LGPROP 71 80 S	ULNPM G 12 *	Includes partial small UC with caret
LGPROP	4352	10/12/82 17:05	71 80	LARGE 80 S	ULNPM G 12 *	Includes partial small UC with caret
LGPROPV	4352	7/06/84 11:39	80	-	ULNPM GV -	Includes partial small UC with caret
LGVERT	4224	7/27/83 09:55	29 30	LGVERT 71 LGVERT 80 S	ULNPM GV -	
LGVERT	4224	10/12/82 17:05	71	LGVERT 29 30 LGVERT 80 S	ULNPM GV -	
LGVERT	4352	9/30/84 10:01	80	LGVERT 29 30 S LGVERT 71 S	ULNPM GV -	
MED	4224	7/27/83 09:55	29 30	MED 80 USER1 80	ULNPM G 9 *	02-1F blank
MED	4224	1/01/80 00:01	71	-	ULNPM G 9 *	Few char. diff. from MED 29 30; 02-1F blank
MED	4224	10/12/82 17:06	80	MED 29 30 USER1 80	ULNPM G 9 *	02-1F blank
MEDIUM	4224	10/01/82 14:02	71	-	ULNPM G 9 *	Appears ident. to MED 80 & USER1 80 except 01 blank
MEDPROP	4352	1/01/80 00:05	71	MEDPROP 80	ULNPM G T 9 -	00-1F blank; proportional
MEDPROP	4352	9/30/84 10:28	80	MEDPROP 71	ULNPM G T 9 -	00-1F blank; proportional
MEDPROPV	4352	7/06/84 11:44	80	-	ULNPM GV T - -	00-1F blank; proportional
MEDVERT	4224	7/27/83 09:55	29 30	MEDVERT 71 80	ULNPM GV T - -	00-1F blank
MEDVERT	4224	10/12/82 17:05	71 80	MEDVERT 29 30	ULNPM GV T - -	00-1F blank
NORMAL	4224	7/27/83 09:55	29 30	NORMAL 71	ULNPM G 11 *	Same as STD set 0
NORMAL	4224	10/12/82 17:05	71 92	NORMAL 29 30	ULNPM G 11 *	Same as STD set 0
NORMPROP	4352	9/30/84 10:22	80	NORMAL 29 30 S NORMAL 71 S	ULNPM G 11 *	Appears same as STD set 0
NORMPRPV	4352	9/30/84 10:27	80	NORMVERT 80 S	ULNPM GV -	Proportional?
NORMVERT	4224	9/30/84 10:08	80	NORMPRPV 80 S	ULNPM GV -	
OCR	4352	7/27/83 09:55	29 30	OCR 71 80	U NPM G 12 *	No LC (same as UC); 00-1F small UC with caret
OCR	4352	10/12/82 17:04	71 80	OCR 29 30	U NPM G 12 *	No LC (same as UC); 00-1F small UC with caret
OLD	4352	1/01/80 00:30	71	-	ULNPM G 16 *	Old English; numbers 2-9 dif. font (STD)
ORATOR2	8320	9/01/85 10:44	66	-	ULNPM G 15 *	00-5F like STD, but 13 high; 60-7F small UC (9 high), rather than LC; set 1 (except 2) same as STD
PAC	4224	1/01/80 00:12	12 48	PAC1 12 48 S PACMAN 12 48 S ROB 12 48 S	ULNPM G 11 *	Sim. to STD, but few unique char. added 20-29

TABLE 1. CHARACTER FILES (Page 3 of 4)

Filename (.CHR)	Size	Last Change	PD No.	Identical File Similar File (S)	Characteristics ULNPMFGVITX H *	Remarks (NOTES at end of Table)
PAC1	4224	1/01/80 00:18	12 48	PAC 12 48 S PACMAN 12 48 S ROB 12 48 S	ULNPM G 11 *	Sim. to STD, but few unique char. added 20-2B
PACMAN	4224	1/01/80 00:02	12 48	PAC 12 48 S PAC1 12 48 S ROB 12 48 S	ULNPM G 11 *	Sim. to STD, but few unique char. added 20-29
PROP	4352	10/12/82 17:03	71	THINPROP 80	ULNPM G T 12 -	00-1F small UC with caret; xyz missing at 78-7A
ROB	4224	1/01/80 00:11	12 48	PAC 12 48 S PAC1 12 48 S PACMAN 12 48 S	ULNPM G 11 *	Sim. to STD, but few unique char. added 20-29
ROMAN	4224	1/01/80 00:02	71	ROMPROP 71	ULNPM G 9 *	
ROMPROP	4224	1/01/80 00:02	71	ROMAN 71	ULNPM G 9 *	
SANSERIF	8320	9/01/85 10:46	66	-	ULNPMFG 11 *	Set 0 like STD except no serifs; set 1 like STD
SAVE	8320	1/01/80 00:01	11	SAVE 71 STANDARD 48 VINTL01 05 55	ULNPMFG 11 *	Identical to STD
SAVE	8320	1/01/80 00:12	24	GAMES 23 24	ULNPMFG 11 *	Ident. GAMES 23 24, except hdr. dif.
SAVE	8320	3/12/86 11:59	71	SAVE 11 STANDARD 71 VINTL01 05 55	ULNPMFG 11 *	Identical to STD
SCR	4224	1/01/80 01:20	12 48	-	U N G 11 -	Mostly symbols; limited punctuation; no LC
SCRIPT	4224	10/12/82 17:03	71 80	-	ULNPM G 12 *	00-1F small UC with caret
SCRIPTV	4224	7/06/84 11:52	80	-	ULNPM G TX *	00-1F small UC with caret
SHADES	4224	1/01/80 00:09	71	-	- G 16 -	All spec. graphics for Knee Toe Graphics
SHARK	8320	12/13/87 20:41	94	-	ULNPM G 10 *	Set 0 nearly ident. GRAFFAK 80; set 1 like STD, except 80-8A, 90-9A, & FF
SMALL	4352	7/27/83 09:55	29 30	SMALL 44 45 80 USER2 80	ULNPM G T 7 *	
SMALL	4352	10/12/82 17:06	44 45 80 92	SMALL 29 30 USER2 80	ULNPM G T 7 *	
SMALL	4352	1/01/80 00:55	71	-	ULNPM G T 7 *	Appears ident. SMALL 29 30 44 45 80, but many FILCOM dif.
SMPROP	4352	10/12/82 17:04	71 80	-	ULNPM G T 7 *	Proportional
SMPROPV	4352	10/12/82 17:06	80	VPROP 71	ULNPM GV T -	
SMVERT	4224	7/27/83 09:55	29 30	SMVERT 71 80	ULNPM GV T -	
SMVERT	4224	10/12/82 17:06	71 80 92	SMVERT 29 30	ULNPM GV T -	
SP	8320	1/01/80 00:00	72	-	ULNPM G 10 *	Set 0 sim. SANSERIF; 91-FF sim. STD with few exceptions
SQUARE	8320	12/03/85 21:23	72	-	ULNPMFG 11 *	Set 0 somewhat normal; set 1 spec. symbols & graphics

TABLE 1. CHARACTER FILES (Page 4 of 4)

Filename (.CHR)	Size	Last Change	PD No.	Identical File Similar File (S)	Characteristics ULNPMFGVITX H *	Remarks (NOTES at end of Table)
STANDARD	8320	10/01/84 17:13	48	SAVE 11 71 VINTL01 05 55	ULNPMFG 11 *	(Referred to as STD in Remarks column)
STAR	4224	1/01/80 00:11	12 48	-	- G 16 -	Entirely spec. symbols; 20 not blank
STRETCH	4224	1/01/80 01:06	71	-	ULNPM G 7 *	00-1F small UC with caret
THIN	4224	10/12/82 17:05	71 80	USER3 80	ULNPM G T 11 *	00-1F small UC with caret
THINPROP	4352	10/12/82 17:03	80	PROP 71	ULNPM G TX12 -	00-1F small UC with caret; xyz missing at 78-7A
THINVERT	4352	6/23/84 11:17	80	-	ULNPM GV TX -	00-1F small UC with caret; xyz missing at 78-7A
THNPROPV	4352	7/06/84 11:48	80	-	ULNPM GV TX12 -	00-1F small UC with caret; xyz missing at 78-7A
USER1	4224	10/12/82 17:06	80	MED 29 30 80	ULNPM G 9 *	02-0F blank
USER2	4352	10/12/82 17:06	80	SMALL 29 30 44 SMALL 45 80	ULNPM G T 7 *	
USER3	4224	10/12/82 17:05	80	THIN 71	ULNPM G T 12 *	00-1F small UC with caret
VERT3	4224	10/12/82 17:04	71	-	X -	All blanks except 1 char.
VICKI	4224	1/01/80 00:39	71	-	15?-	Gives only script "vicki" logo
VINTL01	8320	1/01/80 00:21	05 55	SAVE 11 71 STANDARD 48	ULNPMFG 11 *	Normal boot char. set
VPROP	4352	10/12/82 17:06	71	SMPROPV 80	ULNPM GV T -	
XMAS	8320	12/19/85 19:39	66	-	ULNPM G 10 *	00-1F spec. symbols; 20-7F sim. STD; 80-9F blank; A0-FF vert. down reading alphanumeric

NOTES:

- Files are referenced by filename without the .CHR extension and, if needed to avoid ambiguity, by the PD disks on which the file appears.
- Listing within each common filename group is in PD number order, rather than date order, because of many meaningless dates (1/01/80).
- Identical File means exact match with FILCOM, with possible exception of header; Similar File means more extensive FILCOM differences, but very similar appearing character sets.
- Symbols used in Characteristics column are defined in preceding text.
- Abbreviations used in Remarks column:
  - char. = character
  - dif. = different (from) or differences
  - hdr. = header
  - ident. = identical (to)
  - LC = lower case alphabet
  - sim. = similar (to)
  - spec. = special
  - STD = STANDARD.CHR (identical to VINTL01.CHR, etc.)
  - UC = upper case alphabet
- Character positions in their storage array are referenced in hexadecimal notation, 00 to FF, as shown in the character set illustrations in appendices of various Victor documentation.
  - Character set 0 is located at 00-7F; character set 1 at 80-FF.
  - 0-9 are normally located at 30-39; A-Z at 41-5A; a-z at 61-7A.

\* \* \*

## BENCHMARK GRAPHICS: A REVIEW / Roger Nils Folsom

[Written for, and at the request of, TECHNICALLY YOURS  
P.O. Box 1414, Aptos, California 95001 (408) 688-5201]

Benchmark Graphics version 3.0 is an interesting and potentially very useful program. After having worked with it fairly extensively for a week or so, I like it, although its keyboard is somewhat peculiar. I do not have enormous experience with graphics programs, but I have used Tony Casano's Knee-Toe Graphics (now in the public domain), and some years ago made small use of Benchmark Graphics version 1.0, and also of Victor Technologies' BusiGraf (written in GW Basic), from Victor's Graphics Toolkit (either version I or II).

Benchmark Graphics has two major uses. First, it does two-dimensional "free style" drawings using rectangles, circles, ellipses, arcs (apparently a portion of a circle), curves (with curvature based on the initial slope), straight lines of various thicknesses and broken and unbroken patterns, and even a completely freehand draw, together with a variety of "fill" patterns. Second, it does business graphics: the usual bar charts (either horizontal or vertical), stacked bar charts, pie charts, and line plots or x-y plots. To replace Benchmark Graphics one would need both Knee-Toe for "free style" graphics, and BusiGraf for business graphics. Compared with these alternatives, Benchmark has some major advantages and few disadvantages.

Benchmark basically produces only single screen graphics, although there is a facility for putting two screens together to make up a full page.

I reviewed the latest Benchmark Graphics version on the Victor 9000, 3.0a. It is available also on a variety of other MS-DOS machines, not all IBM compatible. The current version is a vast improvement over version 1.0, although from the user's perspective its basic method of operation, discussed below, appears to be the same as in the earlier version.

### Miscellaneous Remarks

Benchmark is menu driven--but you always can bypass the menu (even bypass calling up the menu) and simply type the (three letter) command, if you remember it. The help facility is very complete, as is the manual--except that explanations of how to move, rotate, or zoom (enlarge or contract) images need to be collected in one place instead of scattered throughout the manual. Also, the manual needs much sharper delineations of new major sections and subsections: indenting, and using different size type styles for headings and subheadings, would greatly improve readability. (The manual's designer apparently missed school

the day his junior high English class covered outline presentation, and never made up the work.) The manual does have an index, although it could be more extensive. For the impatient, there is a "Quick Start" tutorial that I found to be an excellent introduction to the program.

One irritating feature of Benchmark Graphics is that it uses the return (or enter) key, not only to enter commands and information, but also to call up the help menu, if commands and information have already been entered. I find that I have picked up, in using other programs, the previously innocuous habit of hitting the return key extra times, to tell the program to hurry up. Until now, this caused no problems (neither did it do any good!). It simply was an application of my 10-year-old daughter's wisdom, that when you don't know what else to do, hit the return key. But with Benchmark Graphics, each successive use of the return key brings up the Help menu. Few things are more irritating, when one has just signalled the program to hurry up, than to have it divert and read a very thorough (and complete and well done) help menu from a floppy disk. The Victor has plenty of keys; Benchmark should have dedicated one of them to calling for help, as do other programs.

Strangely, for rectangles, circles, ellipses, arcs, curves, or free draws, Benchmark cannot use the variety of thicknesses and patterns that are available when drawing straight lines (although one could painstakingly replace rectangles with straight lines.) This is a puzzling omission (unless I have missed an instruction somewhere in the manual).

Everyone will have different tastes, but I found the business graphics quite slick and easy to use, once the data are available. Bar charts can have as many as 64 bars, pie charts as many as 64 sections, and line graphs and x-y plots six curves (or data can be plotted without curves). Axes can be scaled automatically or manually. A single well-designed menu sets up the parameters for whatever business graph is to be drawn.

Business graphics data can be entered manually into a small 64x64 spreadsheet contained within Benchmark Graphics, or it can be read into this spreadsheet from an outside DIF (Data Interchange Format, a fairly universal standard) file. Thus, to get data from an outside spreadsheet or database, one would first fire up the outside program, write the outside program's data into a DIF file, exit the outside program, enter Benchmark graphics, and then read the DIF file into Benchmark graphics' small spreadsheet. This procedure is clumsy, but workable. To me, it was a major disappointment, because Benchmark Graphics is marketed along with Benchmark's version of the Report Manager three-dimensional spreadsheet, Advanced Version 2.2, which I have used very successfully for several years.

Incidentally, anyone who purchases Benchmark's version of Report Manager (now \$60) will get Advanced Version 2.20e; Datamension itself is now supplying version 2.25 (\$99), because Benchmark's marketing agreement with Datamension requires that Benchmark always be one revision behind. Report Manager AV2.20 (regardless of letter), although now several years old, contains several functions and other enhancements that I have found extremely useful. I have not yet used version 2.25, which is available from the Share House division, Datamension Corporation, 615 Academy Drive, Northbrook, Illinois 60062, and may be available from Technically Yours. Given Benchmark's claim to have an integrated software package (via the Administrator), I had expected, or at least hoped, that Benchmark Graphics would work with the Report Manager spreadsheet directly, rather than needing to go through a DIF file as an intermediary. Report Manager's only deficiencies are that it does not work directly with Lotus 1-2-3 WKS files, and that it contains very limited graphics.

Nevertheless, Benchmark's current business graphics is a marked advance over Benchmark 1.0 business graphics because that first version had no capability of reading in outside data: all data had to be entered by hand, perhaps (I do not recall) each time the program was run! Readers who have ever tried to use Victor Technologies' Graphics Toolkit BusiGraf program (written in GW Basic) will recall that it suffered this same deficiency. BusiGraf suffered a further problem: its x-y plot could deal with only 40 data points.

Benchmark Graphics x-y plots can deal with up to 64 data points. This is better than BusiGraf, but with the Victor 9000 screen having 800 pixel dots horizontally and 400 vertically, one would dream that someday, somebody will come up with an X-y plot program that will deal with equally large data sets.

Benchmark's line graph or x-y plot permits the data to be connected either by straight lines, or by smooth curves. Benchmark does not say so, but apparently the smooth curve facility calculates and graphs an  $(n-1)$ th degree polynomial for  $n$  data points. This facility is well done.

Benchmark Graphics greatly simplifies inserting dimension lines in diagrams. After you simply draw a dimension line wherever needed, Benchmark adds an arrowhead at each end, and asks you for the numerical dimension (e.g. 5.3 inches) which it then inserts in the middle of the line you just drew. You do not need to leave a space in the line to make room for the dimension; Benchmark erases just enough of the line to make room for it. Unfortunately, if during later editing you change the number of digits in the numerical dimension (e.g. from 5.3 to 5.25 inches), the new number will not fit properly. Nevertheless, Benchmark's dimension line feature is quite useful.

## The Administrator

Benchmark Graphics comes with a (shell-type?) program called the Administrator, which lists a menu from which one can select Benchmark Wordprocessing (which I do not have, because I need the symbolic mathematics capabilities of Scroll Systems PS Technical Wordprocessor, an excellent--8 scratchpad buffers!--program, even if one did not use the mathematics), Benchmark Graphics, Benchmark (Report Manager) spreadsheet, Benchmark Telecommunications (not too useful to the general user, because Benchmark telecommunications must be on both ends), and any other programs entered into the Administrator by the user.

I suppose Benchmark's Administrator is somewhat similar to the IBM TopView program, which I have read about but never used. For someone with a hard disk, the Administrator could be quite useful. However, in my dual double-sided floppy system, the Administrator would need to tell me to put in the appropriate disk for whatever program I had selected. Although it can do so, this message is not useful to me--I can remember to put in the appropriate floppy disk all by myself. On a floppy system, one would want to run Benchmark Graphics (or any other Benchmark program) without the Administrator's overhead. However, this is not possible: Benchmark tells me that Benchmark graphics version 3.0 must run through the Administrator. Perhaps some modification would permit bypassing the Administrator, but Benchmark says not.

With or without the Administrator, the ideal way to run Benchmark graphics would be from ramdrive. Unfortunately, if one loads all the Benchmark Administrator and Graphics files into ramdrive, they don't fit--even in a 512k Ramdrive. Numerous files are in the neighborhood of 100k, and one exceeds 200k. Benchmark nowhere provides a diskid or other list stating which files belong to the Administrator and which belong to Benchmark Graphics, much less which files do what, and the numerous filenames do not give such precious secrets away. Fortunately, Benchmark happily told me over the phone which files were essential. Telephone support (although a toll call) was excellent.

Benchmrk.lpd is a printer driver library (280k!) used only when first installing the program, and Benchmrk.adm, .sdr, .vid, .sio, and .prd are associated files that you do not need unless you are changing printers; none of these need be in ramdrive. On the Victor 9000, Reserve.com is a "terminate and stay resident" program that is used only once, just before running the Administrator, so there is no point in putting it in ramdrive either. So I put the following in my ramdrive. On the Victor 9000, modcon.exe requires Bchmark.chr and Bchmark.kb. The Administrator uses Bmrk.com, Benchmrk.dat, Benchmrk.ovl, and Benchmrk.cfg (this last used

only to reconfigure the Administrator's instructions). Graphics requires Graphics.bmk, Grovlay2.bmk, Grbgr1.bmk, Grbgr2.bmk, Grprint.bmk, Grplot.bmk (if you have a plotter), and text fonts such as Normal.scr and Normal ptr (for the screen and printer respectively). For graphics help, load all .gra files. If you are going to create or modify new fonts, also load Fonbuild.bmk and Font.hlp (if you have room in ramdrive--if not the Administrator will run Fontbuilder from a disk). My batch files to load, exit from, and return to Benchmark graphics are given in an appendix at the end of this review.

Incidentally, running everything from ramdrive almost eliminates the time wasted when being punished for excessive "hurry up" use of the return key by getting unwanted help menus, because the Help menus appear almost instantaneously from ramdrive. Valid Technologies' new Cache program might be a useful alternative or supplement to ramdrive, but I do not yet have any experience with it.

The Administrator's installation routine's menu-and-questions gives the impression that the Administrator should load a specified primary program (e.g. Benchmark Graphics) automatically if told to do so, but this feature never was implemented. The Administrator is also supposed to be able to select a default data drive (e.g. drive b) on which images would be saved, but this feature is not designed to work with Benchmark Graphics (but only with the wordprocessor and Report Manager spreadsheet). Benchmark understandably did not include the necessary Benchmark.dfd file on my graphics-only disk. To save an image to a drive other than the default drive, e.g. to drive b, one must include a b: at the beginning of the filename. No great hardship.

One small peculiarity of the Administrator is that after one moves the cursor to the desired menu item, one selects it not by hitting the return key (as is usual with other programs), but by hitting the Continue key. It took me several minutes to find the Continue key, because all my other programs either ignore it or use it for other purposes (as does MS-DOS). Eventually I found it, labeled CONT. On the Victor 9000 one could use keygen to modify the Bnchmark.kb, of course, but putting the Continue code on the return key, for example, would mess up the return key.

To exit the Administrator, one hits the Request Cancelled key, which my other programs either ignore or use as an Undo key. Metasoft, developer of Benchmark software, does seem to live in a world of its own. Perhaps its programs could be patched to give them a "user interface" more consistent with other programs. I love my Victor 9000, but after experiences like this, I remember Apple's claims that the interfaces for all Macintosh programs are identical. If the MSDOS world continues to lose ground to Apple, or at least to suffer inroads, I think unnecessarily nonstandard MSDOS

program keys will be a major reason.

As Benchmark's manual explains, before running Benchmark on the Victor, one must run an MS-DOS utility Reserve.com (fortunately included by Benchmark, because I never had heard of it) to reserve 80k of memory for graphics workspace. I am happy to report that after first running Guy Gordon's versions of the public domain utilities res.com (to make command.com stay resident) and scrnsave.com, and his own ramdisk, Reserve.com caused no conflicts.

### Benchmark Graphics' Basic Structure

Benchmark's business graphics are almost completely automatic, once you have specified parameters such as where in the spreadsheet to find the data, line type, bar graph fill patterns, and titles and labels.

To use Benchmark to create a freestyle graph, you first draw various basic elements (rectangle, circle, ellipse, line, arc, curve, or free draw) called primitives. Periodically you name whatever collection of primitives you have drawn so far. These named collections are called images. For each image, Benchmark remembers every preceding image included in it, and every primitive added to the preceding image, in sequence. At any time, you can call up a list of the preceding images and primitives that comprise the current image, or any preceding image. You can go back and edit any of these image or primitive components: moving or rotating it, zooming (enlarging or contracting) it, deleting it, or changing its characteristics (such as making a straight line dashed instead of solid). As the Benchmark manual points out, the advantage of this approach is that one can build a library of frequently used components and then call them up as needed, much the way lawyers call up boiler plate paragraphs as needed when writing contracts or other legal documents. If one wishes, successive images in drawing the final document can be saved, not only in memory but also on disk. In fact, you can save any part of your drawing to disk, using a rectangular window of any size, and then recall this part later, inserting it anywhere in any diagram.

Image naming is where careful planning is useful, because if you want to move, rotate, or zoom a combination of items--for example, a rectangle superimposed on a circle--you can do so only if you named the combination as a single image. If you did not do so, you must insert a newly named image that includes only the combination of items you want to manipulate. Unlike Knee-toe graphics, for example, at least as it was initially conceived, there is no way to draw a window around part of your diagram containing many components and then move, rotate, or zoom, whatever is in the window. (The change size or zoom feature never was implemented in Knee-Toe graphics, and Knee-Toe permits you to flip but not to

rotate the contents of a window.) Efficient editing of a completed diagram requires one to predict--while you are drawing--the images you may later wish to move, rotate, or zoom, and to remember to name save each of these pieces as separate items as you go along.

Benchmark's basic building block approach is useful--but the need to forecast possible future changes is a limitation. I have not yet figured out a good system for naming the images (pieces) of my "art" (economics diagrams). Benchmark Graphics does let you blank or erase an arbitrary rectangular window in your diagram. All that is needed to make Benchmark as versatile as it should be is to add moving, rotating, and zooming to this blanking capability.

### Texts and Fonts

For labeling drawings, and for text comments within a drawing, Benchmark has a good font that can be enlarged or contracted (although not rotated to any angle other than horizontal or vertical). Moreover, Benchmark Graphics comes with a font editor that permits you to create your own fonts, although I have not used this editor because I am reasonably satisfied with the "normal" screen and printer font (one each) supplied with Benchmark. Nevertheless I was surprised that Benchmark did not supply additional fonts, such as italic and script, and perhaps even a Greek and mathematical symbol font. Benchmark told me they would be happy to provide--at no charge--all the additional fonts they have developed "in house," provided that the user understand that these additional fonts are not professional efforts. Professional or not, they at least would be a place to start: modifying an existing font would be much easier than creating one from scratch.

In using the font editor, one annoyance is that the manual requires you to figure out for yourself the aspect ratio relationship between the screen and printer font (which is dependent on your printer type), based on your printer manual, which in my experience typically are obtuse. But the normal screen and printer fonts that Benchmark supplies already have at least a semi-satisfactory aspect ratio, and it would be nice if Benchmark manual told you what that aspect ratio was.

Unfortunately, Benchmark text fonts are not well coordinated with the rest of the graphics, in the following sense: if you move, rotate, or zoom a drawing (or component of a drawing) that contains both text and graphics, only the drawing is affected. Text must be resized (and moved as necessary) separately.

### Summary

Benchmark Graphics requires a minimum of 384k of Ram on the Victor 9000, and now costs about \$138 (\$125 for members, probably less in the near future) from Technically Yours. It was written by Metasoft of Chandler, Arizona, but now has been purchased (along with other Benchmark programs) by ShareData Inc., Suite C-190, 7400 West Detroit Street, Chandler, Arizona 85226 (602) 961-0005.

Benchmark Graphics is not perfect--for some purposes, I would prefer Knee-Toe Graphics, which has a wider variety of line types and fill patterns. But Knee-Toe's arc-draw and change size (of window contents) capabilities never were implemented. Benchmark's arc and curve facilities, as well as its free draw, are not only available but well done, and its ability to call up pre-drawn images will be very useful once I get used to it. Also, as noted earlier, Knee-Toe has no business graphics capability.

In addition to Benchmark and Knee-Toe and Busigraf, there are other graphics programs available on the Victor 9000, but I have no experience with any of them. They include: Autodesk's AutoCad (an engineering-oriented program, presumably very expensive, that might be useful for other purposes also), Easel Graphics (a forerunner of Knee-Toe Graphics), The Grafix Connection [860 Via de la Paz, Pacific Palisades, California 90272 (213)459-7949] for three-dimensional business graphics, ChartMaker, DataPlot Plus, and MicroChart Plus. The last three, which are business graphics programs, were reviewed by Colin Whiting in 16-Bit Computing (March 1985). Easel received a scathing review by Friedman Wagner-Dobler in 16-Bit Computing (June 1985), but in my judgment most of the criticisms are cured or at least alleviated in Knee-Toe Graphics. Easel apparently does include an arc-draw capability. Picasso and Doodle are two other graphics programs that I recall seeing advertised in 16-Bit Computing, but I know nothing about them.

### Appendix

REM AUTOEXEC.BAT copies useful utilities (although not needed for Benchmark).

```
path c:\\;a:\\;b:
```

```
ramdrive /s=19
```

```
copy ramdrive.com c:
```

```
copy chkdisk.com c:
```

```
copy backup.com c:
```

```
copy cpf.com c:
```

```
copy mvf.com c:
```

```
copy mmf.com c:
```

```
copy whereis.com c:
```

```
pause: Do you want Benchmark Graphics now? (Type  
LOCK C or ALT C if not.)
```

```
bmload
```

```

REM BMLoad.BAT (Note:
    Victornf.kb and victornf.chr
    are my console files.)
c:ramdrive /s=512
reserve
copy modcon.exe c:
copy bmrreturn.bat c:
copy dosretbm.bat c:
copy victornf.* c:
copy bnchmark.* c:
copy bmrk.com c:
copy benchmrk.dat c:

```

```

copy benchmrk.ovl c:
copy benchmrk.cfg c:
copy graphics.bmk c:
copy grovlay2.bmk c:
copy grbgr?.bmk c:
copy grprint.bmk c:
copy grplot.bmk c:
copy normal.* c:
copy *.gra c:
c:
bmrreturn

```

```

REM BMRRETURN.BAT
modcon bnchmark
bmrk
dosretbm

```

```

REM DOSRETB.M.BAT
modcon victornf
REM Save any graphics files from
ramdrive C to floppy drive
B or A.
REM Back up disks.

```

\* \* \*

## DEFEATING THE FUZZY WUZZIES

/ R. W. Johnson

Have you ever tried to TYPE a Wordstar file in MSDOS direct and wondered why your screen suddenly got messed up as though in some kind of strange video? The reason is simple enough: In Victor, ALT N printed to the screen is, according to the Victor manual, a "Shift to Character Set 1". ALT O is a "Shift to Character Set 0". Now, unless you have Character Set 1 installed, as with some kind of EFONT or MODCON, the computer has nothing in its file to use. Thus, after receiving the equivalent of a "PRINT CHR\$(14)" command (which in Wordstar is used as one of the codes in the formatting sequences), it prints either nothing or a few strange blobs and continues doing this. Telling Victor to PRINT CHR\$(15) immediately undoes the damage, even though this is not echoed to the screen, and the regular character set is restored. It is not necessary to re-boot as many people do. (Go into BASIC and try this yourself and see.)

I have a compiled program XX.EXE, which is simply a BASCOM compiled and LINK linked version of the two BASIC instructions, 10 PRINT CHR\$(15): SYSTEM, saved in ASCII. When the computer gets the fuzzy-wuzzies in MSDOS, I simply type XX <CR> and the fuzzy-wuzzies retreat to their lair to pounce again another day.

Incidentally, if you're into such things as passwords in programs you write, then try PRINT CHR\$(14): SYSTEM when the password entered doesn't match the reference. The program will bomb out and leave the prying user with a blanked out screen as well.

[The Public Domain programs ENCODE.COM and DECODE.COM (PD #59) are equivalent programs. So, also, are the programs with the same names in "BEEP - it yourself", VictorTalk, Vol. 3, No. 9, August 1985, p. 12.

The latter really require only the first eight items of data (they are then identical to the PD #59 programs); the remaining data is simply "Created by: (author's name)". Also, see "The Bill Johnson Show", VictorTalk, Vol. 3, No. 2, February 1985, p. 9, for further discussion of "secure video", passwords, etc. - Ed.]

\* \* \*

## MAXELL DISK PROBLEM / R. W. Johnson

After much agony and messed up disks, it seems I have discovered what may not be generally known: Maxell MD2-HD (gold and red box, bearing the words "Perfect for PCAT" printed at an angle) are NOT the same as the old standby MD2-DD (which I have been using for years for our Victors without problems).

Only a few of the MD2-HD disks will format at all on Victor, and most of these show at least a few soft errors. More importantly, files put on the disk cannot be read without error later on. One must use the "ignore" option, and then discover that after the first page or so there is a long line or two of @@@@'s and DDDDD's and everything bombs out from then on.

Naturally, I blamed it on an aging floppy drive, but after several hundred dollars worth of repairs it was proved to be the disks. A new box of Sony MD-2D disks (on sale at Target stores for \$8 per box retail) formatted perfectly - all 10 disks with 0 soft errors, 2 sides each.

Al Bullock confirms this, saying he has heard of other customers complaining about the new Maxell disks. So, a word to the wise - beware of Maxell MD2-HD. As far as I know, the MD2-DD (if available) is OK.

\* \* \*



# FROM THE BOARDS

[FROM THE BOARDS is assembled from messages appearing on various (generally Victor-specific) bulletin boards, mainly The PUB and BAVUG, by John and Mary Jo Knobelsdorf. Only minor editing is done to correct obvious typographical errors and to eliminate some superfluous material. An effort has been made toward a degree of standardized formatting and the elimination of excess space to the extent practical without degrading legibility. Garbles that cannot be readily rectified are allowed to remain. Believing that considerable worthwhile material is contained in these excerpts, our intention is to include this feature in each issue of VictorTalk, as space will allow. - Ed.]

[1] #97 (of 109) MACK MOORE, on 17-JAN-88 01:33  
(Size: 380 Bytes)  
Subject: 48 TPI diskettes

Hate to show my lack of knowledge, but... I offered to copy some White Crane PD utilities for a fellow in Toronto on CI\$. He sent some 48 TPI diskettes. I've never tried these on a Victor and don't wish to harm either the diskettes or the machine. Will they work? He claims he's got a V9K. No communication with him since I received the diskettes he sent.

[1] #99 (of 109) MICHAEL WISHNIETSKY, on 17-JAN-88 07:45 (Size: 507 Bytes)  
Subject: Re: 48 TPI diskettes

48 TPI diskettes usually work fine; as long as the diskettes format o.k., and chkdisk shows no problems, you're home free. I much prefer MS-DOS FORMAT.EXE Ver 2.4, because using the switches allows one to format a diskette for zero soft errors and an immediate return to DOS without the "format another" question. But the zero soft error designation is still valid in Ver 2.7 for MS-DOS 2.1 FORMAT -- but NOT for DOS 3.1 FORMAT.

The command used in the batch file is:

```
format right/c/e/f0
```

[1] #100 (of 109) MICHAEL WISHNIETSKY, on 17-JAN-88 07:52 (Size: 920 Bytes)  
Subject: Re: Protocols, Downloads, etc.

ZMODEM and YMODEM are not available in any comm software packages which will run on the Victor 9000 (at least none that I know of). Kermit will not run on the Victor version of Crosstalk (latest is Ver 3.5). Kermit is

included in Ver 3.61 which was the last IBM released version prior to Crosstalk Mark 4. Kermit Ver 2.29 for the V9000 is on this board, one or more of the NewsDisks, and on disks sold by Victor\*Group and Victor PULSE. It is a separate comm program which includes the Kermit protocol.

There are no upload/download requirements on this board (yet!). Some ham-pawed bear screwed up when you were added to the password file. Your password file has been cleaned up so that you can download to your heart's content. There were a couple of others that had the download restriction switch set incorrectly also. Just goes to show -- if something doesn't seem to be working right, ASK!

[1] #102 (of 109) DAVID ELESH, on 17-JAN-88 15:06  
(Size: 253 Bytes)  
Subject: VT100 Emulation

Thanks to Charles Vaughn and Tony Pompa, I can now do VT100 emulation. You do have to MODCON a keyboard; on my disk, it is called the XTALK.KB, not the VT100.KB. Makes it a little more difficult to figure out what is going on. Thanks again.

[1] #105 (of 109) MACK MOORE, on 19-JAN-88 00:27  
(Size: 190 Bytes)  
Subject: Re: 48 TPI diskettes

Michael: Thanks for the 48 TPI info. They worked fine. Why must the /c and the /e switches be used. I thought the /e "implied" the /c and would return the same error info.

\* \* \*

---

## V9000 HARDWARE

Al Bullock, of Bullock and Associates [39522 Benavente Place, Fremont, CA 94539 (415) 651-6183], currently has available the following hardware, for Victor 9000 computers, to user group members at the prices indicated:

Keyboards (10 function key, French & other)	\$50
Monitors	65

\* \* \*

---

## VICTOR COMPUTERS FOR SALE

Victor 9000, single-sided drives, 128K RAM	\$350
Victor 9000, double-sided drives, 256K RAM	\$500
Victor 9000, 10MB internal hard disk, 256K RAM	\$750

Dave Dillman  
(707) 792-2553

\* \* \*

# THE LIBRARY REPORT / John Knobelsdorf

On January 6, 1988 we released PD disks #93 & 94. On April 6, 1988 we released PD #95. These disks are in the Chapter set and available for copying at the monthly meetings. They also may be ordered thru the mail in the usual manner.

The files on the new disks came from the PUB and BAVUG bulletin boards. Some of the files will require the use of a Victor VI, Plus-PC, a PC-Compatible or one of the various

emulators. Be sure to read the DISKID file on each disk and check the descriptions before you try to use a particular program. Also remember that some of the files on the diskettes are in compressed form.

Use NJSQ (on PD #73) to "Unsqueeze"  
Use LJE " to "Unlibrary"  
Use PKXARC (on PD #87) to "Unarchive"

The contents of PD #93 are:

DISKID93.TXT 3584 Information about the files on this disk PD #93.

## THE FOLLOWING FILES CAME FROM THE PUB

ARCE131B.ARC 7305 Latest version of Bueg unarc; unique features.  
COUNTM4.ARC 32633 Set of utilities to count lines, words, chars.  
DOSTIPS4.ARC 74550 Useful information on DOS, fourth set.  
DOSTIPS5.ARC 78778 Useful tips on DOS, fifth set.  
DOSTIPS6.ARC 56320 Tips on DOS, sixth set.  
JZCHKDSK.ARC 25600 Attractive replacement for Chkdsk. Try it.  
MATHFACT.ARC 28032 Math drill program for kids  
RAW\_COOK.ARC 2048 Informative text on DOS handling of control chars

## THE FOLLOWING FILES CAME FROM BAVUG

BOOTKEY .ARC 1536 Control AUTOEXEC.BAT with a keypress (inc source)  
CERRMSC4.ARC 3712 Critical error handling in MS C v4 (w/source)  
DRVEPS .ARC 3840 Epson drivers for Autocad.  
FIX123V2.ARC 5888 Lotus123 unprotect vers 2 & 2.01 - great  
IBMBIO38.ARC 42240 Ver. 3.8 of Kurchat's emulator w/ Hirner KB mods.  
INVS LINK.ARC 25472 \*Background\* XMODEM file transfers!  
LITEBA27.ARC 7168 Litebar from PC Magazine  
MSUTIL .ARC 9216 Collection of "C" Utilities  
PDOXDEMO.ARC 184832 Paradox 2.0 Demo Program

580608 Bytes in 18 Files 26624 Bytes Free

The contents of PD #94 are:

DISKID94.TXT 3584 Information about the files on this disk PD #94.

## THE FOLLOWING FILES CAME FROM THE PUB

COUNTM5.ARC 33024 Many ways to count chars., words, lines.  
CRACKER .ARC 26624 Disassembler - works on the Victor  
FGREP160.ARC 10496 Latest ver of indispensable text string finder.  
FINREP .ARC 10240 Good text string finder from Eric Gans.  
LSHARK .ARC 21248 Loan amortization and listing program.  
OVER203 .ARC 8192 Get to subdir without specifying path.  
PCO330 .ARC 128000 PC-Outline, ver 3.30; best "thought-processor".  
PS35 .ARC 65536 Norm Patriquin's excellent string search utility.

SHRKSRC .ARC 15104 Pascal source code for LSHARK.ARC  
 SPFND2 .ARC 14336 Yet another GREP for Sam and Charlie.  
 UNIX .ARC 4096 UNIX utility to count repet. of successive lines.

THE FOLLOWING FILES CAME FROM BAVUG

WSSI322A.ARC 114688 Latest version of a great file/library manager.  
 WSSI322B.ARC 107520 Part "B" of above.  
 X00 .ARC 28928 FOSSIL Driver Documentation, etc.

593536 Bytes in 16 Files 16384 Bytes Free

The contents of PD #95 are:

DISKID95.TXT 4864 Information about the files on this disk PD #95.

THE FOLLOWING FILES CAME FROM THE PUB

123TAX87.ARC 55680 Lotus .WKS for '87 taxes; has "what if" capability.  
 ADVBAS40.ARC 155136 Hanlin's ADVanced BASIC functions for QB 4.0 \*FB  
 BNMKGRPH.ARC 11520 Review of Benchmark Graphics version 3.0  
 CHOP3 .ARC 30720 Chops a big file into lil files; good options  
 COOKIE .ARC 6400 Presents ND 30 fortunes one-at-a-time  
 DU .ARC 11136 Quick check of disk bytes used by dir / file  
 FEDBB .ARC 4480 Open Access Federal bulletin boards  
 FIZZ14 .ARC 15744 Permits wildcards and multiple params in DOS comms.  
 GD .ARC 384 Go to drive and directory in one move  
 MT224E .ARC 3328 Modem settings for MNP 3 & 5 on MT224E Modem  
 NULL2 .ARC 5632 Finds and optionally deletes zero files; doc added.  
 PORT .ARC 21888 Ports from IBM-V9000; Brookly Bridge Reviews/Info  
 RSLDRIVE.ARC 8320 RSL Ramdrive 2.00 for WAL V2, works with IBMBIOS  
 SETFDATE.ARC 13696 Set filedate/time; wildcards o.k.  
 SORTF227.COM 2963 Version 2.27 of Bueg's fast sort  
 SORTF227.DOC 9479 Doc file for ver. 2.27 of Bueg's Fast Sort

\*\* The following PUB files require use of an emulator \*\*

BRNOUT23.ARC 19456 #Best PC "screen saver"; w/ ASM source  
 DRAIN .ARC 10368 IBM - Cleans out drive A  
 EDED .ARC 27648 a nice editor.....emibm needed  
 ZIP .ARC 17920 IBM - Neat DOS utility

THE FOLLOWING FILES CAME FROM BAVUG

8088-V20.ARC 4352 Test for V-20 chip  
 CLINK120.ARC 31744  
 FBR158 .ARC 14336 Fast Backup & Restore by V. Bueg v1.58  
 GENASM .ARC 22912 gen asm code from exe or com files  
 NMLCKOFF.COM 128 For AST autoexec: tames NumLock.  
 TPC14 .ARC 39168 Pascal to C converter

\*\* The following BAVUG files require use of an emulator \*\*

DIAGCODE.ARC 8704 DIAGNOSTIC CODES FOR IBM. VERY USEFUL.  
 PCBOSS .ARC 23680 Hard Disk Management System - IBM or Emulator

581786 Bytes in 29 Files 6144 Bytes Free

The Librarian has received three expressions of concern that publishing files and programs in archived form on the V\*G public domain disks creates a difficulty for some users who may not be familiar with the necessity and/or method of unarchiving the file before attempting to use the file.

The principal reason for publishing in archive form is that when two or more archived files are unarchived, we frequently find that the different archive files contain a file with a filename identical to a file in the other archive.

It is easy to imagine the resulting confusion if a particular public domain disk DISKID listed several identical file names with different byte sizes and descriptions. And think what that would do to the "Public Domain Library Index". Fortunately, such a DISKID is not likely to be created, because there is no way to copy more than one file with the same filename onto a floppy disk.

Now someone will suggest that someone (else) unarchive all of the archived files, and revise all of the duplicate filenames. That someone or someone else had better have a lot of spare time and a good imagination.

One of the most frequently found duplication of filenames among archived files is README, or variations of that name. This circumstance originates with Programmers, Sysops and even users like you and I who bundle a bunch of files together into an archive file, including their own individually created README file to explain or document the archive file. The problem is even more complex when someone takes a couple or more archived files that contain files with duplicate filenames, and archives them into a new archive file. There are more than one of those beauties floating around.

Instead of accepting recommendations to continue treating the symptoms, we should develop a cure for the disease. The use of more descriptive .DOC or .TXT filenames instead of README and its variations would be a step in the right direction. Unfortunately, it will be difficult to completely eliminate duplicate filenames, because (in addition to README files) there are currently in existence any number of individual files, including some .ARC files, of different contents but with the same filename.

Since there have been only three expressions of concern during the last year or so since we started publishing archived files, perhaps it is not such a problem after all. Perhaps the V\*G member users of the public domain disks and the archived files would like to express their individual opinions. If so, write a note or letter to the Librarian or the Editor or the Steering Committee, and provide your thoughts and suggestions, pro and/or con. Please include a potential solution for any problem that is identified.

\* \* \*

## TECHNICALLY YOURS

[The following letter (dated 3/7/88) updates the Technically Yours discounted membership offer, which was originally described in VictorTalk, Vol. 5, No. 2, Mar-Apr 1987. A note on Yorgos Hatzis and David Koshinz was given in VictorTalk, Vol 5, No. 6, Nov-Dec 1987, page 21. Excerpts from Technically Yours' most recent price list appear on the following page. - Ed.]

Dear Victor Group;

Thank you for sponsoring your members in Technically Yours. We are pleased to inform you that, because of the enthusiastic participation of Victor Group members in Technically Yours in 1987, we are extending discounted Technically Yours memberships to your users for another year, with no additional sponsorship fee to Victor Group. Victor Group members who would like to purchase new Technically Yours Memberships (priced at \$29.95) can continue to do so for only \$11.95. Existing Victor Group / Technically Yours members can expect to be offered an additional discount from the \$11.95 fee when their Technically Yours membership is due for renewal.

We have some announcements which may be of particular interest to your members. First, we have recently joined forces with Stratos System (Yorgos Hatzis & Dave Koshinz) so that Technically Yours members can get quality Victor 9000 and PC compatible repair service at an exclusive discounted cost. Yorgos and Dave also have Victor parts and V9000's available, and they are assembling an extremely high quality 296 System that can be purchased from Technically Yours. In addition to Stratos Systems' move into the Technically Yours Computer Services Center, we have just completed agreements with Info-Based Solutions (specialists for inhouse accounting and bookkeeping software systems), who will be moving in soon. The folks at Info-Based have Victor 9000 experience and can be helpful to V9000 owners who can use their services.

Technically Yours continues its ongoing commitment to expanding services, better value and price for all of our members, no matter what computer they use. We appreciate our Victor 9000 member base, and thank you for your support.

TECHNICALLY YOURS ENTERPRISES, INC.  
PO BOX 1414  
APTOS, CA 95001  
1-408-688-5201

\* \* \*

TECHNICALLY YOURS

MEMBER PRICES  
(Effective March 1, 1988)

**S O F T W A R E**

**FOR YOUR VICTOR 9000:**

Crosstalk XVI	\$119
White Crane Util. 3 Pac	79
VT Cache (Any V9 DOS)	95
WordPerfect 4.2	239
MathPlan 2.1	139
Brooklyn Bridge	179
Report Manager 2.25	89
Benchmark Graphics	99
Fancy Font	149
DS Backup +	49
dBXL	139
D Base III Plus (WCS)	624
DataEase	\$CALL
SmartKey II	60
MS DOS 3.1	75
CYMA/McGraw-Hill (Accounting Series)	\$CALL
<b>OTHERS</b>	<b>\$CALL</b>

**PC Compatible SPECIALS:**

BROOKLYN BRIDGE	\$89
AUTOSKETCH	50
dBXL	119
AMORTIZER PLUS	79
DS TUTOR (FOR MS-DOS)	21

**H A R D W A R E**

V9000 MONITOR	55
V9000 KEYBOARD EXCHANGE	55
V9000 VALID MEMORY BOARDS	
0KB	149
0KB w/ Clock	189
256KB	199
256KB w/ Clock	239
512KB	259
512KB w/ Clock	289
768KB	299
768KB w/ Clock	349
VT CACHE w/ Board Purchase	79

**V9 SPECIALS (While Supplies Last):**

ACCOUTOUCH TYPING	49
FILES & FOLDERS	59
REPORT MANAGER 2.2	59
MULTIPLAN 1.2	69
BENCHMARK WORD PROCESSOR	69
BENCHMARK 3D SPREADSHEET	69

\*\*\*\*\*  
**STRATOS SYSTEMS**  
 V9000, PC, XT, AT, & 386 COMPATIBLES  
 REPAIR SERVICES & PARTS SUPPLY  
 TY MEMBER DISCOUNTS START AT 15%  
 \*\*\*\*\*

(Bold faced items indicate change from previous price list.)

WE SUPPLY A WIDE VARIETY OF COMPUTER ACCESSORIES

CALL TECHNICALLY YOURS PURCHASING ASSISTANCE AND ORDER LINES AT:  
(408) 688-5201 OR (408) 761-0765

Technically Yours' annual membership fee is \$29.95 and offers you savings through member exclusive prices, product information, support materials, technical information, repair services and referrals for other services.

NON MEMBERS ADD 10% TO THE PRICES ABOVE.  
 Prices are subject to change without notice  
 YOUR CHECK, MASTERCARD OR VISA WELCOME

TECHNICALLY YOURS ENTERPRISES, INC.  
 PO BOX 1414  
 APTOS, CA 95001

\* \* \*

## VICTOR\*GROUP ADDRESS CHANGE

After serving Victor\*Group faithfully for several years, Nancy Weaver has found it necessary to resign her position as Secretary/Treasurer. Her duties are being taken over by Carol Hirner. With this change comes a new mailing address, as shown in the adjoining masthead and forms. Henceforth, it is requested that this new address be used for everything other than VictorTalk editorial matters. These will be expedited by communicating directly with the editor.

\* \* \*

## RE: VICTOR CIRCLE NEWSDISK DISTRIBUTOR CHANGE

VictorTalk Vol. 6, No. 1, Jan-Feb 1988, noted the change of distributor of Victor Circle NewsDisks to Michael Wishnietsky for the Western States. Please note his corrected address:

Michael Wishnietsky  
23 Adams Road  
Bedford, MA 01730  
(617) 274-6689

We are sorry to note that, in addition to having distribution difficulties, the publication of Newdisks has become erratic. NewsDisks 21, 28, 29, and 31 have not been issued. The remaining ones through Newsdisk 33 have been issued. It is understood that NewsDisk 34 is about ready for release by Tony Pompa. It is certainly hoped that means can be found to continue this worthwhile endeavor.

\* \* \*

## VictorTalk

### Address inquiries to:

BAY AREA VICTOR USERS GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

**Subscriptions:** Carol Hirner

**Editor:** Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

**Contributors:** George Anderson, Roger Folsom, R. W. Johnson, Mary Jo Knobelsdorf, John Knobelsdorf, Dan Mardesich, Jack Prichard.

VictorTalk is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in VictorTalk. Copyright 1988 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

VictorTalk is an independent publication not affiliated in any way with Victor Technologies, Inc. Victor is a registered trademark of Victor Technologies, Inc.

**Submit Articles** on MSDOS disks in ASCII or Spellbinder format, with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

## Yes, I want to join.

Enclosed is a check/money order for \$35.00  
(foreign membership, \$40.00 in U.S. dollars)  
for my 1988 Victor\*Group membership.

Name \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

Make checks payable to:

## VictorTalk

1493 Beach Park Boulevard, #180  
Foster City, CA 94404

City \_\_\_\_\_

State \_\_\_\_\_ ZIP \_\_\_\_\_

Country \_\_\_\_\_

Phone \_\_\_\_\_

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
State _____	
City _____ & Zip _____	
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No. Disks	Vol. No. Disks	Vol. No. Disks
4 2 61-64	5 1 78-81	5 5 87-90
4 3 65-71	5 2 82-84	5 6 91-92
4 6 72-77	5 4 85-86	6 2 93-95

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40
														*(34 & 35 are a two-disk set)					
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56*	57*	--	59	60
														*(56 & 57 are a two-disk set)					
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	—	—	—	—	—	—	—	—

1988 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, sets of 6, \$12.00 /set ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_

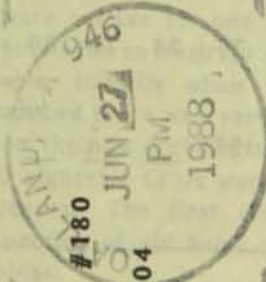
1988 membership fee \$35 (receive all 1988 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR\*GROUP

1493 BEACH PARK BLVD, #180

FOSTER CITY, CA 94404



Brood Wagon 1880s  
23 USA



Lillian M. Gilbert



FIRST CLASS MAIL



PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date \_\_\_\_\_ Office use only:  
 Name \_\_\_\_\_ Order rec'd \_\_\_\_\_  
 Company \_\_\_\_\_ Member check \_\_\_\_\_  
 Address \_\_\_\_\_ Comments \_\_\_\_\_  
 City \_\_\_\_\_ State \_\_\_\_\_  
 & Zip \_\_\_\_\_  
 Country \_\_\_\_\_ Phone \_\_\_\_\_ Order shipped \_\_\_\_\_

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No.	Disks	Vol. No.	Disks	Vol. No.	Disks	Vol. No.	Disks
4	2 61-64	5	1 78-81	5	5 87-90	6	3 96-97
4	3 65-71	5	2 82-84	5	6 91-92		
4	6 72-77	5	4 85-86	6	2 93-95		

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
 21 22 23 24 25 26 27 28 29 30 31 32 33 34\* 35\* 36 37 38 39 40  
 \*(34 & 35 are a two-disk set)  
 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56\* 57\* -- 59 60  
 \*(56 & 57 are a two-disk set)  
 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80  
 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 \_ \_ \_

1988 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_  
 Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_  
 Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_  
 Postage/handling (applies to disk orders only)..... \$ 4.00  
 Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_  
 VictorTalk back issues: 1986, 1987, sets of 6, \$12.00 /set ..... \$ \_\_\_\_\_  
 1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_  
 1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_  
 1988 membership fee \$35 (receive all 1988 issues of VictorTalk) .. \$ \_\_\_\_\_  
 (Foreign membership, \$40 in U.S. funds.)  
 TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

## IBMBIOS / Franz Hirner

Jack Prichard has been after me for some time to write an article on IBMBIOS for VictorTalk. That's a tough challenge. It's not that IBMBIOS is tricky, complicated, or requires a Masters Degree in Computer Science to use or understand. In fact, it's just the opposite. I take IBMBIOS almost for granted just like any other utility program. Since there's nothing difficult or tricky about using IBMBIOS to enable your Victor 9000 to run hundreds of IBM specific software programs, it's difficult to imagine creating an entire article to deal with such a simple application. Those of you who frequent either the BAVUG or the PUB BBS are already familiar with IBM emulators for the Victor 9000. For those of you who have not yet heard of emulators for the Victor 9000, I'll take pen in hand and see if I can fulfill Jack's wishes.

As mentioned from time to time, one of the more enjoyable aspects of running the BAVUG RBBS is working with various programmers developing useful software for the Victor 9000. One such program is Ralf Kurschat's IBMBIOS. IBMBIOS is a software emulator that allows the Victor 9000 to run many IBM specific programs.

Software emulators are fairly common in the computer world. A properly written software emulator enables software written for one type of machine to run on a machine with an entirely different architecture. For example, initially when the 16-bit computers were introduced there was very little software available to run on the new architecture. Until then, the 8080 and Z-80 eight-bit CPUs were employed in most personal computers. The first emulators introduced enabled the porting of old 8-bit programs to the new 16-bit machines.

Other emulators have been written to allow a PC to be used to develop software for the IBM 370 mainframe computer, thereby avoiding costly time charges. (Incidentally, 8080, Z-80, and IBM 370 emulators are available on the Victor\*Group Public Domain Disks or can be downloaded from the BAVUG BBS if you would like to try your hand at running some 370 code on your Victor 9000.)

There have been four basic software emulators written for the Victor 9000. Each provided various degrees of compatibility with software written specifically for the IBM PC, which is what we are concerned with at the moment.

### IBM.COM

One of the earliest emulators was Guy Gordon's program IBM.COM. IBM.COM was part of Guy's dBASE III package for the Victor 9000. IBM.COM was very limited in it's capability - in fact was almost totally specific to dBASE III. In addition to the IBM.COM emulator, Guy had to make patches in the code of the dBASE III program itself in order for dBASE III to run on the Victor 9000.

While IBM.COM allowed dBASE III to run on the Victor 9000, it is very limited since each individual application program would require hand patching, which would be very time consuming and expensive. As a matter of fact, dBASE III was the only program that I know of that was modified to run under IBM.COM.

### EMIBM

The next attempt at an emulator that would allow IBM-specific software to run on the Victor 9000 was EMIBM. It was written by Danial Cerutti in Switzerland. It is generic, rather than specific to a given application as is Guy Gordon's IBM.COM. It does not require that the application program itself be modified to run on the Victor 9000. This is an improvement over Guy Gordon's IBM.COM, which was very specific to dBASE. EMIBM does allow many programs, both commercial as well as public domain, that were written for the IBM PC to run on the Victor 9000.

EMIBM, while an improvement, is still limited to programs that were written using text mode rather than graphics mode for screen display; it does not make use of the IBM screen swapping capability. (Screen swapping allows a programmer to give the appearance of very fast program execution by doing screen updates on a hidden screen and then swapping that screen onto the display. Since the user didn't see the actual printing of each character he is not aware of the real time required to perform screen writes. Screen swapping is very common in commercial software but, at that time, was a little too tricky for amateur programmers). EMIBM also does not properly handle all of the specific IBM BIOS calls and, as a result, is limited as to the programs it can run. EMIBM is limited to running mostly public domain IBM programs, but a few commercial IBM programs will also run under its emulation. Still, it is

a major improvement and many Victor 9000 users are still using EMIBM to run IBM programs today.

### XTBIOS and HXTBIOS

More recently, Ralf Kurschat appeared on the emulator scene. Ralf's first two attempts at emulating the IBM hardware on the Victor 9000 produced the emulators XTBIOS and HXTBIOS. XTBIOS is really not much better than Danial Cerutti's EMIBM. Ralf did properly trap and handle some additional IBM specific hardware and BIOS details, making many additional programs available for the Victor 9000 that until now could not be run, but XTBIOS still had most of the limitations that we found in EMIBM.

Ralf's real gem is HXTBIOS; it is the first emulator that is able to run programs that write directly to the IBM screen, as well as many programs that employ the screen swapping technique mentioned above. This is a major improvement, since many commercial programs employ one of these methods to provide the fast screen display that the user associates with speed. Like all things, there is a price to pay for the performance improvement offered by HXTBIOS; it requires 896K memory to run. Fortunately, Victors with 896K memory are rather common today.

Ralf lives in Germany; quite naturally, he included the European type keyboard in all of his emulators. The European keyboard has a different layout than the American keyboard, additionally, several characters and punctuation characters are totally different than those we use in the United States. With the aid of DEBUG and SST I found myself poking around in the code to make the necessary changes for XTBIOS and HXTBIOS to respond to the U. S. keyboards. Initial modifications were an improvement, but only a beginning.

HXTBIOS has continued to evolve, but under the name IBMBIOS. The current version is 4.01. It is much enhanced over the earlier EMIBM.COM and IBM.COM. As far as I know, further development of EMIBM and IBM has been dropped.

### IBMBIOS

(Much of the following text is taken directly from the EMIBM documentation with some editing.)

IBMBIOS allows the V9000 user to run many programs that were developed for the IBM PC and compatibles.

This is achieved by simulating an IBM PC/XT environment, including the IBM Color Graphics Adaptor (CGA) card, in software.

IBMBIOS can only be run on Victor 9000 systems that are equipped with 896K RAM. Those who don't have this required amount of memory can still make use of XTBIOS or EMIBM, but these emulators fall far short of providing the capability of IBMBIOS to properly handle programs that access the IBM hardware directly. If you don't have full memory in your Victor 9000, you now have a reason to upgrade.

IBMBIOS can emulate the following IBM video cards:

- o MDA - Monochrome Display Adapter
- o CGA - Color Graphics Adapter

The CGA card can have seven screen modes with different resolutions (maximum 640 x 200) and a number of colors (maximum, 4 at a time). All seven modes are supported by IBMBIOS. Colors are represented on the V-9000 by different scanning patterns. CGA emulation is accomplished by CGINIT which reserves 40K for the Victor graphics screen. Selection of the emulated video card is done automatically by IBMBIOS. When emulating CGA mode you can still run many of the Victor-specific TSR software such as RAMDISK or VT-CACHE but CGINIT should be run first in order to avoid memory conflicts. The best way to run CGINIT is from your AUTOEXEC file or from batch files. They're easy to write and, once set up, make the entire process of loading CGINIT and IBMBIOS transparent. IBMBIOS can be installed manually if you prefer. I have renamed my copy I.COM so that it is only a one-letter command to go into IBM mode. If I want to run VF ( a file maintenance utility) I merely type:

```
I VF <CR>
```

and the emulator, along with VF, is loaded into memory and begin execution. When I terminate VF, IBMBIOS conveniently uninstalls itself automatically, leaving my machine in native Victor mode.

If you like, you can enter IBMBIOS <CR> only, and IBMBIOS will load and stay resident, leaving you in "IBM" mode until you decide to exit. When in IBM mode, a special prompt appears on the screen:

```
IBM $p >
```

where \$p is substituted by the current path name. When this prompt appears, all commands can be entered and batch files or programs in the IBM mode can be started just like in normal Victor mode. Note: The display prompt can not be changed by the PROMPT command when in this mode.

The V-9000 stays in the IBM mode until entry of the new command:

```
.E <CR>
```

IBMBIOS switches back to Victor mode and terminates. The old prompt will then reappear and you're back in your native Victor 9000.

If you'd rather automate the entire process you can write a simple batch file to do your chores for you. Lotus 2.01 is probably the toughest, since it requires both CGINIT and IBMBIOS in order to run. You'll most likely want to clear both programs when you exit Lotus, so that you are back in pure Victor mode. Your batch file would look like this:

batch file command	comments
CGINIT	; reserve memory for graphics
IBMBIOS LOTUS	; loads the emulator and Lotus;
	; on termination of Lotus,
	; IBMBIOS also terminates
KILLSCRN	; (from DOS 3.1) releases
	; graphic memory
	; optional
MENU	; or whatever command to return
	; to your Victor system

Programs can have parameters passed to them if necessary:

```
IBMBIOS programnam parameter parameter parameter
```

I won't go into the technical details of how IBMBIOS actually works. Ralf has provided an excellent description of this in the documentation for those who are interested.

Nothing is perfect. IBMBIOS will not run IBM communications programs, but most non-communications programs run fine. I use the Brief Programmers Editor, Microsoft Code View, and MSC 5.0 almost every day. This article is being written with the IBM version of Word Perfect 4.2. As this is being written (May 16, 1988), Word Perfect 5.0 has just been released within the past ten days and is already

running under IBMBIOS. There are a few problems that must be addressed, but so far it looks as though this mini-desktop publishing program will join the list of IBM specific software that can now be run on the Victor.

Following is a partial listing of programs that I or other Victor users have run under IBMBIOS:

```
Word Perfect 4.2
dBASE III
Wordstar 2000 Release I
Wordstar 2000 Release II
Q & A
Nut Tree
Quick Basic 4.0
Javelin
Brief
Miscrosoft C 5.0 w/Codeview
Wordstar Professional
Lotus 2.0 (after unlocking)
Lotus 2.01 (after unlocking)
Quattro
Paradox 1.1
```

There are many other IBM commercial programs that run on the Victor 9000 under IBMBIOS, as well as thousands of public domain programs. If you have questions about a particular package let me know, either through VictorTalk or on Bavug, and I'll provide any additional information I can.

If you've felt left out because you couldn't run some of the newer software packages that are not released in Victor format, then you might want to give IBMBIOS a try. It's available from the Public Domain Library or you can always download the latest version directly from BAVUG BBS (415-574-7914).

[Editor's Note - While IBM.COM and HTXBIOS are apparently not in the Victor\*Group Public Domain Library, the other programs and their documentation can be found on the following PD disks:

```
XTBIOS: PD 87
EMIBM: PD 58, 62, 70, 75, 77, 79, 88
IBMBIOS: PD 93, 96
```

(The last two programs appear under spelling variants.)

\* \* \*

# MAKING MUSIC ON YOUR VICTOR / Andrew Rudiak

Victor 9000s are business computers. Their sound-generating capability is generally limited to warning beeps. However, the CODEC chip can make a variety of sounds, from sirens to foghorns, from belches to Bach. Following are several ways of creating musical sounds. Belches you will have to work out yourself.

Computers have advantages over humans when it comes to performing music. For one thing, they don't breathe. A computer doesn't have an emboucher that fatigues after 45 minutes of performing Mahler. Once the program is debugged, the computer's performance is absolutely consistent, and it doesn't need hours of practice to maintain its form. Change one parameter, and a computer can perform Chopin's Minute Waltz in 15 seconds - or 15 minutes.

Commodores have three sound channels, Ataris have four, and the Apple //GS has 15 channels. Victors have one. Victors are solo, monophonic instruments, more like a trumpet than an organ. That must be kept in mind when programming music.

Victors usually generate music through GWBASIC or VBASICA programs, and that will be the focus of this article. There are two main methods of making music through these languages: The SOUND statement (also useful for sound effects) and the PLAY statement. Each has advantages and disadvantages. Both will be described.

The PLAY statement is versatile and easy to use. It permits two ways to define each note or rest, and establishes changeable defaults for tempo, note length, and octave, so that these parameters needn't be defined for each tone as it is generated. PLAY and its subcommands allow more compact coding.

The SOUND statement, while more difficult to implement, can be used to generate special effects such as glissandi (trumpet smears), sirens, and so on. It also overcomes one limitation of GWBASIC's PLAY statement: Victor Technologies' (or perhaps Microsoft's) failure to implement the Pause (i.e., rest) command.

Musical scores are actually a kind of "program." A programmer has to translate them into a form the computer can understand. Both SOUND and PLAY require programmers to transcribe music as DATA, which can then be executed.

BASIC and music possess great logical similarity. Music has its own equivalents for WHILE WEND and FOR NEXT LOOPS, not to mention GOTOs and IF THEN ELSE branching. What is a cadenza (virtuoso solo ad-lib, for you non-musicians) if not a LINE INPUT statement?

All music falls into two broad categories: strophic and through-composed. Through-composed music contains no systematic repeats, though themes may recur. Strophic music has repeating themes and regular structures. Songs with several verses and choruses are a good example of strophic music. Once the repeated sections are analyzed and identified, it is easy to use them again and again, calling them and running them through loops as needed. One of my earliest transcriptions, a piece called TICO-TICO, boiled four pages of music down to 14 short lines of DATA by taking advantage of strophic structure. Some pieces, however, cannot be compressed in this fashion, and may require hundreds of lines to transcribe. Save pieces like that for later.

\*\*\*\*\*  
\* Programming with PLAY \*  
\*\*\*\*\*

The PLAY statement is easier to use than SOUND. It contains a music-like macro language of subcommands. See Figure 1 for a complete list.

Figure 1. The PLAY macro language

A-G[#,+,-]	Plays notes A-G. #,+ sharpen tone, - flattens it.
L<n>	Sets default note length. L4=quarter note, L2 half notes, etc.
MF	Music foreground. Music executes first.
MB	Music background. Other lines execute while music performs. MB is not implemented in GW-BASIC.
MN	Music Normal. Each note gets 7/8s of the value set by L.
ML	Music Legato. Each note gets 100% of value set by L. Use with caution. It wipes out the space between repeated tones; e.g., four 8ths of the same pitch will sound as a half note.
MS	Music Staccato. Each note gets 3/4s of value set by L. MN and MS speed up the music in GW-BASIC because the implied rest interval (1/8 and 1/4 of the L value) is not inserted. These commands function effectively in VBASICA, however.
N<n>	Plays note <n>, ranging from N1 to N8. N8=middle C. N is a good way to execute a note without changing the octave default. N0 allegedly equals P(L(n)). It doesn't.
> <n>	Go up an octave. Doesn't work. Never implemented.
< <n>	Go down an octave. Doesn't work. Never implemented.

O<n> Sets octave. Values are 0 to 6. Default, o3=middle c to b.

P<n> Pause (rest). Values =1 to 64. Doesn't work in QBASIC.

T<n> Tempo. Ranges from 32 to 255. Default = 120.

. Period. Same function as music notation: raises n by 1.5x. May be used with L and P. May be concatenated: ... etc.

X<string> eXecutes substring. Very handy for compressing repeated motifs. MUST be followed by semicolon (Xa\$; etc.). Cannot be used with VBASICA compiler.

"<" and ">" are mentioned only because they are included in the documentation, which erroneously states that they are implemented in VBASICA 2.0 and later. Don't believe it. They aren't. Too bad. They would have been very useful. Lower case letters work just fine for all the above commands. (P.S: "<" and ">" do work in IBM mode on Victors equipped with Plus-PC.)

The examples for using PLAY in Victor's manuals are abstract and confusing. Below is an example which, I hope, will give you a better idea of how to use these commands. Let's start by programming a simple piece (Figure 2):

Figure 2. Twinkle, Twinkle Little Star using PLAY

10 REM TWINKLE.MUS	Identify filename
20 CLS:KEY OFF	Clear screen
30 LOCATE 13,20:	Put title on screen
35 PRINT "TWINKLE, TWINKLE LITTLE STAR"	
40 REM DATA for TWINKLE.MUS	
50 DATA "MF MN T150 L4 o3"	Set defaults right up front
	Music Foreground, Music Normal, T(empo)150, L(ength)4, o(ctave)3
60 DATA "c c g g a a g2"	"Twinkle, twinkle little star" A duration value can be placed immediately after a note ("g2"), changing that note only, not the default.
70 DATA "f f e e d d c2"	"How I wonder what you are"
80 DATA "g g f f e e d2"	"Up above the world so high"
90 REM	"Like a diamond in the sky"
99 REM METHOD #1 FOR EXECUTING DATA:	
100 READ DEFAULT\$,A\$,B\$,C\$	Treat DATA as substrings; execute it all with one PLAY statement.
110 PLAY "DEFAULT\$,Xa\$;Xb\$;Xc\$;Xc\$;Xa\$;xb\$;"	
120 RESTORE	Release the DATA for re-use.
199 REM METHOD #2 FOR EXECUTING DATA:	
200 DIM A\$(6)	DIMension array
210 FOR PHRASE=1 TO 4	Treat DATA as array variables
220 READ A\$(PHRASE)	READ data
230 NEXT PHRASE	
240 A\$(5)=A\$(4):A\$(6)=A\$(2):A\$(7)=A\$(3)	Equate repeated (strophic) phrases
250 FOR MUSIC=1 TO 7	
260 PLAY A\$(MUSIC)	Run array variables through loop to execute
270 NEXT MUSIC	
280 CLS:KEY ON:END	Reset screen; end program

To run TWINKLE, either remove the documentary comments or turn them into REMarks. Now it will play twice. The first method of execution works nicely for very short pieces. Variations on the second method are more useful for longer pieces with repeated sections.

As programmed, TWINKLE executes in C major. Transposed to F major, it demonstrates the use of N, O, and "l-" in context, as it changes from one octave to another and uses B flat. Figure 3 contains the adjusted DATA lines:

Figure 3. Transposing Twinkle

```
50 DATA "MF MN T150 L4 o3"
60 DATA "f f o4c c d d c2"
70 DATA "o3b- b- a a g g f2"
```

This data doesn't change. It's good in any key.

Notice the octave change (o4)

Octave changed back. Note that the flat sign (-) is specified each time for each b-. Key signatures do not establish defaults. All notes use the piano-key equivalent names; use F for E# and B for C- for PLAY statement purposes. However, either b- or a# is OK

```
80 DATA "N49 n49 b- b- a a g2"
```

N49 was used to avoid designating o4 for two notes, then redesignating o3.

For some pieces, "quick and dirty" works best. Just transcribe all necessary DATA, then put a line at the bottom of the program which says:

```
500 READ A$:PLAY A$:GOTO 500
```

This isn't very elegant. It terminates with the following error message: "OUT OF DATA IN 500." But music programmed this way is short, uncomplicated, and easy to read. Analyze each piece to decide which route to take.

The following practices are useful when converting music notation to BASIC data: use lower case letters to represent tones (a-g). This helps distinguish between notes and commands (which should be upper case). It's particularly helpful distinguishing 0 (zero) from (capital) O. Try to make one phrase equal one data line, unless the data line is very short or very long. If the phrase covers several measures, put a space between each measure's data. This technique makes it easier to find the inevitable transcription errors. Also, exact coding of phrases makes them "portable." You can then reuse them if they recur later in the piece. You only have to code it once, though you may have to "rename" it (see line #240, above).

Save your music (or for that matter, any BASIC) program in ASCII format, using the "A" switch (e.g., SAVE "MUSIC-BAS",A<CR>). Each version of BASIC seems to tokenize differently from every other version. Your program will only work with different BASICs (or be readable from DOS) if SAVED in ASCII format. Also, programs must be saved in ASCII format to be compiled.

```
*****
* Programming with SOUND *
*****
```

An 1860's housewife's guide—in addition to tips on how to prime wells, geld calves, and stay moral in a decadent society—included some recipes for boiling, baking and frying potatoes. The latter recipes concluded with the admonition that frying and baking were a waste of time: sensible housewives boiled potatoes, per recipe #1.

In that spirit, be aware that programming music with SOUND is tedious. Practical programmers use PLAY. Don't say you weren't warned.

The format used with SOUND is SOUND <frequency>,<duration>. Frequency is in cycles per second using any valid numeric expression with values from 37 to 32767. Use values over 14000 only if programming for dogs, bats or dolphins. Duration can be any positive number from 1 to 65535. 0 turns the sound off. Duration is measured in clock ticks, which are set at 18.2/second. According to the manual, SOUND can be buffered with the MB (Music Background) subcommand. This will only work—if it works—in VBASICA, as MB is documented but not implemented in GWBASIC.

Victor included PHAT.BAS (pronounced P-Hat) with the Graphics Tool kit. This program, ~~ffffd~~ *ffff* borrowed from IBM's demo disk, uses SOUND to play a short version of the Mexican Hat Dance. PHAT is one of the few implementations of SOUND I have encountered, so, edited and annotated, it is included as Figure 4, below. Extraneous code has been removed for clarity.

Figure 4. PHAT.BAS, using SOUND

```
1000 DATA 52,2,57,2,0,2,52,2,57,2,0,2,
          52,2,57,6,0,4,52,2,57,2,59,2,57,2
1005 DATA 56,4,57,2,59,2,0,8,52,2,56,2,
          0,2,52,2,56,2,0,2,52,2,56,6,0,4,
1010 DATA 52,2,56,2,57,2,56,2,54,4,56,2,
          57,2,0,6,64,2,63,2,64,2,61,2
1015 DATA 60,2,61,2,57,2,56,2,57,2,52,2,
          0,4,49,2,50,2,52,2,54,2,56,2
1020 DATA 57,2,59,2,61,2,62,2,59,2,0,4,
          62,2,61,2,62,2,59,2,58,2,59,2
1025 DATA 56,2,55,2,56,2,52,2,0,4,64,2,
          63,2,64,2,66,2,64,2,62,2,61,2,
1028 DATA 59,2,57,2
1030 DATA -1,-1
1035 DIM F(36)
1040 FOR I=0 TO 36
1045 F(I)=220*2^(I/12)
1050 NEXT I
1080 F(0)=30000
1085 READ NOTE,DUR
1090 WHILE NOTE > -1
1095 IF NOTE=0 THEN NOTE=45
1100 SOUND F(NOTE-45),DUR
1105 READ NOTE,DUR
1110 WEND
```

The DATA lines 1000-1025 provide the labels and durations for the musical notes. Actual frequencies are generated by lines 1040-1050 plus a special frequency in line 1080 (really an inaudible pitch used for rests). The formula  $220*2^{(I/12)}$  generates 3 octaves of the tempered scale starting at A below middle C. These values are used in the F(I) array variables. The WHILE-WEND loop is readied in 1085. For some reason, 45 was added to each note value in the DATA lines, then is subtracted in the WHILE-WEND loop (line 1100). The condition in the WHILE line (WHILE NOTE>1) is triggered by the values in DATA line 1030, and the program shuts down.

Using this algorithm, you can substitute your own data and play any piece you like. Just keep the frequencies and durations in the right sequence. Figure 5, below, shows how to extend the range from three to seven octaves.

Figure 5. Extending algorithm in PHAT.BAS

```

1035 DIM F(85):          'Seven octaves
1040 FOR I=1 to 84:      '{Frequencies
                        {generated, plugged
1045   F(I)=55*2^(I/12): '{into values of F(I)
1050 NEXT I:             '{Next frequency
1080 F(0)=30000:        'Human-inaudible tone
                        used for rests

```

Establish your own values for durations. They don't have to be integers.

The SOUND command can do one thing easily that PLAY cannot do at all. That is transposing (i.e., changing keys) without completely recoding the data. SOUND uses numeric data exclusively. Thus, adding an integer value (even a negative value) to frequency data allows you to change keys.

I recently coded a short piece called CLAIMJUMPER which transposes one-half step up on each repeat. Instead of coding each key change separately, I coded it once and added 1.0 to each value of F(n) on each run through the loop. Thus, F(37) becomes F(37+1) or F(38) - the next note up the scale. This allowed the piece to execute in F Major, F# Major, and G Major on successive repeats, and probably saved 4 hours of coding and 4K bytes of disk space.

Using a parser line made it possible to code resettable duration defaults. Thus, duration data did not have to be coded for each and every note. The parser (an IF... THEN statement) causes a slight but discernible hesitation in the execution of the music. However, it reduced data coding by an additional 45%.

```

*****
* Terra Incognita *
*****

```

There are limitations to programming music with GWBASIC and VBASICA. Until recently, there was no way to compile these programs after you wrote them. They couldn't be shared (legally) unless the sharee also had the above languages. Hardly anyone owns the compiler, so the problem still exists. Also, GWBASIC and VBASICA can't use display driver escape codes, so there is no way to control audio level from within your program.

Theoretically, you should be able to program music with other languages. The only problem is that hardly anyone knows how to access the CODEC. Victor Technologies hasn't been generous with this information. At least two programmers have found a way, however.

MUSIC.DOS (Figure 6) on Victor Circle's NewsDisk 9 (ND9), by Michael Stal of Munich, shows how music can be generated using MSBASIC, a language every Victor owner should have. Unfortunately for most Americans, REMarks in this file are in German. This limits comprehension. [A translation of lines containing German words has been added. - Ed.] One puzzler is that as data values get lower, pitches get higher. In real life, higher numbers yield higher pitches. Perhaps there is a reciprocal tucked away somewhere in the algorithm. Also, how does one generate the frequency data used by MUSIC.DOS? To work on this problem, check out ND9.

Incidentally, the MSBASIC program in MUSIC.DOS works just fine in GWBASIC and VBASICA. The functional part of MUSIC.DOS appears to be a DEF SEG statement and three POKE statements. Some sharp TURBO Pascal programmer should be able to work out Pascal equivalents using the same addresses. This would give Victor users a capability Borland International left out of the version they sell us. How's about it, TURBO users?

Mark Hansen of Orion Microsystems in Portland, Oregon, developed some routines which allow the CODEC to be called from MSBASIC. Mark's programs include CODEC.ASM, CODEC.COM, CODEC.BAS, and CODEC.TXT. They are documented to a fare-thee-well. CODEC.COM must be run to initialize the CODEC before you attempt to use it through CODEC.BAS.

CODEC.BAS allows you to plug in your data in the same format used by the SOUND statement. However, since you are programming in MSBASIC, it is also possible to change the sound level as your piece performs. Diminuendos (gradual decreases in loudness), crescendos



Figure 6. MUSIC.DOS

```

10 REM *****
100 DEF SEG=&HE802
110 REM ein demoprogramm, das ich von Michael Stal aus Muenchen erhalten
120 REM habe. Es zeigt, wie unter Basic auch ohne den Codecdriver Tuene
130 REM aus dem Victor 9000 gelockt werden koennen. ueli haenni
140 OKTAVE=1
150 FOR I=1 TO 27
160 READ NOTE,DAUER
170 POKE 10,5
180 POKE 8,INT(NOTE/OKTAVE):PRINT INT(NOTE/OKTAVE)" ";
190 FOR D=1 TO DAUER*30*OKTAVE:NEXT D
200 POKE 10,0
210 NEXT I
220 OKTAVE=OKTAVE+1 'EINE OKTAVE HOEHER WIEDERHOLEN
225 RESTORE 250:PRINT:PRINT"*****":PRINT
227 IF OKTAVE>4 THEN END
230 GOTO 150
240 REM DATEN FUERS LIED (TONHOEHE,DAUER)
245 REM "Alli mini Uenteli, schwimme n ufem See"
250 DATA 236,1,211,1,188,1,177,1,158,2,158,2,140,1,140,1,140,1,140,1,158,4
260 DATA 140,1,140,1,140,1,140,1,158,4,177,1,177,1,177,1,177,1,188,2,188,2
270 DATA 158,1,158,1,158,1,158,1,236,4
999 rem *****

```

Translated lines:

```

110 REM A demonstration program, that I have received from Michael Stal of
120 REM Munich. It shows how tunes can be called from the Victor 9000 under
130 REM Basic without the Codec driver. ueli haenni
140 OCTAVE=1
160 READ NOTE,LENGTH
180 POKE 8,INT(NOTE/OCTAVE):PRINT INT(NOTE/OCTAVE)" ";
190 FOR D=1 TO LENGTH*30*OCTAVE:NEXT D
220 OCTAVE=OCTAVE+1 'REPEATS AN OCTAVE HIGHER
227 IF OCTAVE>4 THEN END
240 REM DATA FOR SONG (TONE,LENGTH)
245 REM "(Song title - not certain of translation)"

```

(gradual increases in loudness) and sforzandi (sudden increases in loudness) are finally possible. You can also change your monitor's contrast and brightness while your music executes. CODEC.BAS has one more sterling virtue: it eliminates the horrendous click which occurs between each note when you use GWBASIC.

CODEC.BAS may prove to be the very best way to program music. It uses the most generic language around, gives you control over your screen and your speaker, eliminates the click, and can be compiled, eliminating the need even for MSBASIC. CODEC.BAS and associated files

and source code are available on Victor Circle Newsdisk #17.

Nitpickers may have noticed that no mention is made of the Audio Toolkit in this article. It's a personal prejudice. I find the difference between programming music and recording it with the Audio Toolkit to be about the same as the difference between oilpainting and Instamatic photography. The picture may be clearer, but the art has disappeared. Someone else can write about using the Audio Toolkit, if they want to.

\*\*\*\*\*  
\* POLYPHONY \*  
\*\*\*\*\*

I said above that Victors have only one sound generator. That's true. However, it is possible to fake two or more voices by moving rapidly from one voice to another. This technique can be called pseudopolyphony. It works the same way as motion pictures do. Movies are really a series of still photographs, moved past the eye at 24 or more frames per second, giving the illusion of movement through persistence of vision. Pseudopolyphony is a series of rapidly alternated notes - often 30 or more per second - which fool the ear through persistence of hearing.

The drawback to motion pictures is flicker. The kindred drawback to pseudopolyphony is warble. The music does not sound steady. Properly projected films exhibit little or no flicker. However, even carefully-programmed pseudopolyphony has a distinct warble, because notes are not sounded simultaneously, but consecutively.

There are other problems you will have when using this technique. With low frequency tones (for example, C three octaves below middle C), the speed of execution may be faster than the frequency of the tone, and the tone won't sound. This fracturing limits how many notes you can program per beat. Fewer notes increase the warble. More notes increase fracturing.

Also, while the computer accurately generates the exact frequencies which mathematically comprise the tempered scale, human ears perceive very high pitches as being sharper than they actually are. Once you get to the high range of a coloratura soprano, every note seems to be horribly out of tune. (This is a problem with sopranos, too.) This limits you to using the mid-range of the computer's voice (roughly 2 octaves either side of middle C). But what the hell. If you were really serious about music programming, you would have gotten a Macintosh and a MIDI keyboard and avoided these headaches.

Programming pseudopolyphony using the PLAY macro language is just like regular music programming, only vastly more subdivided. 4/4 time becomes 32/32 time when programming for two voices; 48/48 time when programming for three voices, and 64/64 time when programming for four voices. I don't recommend doing four voices very often. It's usually very fractured. The pattern I use depends on the number of voices being used. Generally, it goes like this:

- 2 voices - Bass to Soprano (1-2)
- 3 voices - Bass to Soprano to Tenor (1-3-2)
- 4 voices - Bass to Alto to Tenor to Soprano  
(1-3-2-4)

Staggering the voices is necessary because a straight-line progression (1-2-3-4) produces a distinct "wahwah" sound. This effect is nice on occasion, but it is distracting.

Roughly 8 times as much data is required to define a piece, rather the doubling you might expect when for a second voice. Eight, 12 or 16 notes now define each quarter note, depending on the voicing.

One beat of a pseudopolyphonic third (c to e) looks like this:

```
o3L32cececece
```

A triad looks like this:

```
o3L48cgecgecgecge
```

And a full chord looks like this:

```
o3L64cgen49cgen49cgen49cgen49
```

(Note the use of N49 to avoid changing the octave default.) Multiply this by 2-4 beats per measure, then multiply that by 16-240 measures per piece. That's one huge pile of data.

What a pseudopolyphonic transformation looks like in music notation is shown in Figures 7 and 8.

(Incidentally, these graphics were created with a program called EZENOTES, by ace Victor Circle programmer Jan Ewing. EZENOTES enables you to create musical scores on screen and then dump them to a printer for near type-set quality musical manuscripts. It's available for \$75 from Jan. Write to: Jan Ewing, 409 W. 24th Street #14, New York City, NY 10011 (212) 675-1974. If Jan gets enough orders, he may just update EZENOTES from version .9 to version 1.0 for the Victor.)

Because of the enormous amount of data required, it becomes necessary to confine each DATA line to one measure or less if you intend to compile the program or include animations. The MB (music background) buffer only holds 32 notes at a crack. You can't put any more than that into a given array variable or you will get unprogrammed and unwanted pauses in the music while the buffer refills. Two beats of 4-voice data will fill the buffer. The answer for logical lines requiring more than 32 notes is to store the data in two-dimensional arrays, the first half measure in the first pocket, the second half in the second pocket. (E.g., if the array is DIMensioned to MU\$(16,2), then you have measure (1,1) and measure (1,2), etc.) This will help you keep all data for one measure in one logical DATA line without overfilling the buffer.

Of course, if you don't plan to compile or animate, this isn't strictly necessary. It's a good practice anyway. When debugging, it helps you pin down your inevitable keyboarding errors to a given measure or a given DATA line.

Figure 7. Stars & Stripes Forever (simplified)

Figure 8. Stars & Stripes Forever (pseudopolyphonic)

Once your DATA is entered, you manipulate it the same way as you did with monophonic music. I recently programmed a four-voice Bach fugue. It required four solid pages of data, but this is all that was required to execute it:

```
500 FOR N=1 TO 115:READ MU$:PLAY MU$:NEXT
```

On the other hand, when I programmed The Stars & Stripes Forever, about 40 READ lines and another 40 PLAY lines

were needed to execute it. Not to mention 124 lines of DATA. Every piece has its own problems. The fun comes in working them out and listening to the results.

This article has been necessarily somewhat abstract. If anyone wants something more concrete (i.e., actual listings), contact me. I would be thrilled to find someone else as interested in making music on the Victor as I have been. I share my source code.

\* \* \*

# KEYBOARD FILES / Jack Prichard

This article is a companion article to the one on Character Sets in the last issue of VictorTalk (Vol. 6, No. 2, Mar-Apr 1988). Like the ability to easily change character sets, the ability to easily change keyboard assignments is also one of the many desirable features of the Victor 9000 computer. The keyboard assignment sets, similar to the character sets, reside in files with the extension .KB, a large number of which are similarly included in the Public Domain Library. The purpose of this article is to serve as an index to these files.

Much of the introductory information in the article on Character Sets applies similarly to keyboards; it will not be repeated here. This applies particularly to MODCON, the program also used to install different .KB sets into the operating system, in the same manner as .CHR sets are installed.

Some of the details of .KB files are described in the Graphics Tool Kit under the program KEYGEN. It provides for viewing the .KB file header and for changing the character or function assigned to each key of the keyboard. Like .CHR files, .KB files may be of a number of different lengths. (The lengths found in the Public Domain Library are 1280, 1322, 1324, 1408, and 1536 bytes.) Also, like .CHR files, .KB files start with a header. The header is typically 90 bytes in length; apparently it always starts with K (for keyboard) and ends with 10.

The .KB files in the Public Domain Library are listed alphabetically in Table 1. The procedure used to gather this information is essentially the same as that used to gather the information on the .CHR files. As no satisfactorily compact way was found to characterize the various .KB files, it was decided instead to simply:

- o Indicate, by an \* after the PD No. in Table 1, the existence of documentation (usually as a .DOC file) on the indicated disk.
- o Include an accurate copy of the header information, as this sometimes gives clues relative to the use of the .KB file. This information appears in the last three columns of Table 1; it is in exactly the form it appears in the header, except for some re-spacing for legibility (and the omission of the terminating 10). (The column headings are those used in KEYGEN, part of the Graphics Tool Kit.)

While the Victor 9000 feature of having a software programmable keyboard can be a considerable convenience, it can also be a trap! Many application programs install keyboard changes when they are invoked; some return the keyboard to its original condition when the program is exited, some do not.

For example, here's what happened to me: I discovered that the version of FORMAT that comes with MS-DOS 2.11 sometimes would not work. In fact, the initial screen would be displayed but none of the keys seemed to work and the program could not be exited. After re-booting a few times in disgust, I carefully explored possible means of escape. At last I found it! The sequence ALT CLR HOME and ALT C will exit FORMAT to the operating system and avoid the nuisance of rebooting.

I found that the reason for the problem was that FORMAT depends on the boot assignment of the cursor control keys and function keys 1, 2 and 3. When FORMAT failed to work I had previously run Spellbinder (Ver. 5.40), which had changed the function of the critical keys. Since Spellbinder apparently does not reinstall the original keyboard when it is exited, this must be done somehow. It can be done in either of two ways:

- o Use a batch file (.BAT) to enter Spellbinder; include a MODCON instruction to reinstall the original keyboard on exiting Spellbinder:  
SB  
MODCON BOOT  
The original (boot) keyboard and character set can be captured by previously running:  
MODCON \* BOOT  
This, of course, should be run when first booting, before running any program that could install another keyboard or character set. It will file the desired information as BOOT.KB and BOOT.CHR on the default drive. If the default drive is other than a RAMdrive, this need be done but once.
- o Prior to running FORMAT (or any other program requiring the boot keyboard and character set) (and after previously saving BOOT.KB and BOOT.CHR), simply run:  
MODCON BOOT

Since I frequently flip in and out of Spellbinder and only infrequently use FORMAT, the second solution is the preferable one for me.

TABLE 1. KEYBOARD FILES (Page 1 of 2)

Filename (.KB)	Size	Last Change	PD No.	Display Class	Name	Style	Description	
							Originator	Date
AUSWP4	1324	6/16/83 00:01	47	K0 Sirius Int'l	Aust/WP	04	Australian Wordprocessing C Barson	83/06/16
BASICUTL	1324	4/20/84 19:24	60*	K0 Victor Int'l	Standard	23	MSBASIC/BASIC Utility Keyboard Satore Center	4/84
BITS	1324	7/03/86 13:21	97*	K01 BITS.KB	BITS.KB	01	Used with MAKEBITS.BAS Jan Ewing	7/3/86
BRIT01	1408	7/03/82 11:59	05	K0 British	British	01	British Standard Sirius S/W	82/07/03
CENT1	1324	1/01/80 00:03	82	K0 game	Grafix	00	Special for Black Box toe knee	1-10-83
CONTROL	1324	6/05/87 18:12	92	K0	with control.		chr small caps ital etc s.e.m.	87:6:5
DESKCOMP	1280	12/09/84 18:42	31	K0 Sirius Int'l	American	01	Sirius American Standard Sirius S/W	82/07/03
DOS2	1536	9/01/85 10:44	62*	K0 Dos 1.25a kb	mxm/dos	1.7	DOS keys, CTRL, :,/	M.X. Moleski, SJ 30june85
EDLIN	1322	1/31/84 10:38	05*	K TEMP				
GAMES	1324	12/16/84 20:51	23	K0 Sirius Int'l	American	01	Games Keyboard Wick	84/1/16
HEBREW	1324	4/19/83 13:16	87	K HEBREW/ENGLI	HEBREW		English with Hebrew alternate ms3	17-4-83
KMAN	1324	9/28/84 13:33	59	K Victor				9-28-84
KTG	1324	1/01/80 01:09	71	K0 Application	Grafix	00	Special for knee toe Graphics toe knee	10-8-84
MBSI	1324	5/26/84 19:11	59*	K VICTOR			1=F1 2=F2 9=F1 10=F2	9-14-83
MODEM	1324	11/05/85 21:01	59	K SAVED			Format/Archive/Restore F1/F2,F3/F4 A W Happel, O.D.	11/5/85
MUTFUN	1324	4/08/87 22:18	92	K0 Victor Int'l	Domestic	03	703 Victor Programming Sirius S/W	82/03/03
OS	1324	1/01/80 22:16	05	K O/S kbd			function keys do MS-DOS commands David M. Barton feb 2, '84	
POSS	1324	1/01/80 00:16	60*	K0 Victor Int'l	Standard	01	Victor Standard Victor S/W	10 feb 83
REVM3	1324	4/27/87 09:41	87	K0 Victor Int'l	Standard	01	Victor Standard Victor S/W	10 feb 83
SAVE	1322	1/01/80 00:13	24	K SAVE				
SAVE	1322	11/18/85 08:44	59	K SAVE				
SAVE	1322	3/12/86 11:59	71	K SAVE				
SAVE	1322	6/03/87 06:39	92	K SAVE				
SCS20PD	1536	5/10/84 23:55	46	K0 Victor Int'l	Standard	23	MSDOS 2.0/Public Domain Keyboard Satore Center	4/84
SCS20PD	1324	4/20/84 19:11	60*	K0 Victor Int'l	Standard	23	MSDOS 2.0/Public Domain Keyboard Satore Center	4/84

Note: \* denotes documentation file on indicated disk.

TABLE 1. KEYBOARD FILES (Page 2 of 2)

Filename (.KB)	Size	Last Change	PD No.	Display Class	Name	Style	Description	
							Originator	Date
SCSDBASE	1324	4/20/84 18:27	60*	K0 Victor Int'l	Standard	23	dBASEII Keyboard Configuration	Satore Center 4/84
SCSML	1324	4/20/84 18:53	60*	K0 Victor Int'l	Standard	23	MicroLink Keyboard Configuration	Satore Center 4/84
SDS125	1324	4/20/84 19:39	60*	K0 Victor Int'l	Standard	12	MSDOS 1.25 Utility Keyboard	Satore Center 4/84
SQUARE	1408	12/03/85 21:28	72	K SQUARE				
TEMP	1322	1/01/80 00:01	05	K TEMP				
VICTORMP	1322	11/11/83 07:34	05	K TEMP				
VICTORMP	1324	11/01/85 15:21	59	K MULTIPLAN			MOD FUNC KEYS 9 & 10 FOR PAGEING	J.P. 5-30-84
VICTOR	1322	1/01/80 00:21	59	K VICTOR				
VICTORWS	1324	2/17/83 13:52	05	K0 Victor WS	VictorWS	03	Victor WordStar	Victor Tech, Inc 83/02/17
VICTORWS	1324	11/01/85 15:08	59	K	VICWS03		CHANGE DELETE TO ^G	J.P. 9/18/85
VMIN	1536	9/18/85 21:56	67*	K0 xt.kb & vminxt/vmin	0.1		^M after a..z, 0..9, vmin3.2fk's	M.X. Moleski, SJ 2sept85
WPKEYS2	1408	7/31/87 22:17	88	K Internat'l	AmerWP41	03	WordPerfect 4.1 (American)	AEB/GDB/BMP 7/7/87
WS	1408	1/01/80 22:02	05*	K0 WordStar	American	02	Std V9000 kb, Wordstar keys	David M. Barton 2-1-84
WSTURBO	1324	12/04/85 09:59	59*	K	VICWS03		WS KB FOR TURBO PASCAL	10/27/84
XT	1324	8/19/85 02:47	73*	K0 My xtalk.kb	Link/xtk	0.1	DOS keys, ^M after everything	M.X. Moleski, SJ 12jy85

Note: \* denotes documentation file on indicated disk.

\* \* \*

DISKS - DSDD VS. DSHD / George Anderson

[Several readers have commented about problems with various company's HD disks; this article explains the problem. - Ed.]

In the last issue of Victor Talk, R. W. Johnson warned of the problems of using Maxell MD2-HD disks, rather than the older MD2-DD disks. His advice was right, and the rest of us benefit from his agonizing experience.

You might be interested to know why we must use DSDD (Double Sided Double Density) disks. In the beginning (V9000, that is) Victor adopted a special format for its disk drives in the V9000. We have all heard the changing speed the drives go through whenever disks are formatted. Well, they change speed too when we read them (but it isn't quite

so easy to hear the speed changes then, unless you listen closely for them). The speed change was adopted to allow each sector to be more nearly the same length, whether it is located on an outside track or an inner track (remember those radial 'pie cuts' across the tracks on a diagram of a disk? They define the boundaries of a sector. The sector, of course, resides only on a single track). Now, the cuts do not cross all the tracks on the same radius line as before. The sector boundaries on the outer tracks are staggered from those on the middle tracks and the inner tracks. Each time the drive motor speeds up, it is adjusting sectors to make each one more nearly the same length.

Victor chose this format to allow them to place 600K plus bytes on one side of the disk without going to a special disk magnetic material to achieve that bit density (bit density - meaning how many bits (eight to a byte) per inch of track).

(Continued on page 27)

# EXPLORING MS-DOS, Part 2 / George Anderson

Just to review for a moment, let's recall the function of MS-DOS and the major elements which make it up. By now you are aware that DOS, the disk operating system, really has outgrown the title. (As in the first article, I will generally abbreviate MS-DOS to simply DOS.) Not only does it control the disk or disks in the system, its job has grown over several incarnations, to the point that it now provides a standard (well, sorta standard!) interface between the CPU and the various other hardware and software devices that make up our computer. These devices can be printers, scanners, mice, plotters, ramdisks, hard disks and/or floppy disks, to name a representative sample. DOS ties together all the peripherals with the CPU so they will work together. It provides us with a means of loading and running our programs and it provides a standard method of filing our data away on disk, and of retrieving it when we are ready to use it.

DOS is a collection of programs consisting of:

- system programs
- resident or internal programs
- transient or external programs.

We covered the system programs in the first article, and we will take up the resident programs in this one.

Resident commands are ones that are stored in RAM by COMMAND.COM at the time you boot your computer. This type of command is sometimes called "Terminate and Stay Resident", or TSR for short. They do not appear on a directory listing because they are contained in COMMAND.COM, and that is the only title that shows in the listing. Resident commands are instantly available when called, and do not require path or drive information to invoke them. They are simply named, and they are there, stored in RAM, ready to use immediately. A list of the resident commands for DOS 1.25 is given below, followed by the additions and the few deletions for DOS 2.11 and 3.1:

## DOS 1.25 RESIDENT COMMANDS

COPY	DIR	TIME	TYPE
DATE	REN	TOD	?
DEL			

## DOS 1.25 RESIDENT BATCH COMMANDS

PAUSE	REM
-------	-----

In DOS 2.1, Microsoft made a major change to accommodate many new services within DOS. Among these services was a new concept called 'directories'. This concept was aimed at making it easier for hard disk users to group files of a similar or related nature by placing files on a single subject into a directory, separating them from files on another subject. That way, the directory title would clue you to the general subject content of the files included within that directory. If you think of a directory as nothing more than a file folder containing files on one subject, you will have gotten the idea; and have cleared up much of the smog which has been written on directories. While this concept was aimed at bringing order out of chaos for hard disk users, DOS 2.0 and above allow you to apply it to floppy disks, also. We will be exploring the pro's and con's of directories later on, but this was to introduce the substantial growth in resident commands between DOS 1.25 and DOS 2.11. In DOS 2.11 some fourteen resident commands were added, plus five resident batch commands:

## DOS 2.11 ADDED RESIDENT COMMANDS

BREAK	CLST	MKDIR	SET
CAUX	CTTY	PATH	VER
CHDIR	EXIT	PROMPT	VERIFY
CLS	HISTORY	RMDIR	VOL

## DOS 2.11 ADDED RESIDENT BATCH COMMANDS

ECHO	IF-EXIST
FOR-IN-DO	SHIFT
GOTO	

And, two commands were deleted:

## DOS 2.11 DELETED RESIDENT COMMANDS

TOD	?
-----	---

With DOS 3.1, no new resident commands were added - either in the internal commands or in the batch commands, but two of the commands added in DOS 2.11 disappeared in DOS 3.1:

## DOS 3.1 DELETED RESIDENT COMMANDS

CAUX	CLST
------	------

We will discuss the batch commands along with the batch files in the fourth article of this group of DOS articles. Now let's take a look at how we may make use of the resident commands we have listed. First, we will look at those commands common to all three versions we are examining - i.e. DOS 1.25, 2.11 and 3.1:

#### DOS INTERNAL COMMANDS-IN-COMMON

COPY	REN
DATE	TIME
DEL	TYPE
DIR	

A few general comments here will help to clarify the principles behind DOS command organization, and allow you more freely to utilize the great variety of combinations in commands available to you through the use of switches and other options. Many of the options will apply to the external commands as well. We will discuss them in the next article.

Discussions concerning directories, below, apply only to DOS 2.0 and above. But this merely means that the number of different types of 'pigeon holes' you have with which to store and organize your files on disk is less complex in DOS 1.25. Of course, it's your choice as to whether you use these extra 'pigeon holes' in DOS 2.11 and 3.1.

Some of the commands have 'switch' options which can modify the command used in useful ways. These are sometimes confusing in that the same switch letter means different things when attached to different commands. Watch for the definitions appropriate to specific commands when you use switches. We will discuss some of the more common ones as we go through the resident commands. The others, then, should be more understandable in reading the manual. Wild cards may be used on all commands requiring a file name in order to operate. The principle of application is the same as for the examples given for COPY, below; so plan to explore their use on other commands, as well.

Although commands normally require the RETURN key to be pressed in order for the command to be carried out, I have not shown the RETURN in the examples to follow. Assume RETURN is necessary, unless a note to the contrary is included.

COPY Command [Switches: /A, /B, /V]

The COPY command copies files (data or program) in a number of different ways; from one disk to another, from one directory to another, from a directory on one disk to a

directory on another disk, or even from a file to an output device connected to your computer. COPY may also be used in creating a file with the same contents as an old one, but with a new name; or concatenating files together (appending or merging).

The output devices with which COPY is involved are usually a disk, the screen itself, or a printer connected either to a serial port (COM 1, for example) or to a parallel port (LPT1 or LPT2). The name of the printer itself (e.g.: PRN) may also be used. The basic organization of the command is as follows:

COPY (whatever file you choose) from (its source location) to (the location you choose).

So, we may COPY a file to disk, to a printer, from disk to disk, to a different directory on the same disk, or we may COPY to the CONsole (the screen) - all by the use of the COPY command. And we can concatenate files to create new files containing the data from all the old files used. Let's look at some examples using COPY:

EXAMPLE 1: To copy a file named BAD.STF from floppy drive A to floppy drive B, type the following:

```
COPY A:BAD.STF B:
```

NOTE: This example allows the default drive to be ANY drive connected to the machine.

We might already have set the default drive to drive A. In that case, we could shorten the command:

EXAMPLE 2: To copy the file above from and to the same drives, but with drive A as the default drive:

```
COPY BAD.STF B:
```

NOTE: You don't have to specify the default drive in the command, but there are times when you might wish you had!

EXAMPLE 3: To copy an ASCII file (readable as stored on disk) named GOOD.STF from disk to printer, using COPY (there are other ways, also):

```
COPY GOOD.STF > PRN (default printer)
or
COPY GOOD.STF > LPT1 (parallel port)
or
COPY GOOD.STF > COM1 (serial port)
```



Where: '!' is a symbol sending the file copy specified on the left of the symbol, to the device or port on the right. If it is a port, the file will be sent to whatever device is connected to that port. A serial port may mean a modem, for example, rather than a printer, if that is what is currently connected.

Now, suppose you want to copy all files belonging to a single class, at one time. Here is the place for 'wild cards' used to replace the variable part of the file names you want, coupled with the fixed or common part of the filenames. You can do this quite easily like this:

EXAMPLE 4: To copy a group of files having a common part to the filenames, e.g.: START.BAT, GO.BAT, QUIT.BAT and END.BAT from the default A drive to drive B, type the following:

```
COPY *.BAT B:
```

Where: the default drive is not required, and the '!' stands for START, GO, QUIT and END. Since all the three-letter extensions are BAT, we can type that in directly; but we have saved typing in all the file names.

EXAMPLE 5: To copy a group of files with similar file names, we have an alternative we can use. If the file names are LEADER.SCR and LOADER.SCR, and the copying is between the same drives as in Example 4, merely type:

```
COPY L?ADER.SCR B:
```

Where: the '?' stands for the variable letter in the file names.

The usual way of using these alternatives is to use '?' when there are one or two variable (or UNKNOWN) letters, and to use '!' when either the file names are different, but the extensions are the same; or the reverse condition exists, and the file names are the same, with the extensions different.

Concatenation strings together more than one file to form a new file holding the contents of all the files used. It is performed using COPY as shown in the next example:

EXAMPLE 6: To concatenate three ASCII files having the names TAXRCPT.JAN, TAXRCPT.FEB and TAXRCPT.MAR, all on drive A, to form a new file TAXRCPT.1Q, to be placed on drive A, type in the following command string:

```
COPY A:TAXRCPT.JAN + A:TAXRCPT.FEB  
+ A:TAXRCPT.MAR A:TAXRCPT.1Q
```

Switches /A and /B are used to define the type of file being concatenated. An optional method for the concatenation above uses the /A switch placed AFTER the first file name, and designating it as ASCII, together with all following files (until another switch is invoked). The /A switch is the default switch for concatenation. To concatenate BINARY files, use the same form, but substitute the /B switch in place of the /A given in the optional description above.

The /V switch will verify the copy made. It does this by reading the source a second time and comparing it to the destination copy. If both agree, the copy is considered verified.

One feature of COPY is that it 'follows the road signs' (pointers to the locations where parts of the file are stored - not necessarily in adjacent positions) and reassembles the file being copied in contiguous order in the process. You might wonder what difference it makes, since you have obviously been able to read from and write to a fractured file before. The main reason you might care is that it is faster to read or write to the file if it is in contiguous order. It also takes up less space on disk when the file is in contiguous order.

There are a few places where COPY does not work as you might expect. The most notable exception is with the DIRectory command. If you try COPY DIR > PRN, you might well expect to get a copy of the current directory printed for you. Not so. However, if you use DIR > PRN, you will get a printed copy.

Another place COPY does not work as expected is in trying to copy a file from one disk drive to the same drive (and to the same directory, for DOS's 2.11 and 3.1). This action will get you an error message telling you that you cannot copy to the same drive.

Keep in mind that if you are making up the file and typing it as you go, you can still use COPY. For example, you might create a text file or batch file using COPY CON (the filename you choose) by typing the file content on the keyboard. COPY CON sends the keyboard output (copies the keyboard output) to a location called CON - which is merely the shorthand name assigned to the CONsole (or screen/keyboard combination). Of course, any time you type the default output is to the screen, but with COPY CON, you may save the result to disk for use again. Simply

follow your typed input (to COPY CON) with an ALT-Z (displays as '^Z') as the last line of the typed copy. This will store the copy to disk. Let's look at an example:

EXAMPLE 6: To create a file named SETUP.TXT using COPY CON, and to store it on disk, type the following:

```
COPY CON SETUP.TXT
(The body of the text goes here)
^Z
```

The text will be stored on the default disk drive with the file name SETUP.TXT. To put it on a different drive, add the drive letter and ':' in front of the file name (with no space between).

#### DATE Command

The DATE command produces the current date, if you have a clock option in your computer. If you do not, it will produce the date you type in on boot up, which may account for some strange dates on some Victor specific software (I found a 1992 date recently - Hmmm!). If the DATE command is typed with no parameters, the date is returned as follows:

EXAMPLE 1: DATE is typed:

```
Current date is Tues 5/24/88
Enter new date:
```

Where: Pressing the return key bypasses the request for a new date.

The American format for DATE presents the date using the '/' as separator. The European format uses the '-' as separator. The American format is:

month/day/year

The European format is given as:

day-month-year (with the year in four digits)

#### DEL Command (ERASE)

The DELete command deletes the file(s) chosen. ERASE is a synonym for DEL, and will produce the same results. Wild cards may be used, as described under the COPY command. The complete file name and extension (if any) must be given for the command to work, unless parts have been defined under the rules for wild cards. Here's an example of DEL:

EXAMPLE 1: To delete the file on the default drive named POEM, from the default drive, type the following:

```
DEL POEM
```

EXAMPLE 2: To delete the file named ARTICLE.TXT from drive B, with the default drive as A, type the following:

```
DEL B:ARTICLE.TXT
```

#### DIR Command [Switches: /P, /W]

The DIR command lists the files in the current directory. Wild cards may be used to list groups of files, or to substitute '?' for unknown or variable letters in the file name. Following DIR with a drive letter (e.g.: DIR A:) lists the directory for the whole drive with DOS 1.25, or lists only the current directory (whether root directory or subdirectory) with DOS 2.11 or 3.1.

The files are listed with file name, byte size of the file and date and time last modified.

If the /P switch is used, the listing pauses after each screenful. If the /W switch is used, a wide display with the file names only (five per line), is produced.

A simple way to determine if a single file (for which you know the name) is in the current directory is to type the following:

EXAMPLE 1: To see if a file named SINGLFIL is in the current directory, type the following:

```
DIR SINGLFIL (if that is the way you shortened it)
```

The screen will show that one particular listing if it is on the current directory.

Directory paths to subdirectories, on the same or different disks, may be used with the DIR command from DOS 2.11 or 3.1. An example might be:

EXAMPLE 2: To show a directory for disk C, subdirectory DBASE from disk A, type the following:

```
DIR C:\DBASE
```

Where: You want a subdirectory located on drive C, called DBASE. C: is the drive designator, '\ ' is

a separator between the root directory and the named subdirectory. C:\ would represent the root directory itself. C:\DBASE\MAIL would represent the sub-sub directory MAIL, located in the subdirectory DBASE, which is located in the root directory of drive C.

#### REN Command

The REN command RENames a file. It is used as follows:

EXAMPLE 1: To rename a file named MAILLIST.DBF to MAILLST2.DBF, type:

```
REN MAILLIST.DBF MAILLST2.DBF
```

Wild cards may be used. As an example:

EXAMPLE 2: To change the extension from .TAX to .TXS on a group of files, type the following:

```
REN *.TAX *.TXS
```

#### TIME Command

The TIME command shows the time, and provides for changing the time, if you choose. The time change for DOS 2.11 and 3.1 does not permanently change the time if there is an on-board clock. The changed time will last only until the computer is shut off. An example of the format is:

EXAMPLE 1: To see the time, type:

```
TIME
```

and the response is:

```
Current time is 19:43:36.50
```

```
Enter new time:
```

Where: Time is given in hours, minutes, and seconds, separated by colons. (Seconds are given to .00 or .50 second.)

When entering a new time, it is not necessary to enter seconds. (However, time can be entered to .01 second.)

#### TYPE Command

The TYPE command is used to display a file on the screen. If the file is ASCII, the file will be readable on the screen. If the file is not, there will be character salad on the screen.

TYPE may be used with pipes and redirection. An example of a piped TYPE command follows:

EXAMPLE 1: To type an ASCII file (on drive A)

named README.DOC to the screen by one screenful at a time, type:

```
TYPE A:README.DOC | MORE
```

Where: The '|' symbol 'pipes' the message to the screen until it is full. MORE tells the TYPE program to do the same when a key is pressed.

We have already seen an example of a redirection of COPY. This merely means that we want the output of the COPY command to go somewhere other than to the screen (the default output). We can send the output of the TYPE command to the printer in the same way we sent the output of the COPY command. It looks like this:

EXAMPLE 2: To redirect a TYPEd file named A:README.DOC to the printer, type in the following:

```
TYPE A:README.DOC > PRN
```

That finishes the resident commands common to the three versions of DOS. Next let's look at the two resident batch commands. They are PAUSE and REM.

#### PAUSE Command

The PAUSE command pauses operation of the batch file until an action is taken by the operator. You might want to include PAUSE to give time to insert a data diskette into a drive. To make it clear what the pause is for, there is provision for displaying a message or comment built into the PAUSE command. An automatic (separate) message is displayed at the end of the PAUSE command. It says "Strike any key when ready...". Here is an example:

EXAMPLE 1: To use PAUSE with a comment, you might type something like this:

```
. a batch command goes here  
. a batch command goes here  
PAUSE Insert a formatted diskette in drive B
```

The display on the screen will look like this:

```
Insert a formatted diskette in drive B  
Strike any key when ready ...
```

Any key struck will cause the batch file to continue. At any point in the execution of a batch file, the execution may be aborted by typing ALT-C. A message is displayed - "Abort batch job (Y/N)?". Typing 'Y' aborts the job.

## REM Command

The REM command displays a comment (but does not stop execution, as PAUSE does) during execution of a batch file. The comment must be separated from the REM command by a comma, or by one or more spaces. DOS 2.11 and 3.1 provide an additional, alternate method for displaying comments. We will look at that one when we take up the ECHO command.

For those using DOS 1.25, there are two more resident commands. They are TOD and ?. The TOD command displays the date and the time. It may be called from DOS at any time. The ? command displays which drive is the system drive, if it is different than the default drive, and the COMMAND version number.

## ADDED RESIDENT COMMANDS COMMON TO DOS 2.11 AND 3.1

DOS 2.11 added a number of resident commands, bringing substantial added performance to the user. Of the sixteen new resident commands, six were added in support of the directory concept introduced in DOS 2.11. Here are the added commands:

### BREAK Command

The BREAK command controls how often DOS checks for ALT-C (^C). If BREAK is OFF, DOS checks for ALT-C while reading the keyboard or writing to the screen. If BREAK is on, DOS extends the checks to reading and writing to and from disk, as well. BREAK may be invoked from the DOS prompt, or it may be placed in the CONFIG.BAT or AUTOEXEC.BAT file. If it is placed in a .BAT file, its action is automatically invoked for as long as you are in DOS. Some programs provide for BREAK from within the program. Others mask it out, or do not use DOS at all; in which case, ALT-C does not work.

### CHDIR Command (CD) [Switch: /A]

The CHDIR command (Change Directory) requires drive and path information to work. It changes from one directory to another. An example would be:

EXAMPLE 1: To change directories from the root directory (or from anywhere) to a subdirectory named WORD on the same drive, type:

```
CHDIR \WORD or CHDIR WORD
```

Where: \WORD defines the subdirectory to change to.

EXAMPLE 2: To change directories from one subdirectory to a sub-subdirectory in a different subdirectory, type:

```
CD \DBASE\MINE
```

Where: CD is an optional form of CHDIR.

The /A switch displays the working (default) subdirectory on each drive.

### CLS Command

The CLS command clears the screen and displays the prompt and cursor at the HOME position in the upper lefthand corner of the screen. I put up with cluttered screens with DOS 1.25 for quite a long time before I was finally rescued with DOS 2.11. What a relief!

### CTTY Command

The CTTY command lets you change the device from which you issue commands. CTTY stands for ChangeTeLeTYpes, so you can see there is a history behind it. It may be used, as in the following example, to switch command operation to a remote terminal. That remote terminal might be in your office, but remote from you, or it could be at my house. by using CTTY you could (in part - there is more to it) operate my computer from your computer. Here is an example of how you would invoke CTTY:

EXAMPLE 1: To invoke CTTY and assign command to a computer port called AUX, type:

```
CTTY AUX
```

To move the I/O command back to the console and keyboard, type:

```
CTTY CON
```

### EXIT Command

The EXIT command exits the command processor currently in force and reverts to the next previous one, if one is in force. The command processor in DOS is COMMAND.COM, but some programs have their own command processors; and in DOS it is possible to have more than one level or 'shell' of command processor. When this occurs, you will usually be instructed in your software manual to use the EXIT command under the specified circumstances. Otherwise you will not run into it unless you tackle some moderately sophisticated programming.

## HISTORY Command (HI)

The HISTORY command records the commands you have used (up to 256 characters), and allows you to recall them and to pick from the list one you want to use again. Once picked, you may edit it in the usual fashion. If HISTORY is invoked without a number, the HISTORY file of commands is presented on the screen. It is invoked with a number to select a command from those stored. For a good typist, the time to access this command is probably at least as long as to retype the command. However, if your fingers are tired...

EXAMPLE 1: To select a HISTORY command, say the fifth one in the displayed list, type:

```
HI 5
```

## MKDIR Command (MD)

The MKDIR command is a near twin of the CHDIR command. MKDIR means MaKeDIRectory, and follows the same format as CHDIR. MD will always be used before CHDIR can be used. We must make a directory before we can change to it. As an example:

EXAMPLE 1: To make a subdirectory in the root directory of drive C: named BIRTHDYS, type the following:

```
MD C:\BIRTHDYS (this could be done from any drive, and does not need to be done from the drive on which it will appear)
```

EXAMPLE 2: To make a sub-subdirectory named BIRTHDYS under a subdirectory named FAMILY, and if you have already changed to the subdirectory FAMILY, you may type:

```
MD \FAMILY\BIRTHDYS  
or just  
MD BIRTHDYS
```

Where: The rule is that you may dispense with the current directory name by using the second form above.

## PATH Command

The PATH command tells DOS where to search, other than the current directory (which it will always search) for an external command or program. The search takes place in the order you specify in the path command, so if you want to speed up a search, plan to put the most likely sources

first in your path statement. An example of a path statement is as follows:

EXAMPLE 1: To set up a path statement, tell DOS where you want it to search, e.g.:

```
PATH = C:\;C:\DOS;C:\BIN
```

Where: C:\ is the root directory on drive C; C:\DOS is a subdirectory named DOS; and C:\BIN is a subdirectory named BIN. The directories will be searched in the order in which they occur in the PATH statement.

## PROMPT Command

The PROMPT command changes the DOS command prompt. We showed several examples of how prompts could be changed in the first article.

## RMDIR Command (RD)

The RMDIR command means ReMoveDIRectory. When you want to get rid of a directory, or even if you wish to change the directory name, the command to use is RMDIR. This command will ONLY work when the directory you wish to remove is completely empty. In addition, it cannot remove itself. This means you must:

1. backup all the files in the directory (if you wish to save them),
2. DELeTe all the files in the directory,
3. Change DIRectory to the parent directory of the one you wish to remove, and then
4. use RD.

RMDIR (RD) has the same format as MKDIR (MD) and CHDIR (CD). Now you can see how the three are related in the functions they perform.

## SET Command

The SET command sets a string of characters (a word or line of text) equal to another string for use later in a program. Setting one string equal to another merely means that the word(s) (could be a file name) on the right of the equals sign are substituted for the word(s) on the left. By using the SET command, the substitute value is stored, together with the information as to what to substitute it in place of. One example might be:

EXAMPLE 1: To SET the string C:\INC as a substitute for the string INCLUDE, type:

SET INCLUDE = C:\INC

Where: Everything to the right of the equals sign is to be substituted for INCLUDE, the string on the left.

#### VER Command

The VER command displays the version number of the DOS in use. This would be VER 1.25 or VER 2.11, etc.

#### VERIFY Command

The VERIFY command turns on or off the verifying of material being written to a disk. This is similar to the action of the /V switch used with COPY. The proper format is to type either VERIFY ON or VERIFY OFF, depending on which you want. If, and only if, an error is found in the process, an error message appears on the screen.

#### VOL Command [Switch: /C]

The VOL command displays the disk volume label or name, if one has been assigned. This can be a convenience in reminding you of the contents of a given disk (among many) by the disk name you assigned it. If you did not assign it a VOLUME name when the disk was formatted or subsequently, there will be none to remind you. The switch /C allows you to change the volume label.

You might want to try this DOS feature on a disk or two to see if it is something you want to use regularly. (I don't use it, but then I probably don't have enough filled disks to make it worth while yet.)

### ADDED RESIDENT BATCH COMMANDS

Five new resident batch commands, added in DOS 2.11, extended the usefulness of batch files greatly. This was the beginning of providing batch files with logical decision making capability. The five new commands were listed earlier in this article. We will review them here because they are true resident commands, but we will go into more detail on them in the last article on batch files. Now let's examine them to see what they can do for us.

#### ECHO Command

The ECHO command is a default batch file command which displays a line of text on the screen. Since the default condition for ECHO is ON, any one-line message occurring immediately after ECHO will be displayed on the screen.

An option is for ECHO to be turned OFF. When ECHO is OFF, as you might expect, any message following ECHO OFF is not displayed, except for a line which explicitly begins with ECHO ON:

EXAMPLE 1: To display only selected lines of text on the screen from a batch file, type:

```
ECHO OFF
ECHO You will see this line on the screen.
BUT YOU WILL NOT SEE THIS LINE
^Z
```

Where: The first line changes the default so that nothing will be displayed on the screen. This is modified by the second line, but only for one line of text (the line on which the echo appears). The third line demonstrates that this line, without an ECHO ON command, will not display on the screen.

Notice the difference in the effect of OFF as opposed to ON used with ECHO. When OFF is used, its effect holds for the entire batch file, unless modified a line at a time by beginning the line with ECHO ON. This technique, using ECHO, is much more flexible than REM, since you can present a message, or messages, interspersed with commands in the batch file; and you may do so without a series of REM's cluttering up the left side of the screen!

#### FOR-IN-DO Command

The FOR-IN-DO command (called FOR for short) is designed to allow you to perform one operation on a set of files:

EXAMPLE 1: To use the FOR command to copy a selected set of files from drive A to drive B, type:

```
FOR %% c IN (*.BAT) DO COPY A: B:
```

Where: The '%%' indicates a variable (more than one value or 'thing' can be assigned to it). It is followed by a character (not a number 0 through 9) to identify this use of the '%%'. The parentheses surround the name of the set of files with which you wish to work, and the rest uses commands we have discussed before.

This will copy all the .BAT files on drive A to drive B.

#### GOTO Command

The GOTO command is used to skip commands and go to a specified command and take up execution at that point. It

can also be used to loop back from the end (or some point in the middle of the batch commands in the file in use) and to take up the command execution cycle at or near the beginning of the file. GOTO is usually associated with LABELs as an indication of where to GOTO. We will look at an example which includes a LABEL, but will delve into the GOTO and its LABEL more thoroughly when we discuss batch files.

EXAMPLE 1: To use GOTO in a batch file:

```
ECHO OFF
ECHO If you want to quit, type X
IF-EXIST X GOTO END
ECHO My Name is Marilyn
:END
^Z
```

Where: The first message gives you a choice - enter X or not. The third line combines two batch commands, IF-EXIST and GOTO. The last line ":END" starts with a colon, the convention we earlier called a LABEL, a place name, a location for the program to jump to or GOTO. It must have a ':' in front of the LABEL name, or the computer will not recognize it as a LABEL. The second message will be printed to the screen only if an X is NOT entered.

The ability to set up conditions, test for them and decide what action to take depending on the results can be very handy in a batch file. It can be set up to safeguard you files while you are doing housekeeping chores, and many other things. Part of the fun is to devise new and useful ways to string these commands together for even more flexibility.

#### IF-EXIST Command

The IF-EXIST command was used in the example above, and its use should be clear from that.

#### SHIFT Command

The SHIFT command allows you to use more than ten variables (the limit otherwise) in a command. It is somewhat complicated, and is not used often. I will cover it in some detail in the batch file article, since it can be very useful.

#### ADDED RESIDENT COMMANDS USED ONLY IN DOS 2.11

Finally we will take up two resident commands, CAUX and

CLST. The CLST command appeared in the first article, since it is invoked in the CONFIG.SYS file as part of the machine setup. The CAUX command is similar to CLST. Both define drivers in the CONFIG.SYS file. But where CLST defines the 'List Device' driver (usually the printer, PRN), CAUX defines the driver for the AUXiliary Input/Output port.

EXAMPLE 1: Driver definition using CLST and CAUX is performed as follows:

```
CAUX PLOTTER
CLST PRN
```

Where: PLOTTER is a valid driver for a plotter connected to the AUX port. PRN is the default driver name assigned by DOS, and installed by default in your CONFIG.SYS file.

The device driver must have been loaded at, or before, the time the CAUX or CLST command invokes the driver:

EXAMPLE 2: Two ways to invoke these drivers:

1. DEVICE = PLOTTER  
CAUX = PLOTTER  
or
2. CAUX=\DEV\PLOTTER

Note that the first way requires two steps be entered, while the second way requires only one.

In the next article, we will explore the external commands, and how to use them.

\* \* \*

---

#### VICTOR COMPUTERS FOR SALE

- 4 Victor 9000, dual floppy disks, 256K RAM
- 1 Victor 9000, floppy/10Mb hard disk, 256 K RAM
- 1 Televideo personal terminal
- 1 Novell 47Mb hard disk
- 1 Novell File Server 6 station  
includes Netware 6B & cards

Lotus 123, Supercalc, Word Perfect, dBase II, Wordstar, Open Systems, Multiplan, and various utilities. Many extras including manuals and extra keyboard. All for a low price of \$3000. or OBO.

Contact: Carol, Cascadian Farm, Star Rt., Rockport, WA 98283 (206) 853-8185

\* \* \*

# A RAMDISC THAT SURVIVES RESET / Issue

[A recent flyer from Issue seemed of sufficient interest to warrant reproducing here. - Ed.]

Issue has recently been concentrating its attention on software products for the Victor Sirius/S1/V9000, in order to complement its existing range of high quality hardware products for this revered machine. Early this year Issue produced a RAMDISC programme for the Sirius which is sturdy and reliable but otherwise unexciting. The latest release of the Issue RAMDISC however has the unprecedented ability to restore itself to exactly the same state it was prior to a reset, either brought about by a 'warm boot' from the keyboard or a 'cold boot' from the reset button.

Using any ramdisc results in a dramatic increase in speed, as all loading of programmes, their respective overlays and data files are from fast, solid-state memory instead of slow, electro-mechanical devices like disc drives. (A classic example of this is found with Lotus 1-2-3, which on the Sirius takes 47 seconds to load from floppy and 3 seconds to load from the Issue RAMDISC). Essentially, a ramdisc programme reserves a section of RAM and configures it to appear as a disc drive to the Operating System. Numerous such programmes exist, however the 'fly in the ointment' as far as conventional ramdiscs are concerned is that if the computer hangs, or if a reset is required for any reason, the entire contents of this 'virtual' disc drive are lost. Thus, time is wasted reloading the ramdisc and valuable data can be lost forever.

Now Issue has removed this disadvantage. On re-booting a Sirius fitted with the Issue ROMs (released 2Q 1987) and running release 1.8 of the Issue RAMDISC, the user finds his virtual disc drive (drive C:, D: or whatever) completely unchanged, and work can immediately continue.

The only feature of the Issue ROMs which make this possible is the 'non-destructive' RAM test. This is a feature which other manufacturers would do well to emulate: almost all microcomputers test their memory on booting and in the process destroy the memory contents. It's almost like running CHKDSK and erasing the disc!

The Issue RAMDISC programme v1.8 is available now and costs £50.00 (inclusive). The Issue ROMs possess several other advanced features and cost only £36 per set (we think that fitting them to a Sirius is such a good idea that we keep the price artificially low). Users of earlier versions of the Issue RAMDISC can have their master discs updated for our standard software update charge of £5.00.

Other recent software releases include INDICATE, a memory resident Caps Lock Indicator for the Sirius (£25.00 inclusive), and the announcement of a coprocessor 'piggy back' board featuring an 8087 is imminent.

Celebratory Offer! RAMDISC & ROMs £72

U.S. \$ price = £ x 1.8 incl. air mail shipping. U.S. \$ check o.k.

Issue Postbus 3707 1001 AM Amsterdam Nederland  
Issue PO Box 222 Brighton BN13BR UK

\* \* \*

## POWER SUPPLIES IN THE V9 / George Anderson

Having had a session with disturbing results with power supplies in the V9 recently, I thought I'd lay out some of the results and conclusions while the experience is still fresh. File this for a rainy day when something in your power supply needs fixing.

V9's have power supplies installed by Victor, but manufactured by two different companies. They have been made by COMPOWER and by ASTEC. The ASTEC supplies were used on V9's with factory-installed hard disks, it seems. They may also have been installed in the later floppy-only machines. The ASTEC supplies are heavier duty than the COMPOWER supplies.

I put a hard disk into a V9 with a COMPOWER supply in it. The first drive was a low power, slow drive. (The drive didn't work out.) The replacement drive draws more power, and the COMPOWER supply doesn't like that much load. The result was that after an hour of operation, the screen slowly faded to black. A wait of 20 to 30 minutes with the V9 shut down for the supply to cool off, and the screen was as fresh and bright as before. BUT there was this really hot spot on the back of the V9, just toward the center of the back from the cooling slots. By bolting a small fan (same as the type now in the V9) to the slots, and blowing inward, the former hot spot was cooler than room temperature after 16 hours of continuous operation. This approach can be used to extend the life of a perfectly working power supply, and might warrant some looking into if you have an internal hard drive, added after it left the factory. It is not necessary with the ASTEC supply, but would extend its life also. The cooler they are, the longer they last.

\* \* \*





requirements of the printer's tractor, so I set this column at 0 also.

Once the general format is established with the .Y and .YT tables, install them at the top of the text ("FT" and "FY" on the command line). Put in a header line ("H" followed by the text you want for the title). Next, go through the entire document using the "V" command, to see if the tables are doing what you thought they would. Check now to see if the page breaks occur cleanly. Sometimes you will want to force a page break to keep a paragraph or tabular material all on one page. Do this by installing a ".E" dot command at appropriate locations.

**THE NITTY GRITTY:** Wasn't that just like fast-forwarding your VCR? Anyway, now you are ready to print. Go to the top of the document if you aren't already there. If you know how many pages you are about to print, use half that number. Round upwards to the next whole number if the page count is odd. For example, if you are about to print 33 pages,  $33/2=16.5$ ; 16.5 rounded up = 17. You must use a whole number. 17 will be used in the following example. If you don't know the page count, just use some large number in the batch command that you are sure is more than half the total pages in the document. Spellbinder will stop printing when it runs out of text no matter what number is used.

Type the batch command "**17/p/j**" <CR> on the command line. This tells Spellbinder to print one page, "jump" over the next page, and to do this 17 times. 17 pages later, all your odd-numbered pages from 1 to 33 are printed. At this point, rewind the paper (or remove it) and turn it over.

Reinsert the paper into your printer. Roll it back into position. You should now be staring at the blank backside of page 1. Use the "T" command to return to the top of the document. Now type "**16/j/p**" <CR> on the command line. This time, Spellbinder jumps over odd pages and prints the even pages.

If you did everything correctly, the document is now printed and titled on both sides of the pages, consecutively numbered. You can test beforehand by substituting "V" for "P" in the batch commands and checking the results on your monitor.

Now, as to Tables of Contents: print the Table of Contents **only** on a separate sheet of paper. As you View your reformatted document prior to printing it, pencil in the new page numbers on the printout. Transfer those correct(ed) numbers to the Table on screen. Print as directed.

For those of you who skipped ahead to the NITTY GRITTY, different strategies are required. One is to skip over the page on which the Table of Contents would normally be printed, modify the file with the proper page numbers, then roll back to that page and print the Table. Another strategy is to print the whole document, Table included, wrong numbers and all. Then, after you know the correct page numbers, modify the file, print the corrected Table on a separate page, and glue the correction over the old Table. It's inelegant but effective.

There it is. At best, you have very professional-looking printouts, consecutively paginated, justified and titled - the whole ball of wax. At worst, you have used 50% less paper and saved 50% of the space this document would otherwise occupy in a folder. And, of course, this technique can be used on any document whatsoever. Have at it, format freaks!

[Some additional suggestions from the Editor follow:]

1. Spellbinder (and probably most other word processors) have non-printing characters. If these occur in the text and special provisions are not made, these characters will be lost (not printed). In Spellbinder these are the following six characters: ! ^ ` { } ~. The method of dealing with these characters is to mark them for enhancement, then specify no enhancement. No enhancement is obtained either by using the in-line command !7 or by using 7 for the special character entry in the Y table.
2. In text containing tables, the columns generally will be ill-aligned if either justification or proportional spacing is used. Therefore, these entries in the Y table should both be 0 for the tabular portion of the text.
3. The suggested Y table entries for maximum and minimum space between words (15 and 5; that is, 1.5 to 0.5 character spaces) produces rather neat looking text, but leads to excessive hyphenation. This is undesirable on two counts: it involves extra work in performing the hyphenation and leads to reduced ease of reading. [I normally use entries of 35 and 7. While this occasionally leads to wide spaces, it rarely requires hyphenation. As a specific example, the original text of this article required ten (end-of-line) hyphens, when entries of 15 and 5 were used; no hyphenation (as shown above), when 35 and 7 were used.]

The above points are not unique to back-to-back paging, but apply to any use of Spellbinder for printing ASCII text such as .DOC or .TXT files.

\* \* \*

# THE LIBRARY REPORT / John Knobelsdorf

On May 4, 1988 we released PD disk #96. On June 1, 1988 emulators. Be sure to read the DISKID file on each disk we released PD #97. These disks are in the Chapter set and check the descriptions before you try to use a and available for copying at the monthly meetings. They particular program. Also remember that some of the files also may be ordered thru the mail in the usual manner. on the diskettes are in compressed form.

The files on the new disks came from the PUB and BAVUG Use NJSQ (on PD #73) to "Unsqueeze"  
bulletin boards. Some of the files will require the use of a Use LJE " to "Unlibrary"  
Victor VI, Plus-PC, a PC-Compatible or one of the various Use PKXARC (on PD #87) to "Unarchive"

Information about PD #96 follows:

DISKID96 .TXT	5760	Information about the files on this disk PD #096	
3CT .ARC	9344	Pun on "TreeCount"; nice display of filesize info	
85010 .	896	VPC Screen resolution	Victor Bulletin
85011 .	1408	HD Park for Plus PC	Victor Bulletin
85012 .	896	256K RAM chips for VI	Victor Bulletin
85013 .	896	VPC Boot switch settings	Victor Bulletin
85014 .	1024	Audio Tool Kit	Victor Bulletin
85015 .	1408	VPC Bugs	Victor Bulletin
85016 .	1536	Using SEARCH for Backup & Restore	Victor Bulletin
85017 .	2176	VI and Plus PC to Diablo cable date	Victor Bulletin
85018 .	1792	Removing Plus PC from a V9000	Victor Bulletin
AFILTER .ARC	12160	Convert WP Text Files v. 1.7.2 - Great	
ANSI .ARC	5760	ANSI Driver for Victor 9000	
ANSIHLP3.ARC	6656	ANSI documentation	
ARC521 .COM	58368	06-26-87 Really the latest ARC from SEA	
ARCE31B .ARC	11264	Buerg's ARChive Extractor v3.1b (inc. squashing)	
ARCF106 .ARC	8064	Find text in ARChives by V. Buerg v1.06	
BIOS211 .ARC	7808	Victor 9000 BIOS description	
COVER .ARC	2560	Make diskette cover labels with file names	
DELIFARC.ARC	11776	Delete file(s) if and only if ARChived (by JN)	
DRVTST .COM	17664	Lock out bad sectors on Hard Disk	
GETSCRN .COM	512	Set aside area for V9 graphics	
GMA5M .ARC	2944	Set and Clear memory area for Graphics	
GRFXRTNS.ARC	5760	Assembly language modules for the Victor 9000	
HEADTAIL.ARC	26880	Displays n lines of head, tail, or body of text.	
185004 .	1024	Accounting Plus conversion	Victor Bulletin
185005 .	1536	Victor Field Diagnostic Software	Victor Bulletin
IBIOSDOC.ARC	12160	IBMBIOS simplified by Fuzzy Bear	
IBMBIOS .A41	54272	IBM Emulator w/ Hirner's keyboard	
KILINARC.ARC	16384	clears directory of files that are safely arced	
LINENUMB.ARC	10496	Adds line numbers to text; with doc and TP source	
LPTC400 .ARC	22912	Redirect printer output to file (with source code)	
MAKEREAD.ARC	2048	Turns text file into com that displays the text.	
MONOP64 .ARC	65152	Monopoly Game; needs emulator	
MXSET .ARC	15104	MX printer setup routine - nice	
NUM2 .ARC	7680	Adds line numbers to text file; easy to use	
PC-RR .ARC	59392	PC Model Railroad - a game - needs emulator	
PRNTFONT.EXE	17792	no description available	
Q&A .TXT	12800	Common problems encountered by V9 users	

SCAVENG2.COM	17536	Check hard disk for bad sectors
TYPERS.ARC	10880	Turn computer into a typewriter for envelopes, etc.
VIC DISK.DAT	12672	V9 Disk Routine Data
VPDSW.ARC	5632	Print Sideways on Epson
VTREE.ARC	6784	Graphics display of directories
XC-TOOL.ARC	9728	Counts words/lines/chars; alters attributes;.....

567296 Bytes in      45 Files.      14336 Bytes free.

Information about PD #97 follows:

DISKID97 TXT	4096	Information about the files on this disk PD #097
BITS.ARC	27264	Victor Graphic Setup
BXD21.ARC	26752	P D Symbolic Debugger
COLEX.ARC	14208	Displays "columns"/words in specifiable order
DIRSTATE.ARC	11648	Maps directories and files with sizes and totals
EMIBMWP.ARC	5632	KB modified to use with WordPerfect
FBR161.ARC	15104	Hard disk backup per V.Buerg, Aug 87-Use Emulator
HIDE RO.ARC	384	Change File to Hidden or Read Only & Back Again
LOC100.ARC	24448	File location/text search program
LREVERSE.ARC	12032	Mirror-image of text lines; why? see doc.
M106DOC.ARC	14976	Doc. for the MV replacement - written by Bob Stout
MMVB106C.ARC	42368	An extensive MoVe replacement - cross disk boundaries
NARC13.ARC	52224	Great Un-Arc program with menus and help
PKARC35P.ARC	1792	Patch defaults in PKARC v3.5
PKX35A35.EXE	73728	Superfast ARChive utilities v3.5
PRN2FILE.ARC	18176	Redirect Printer Output to File
PROSET34.ARC	16256	Utility for IBM Proprinter w/menus
PWMENU.ARC	140672	Very nice menu driven DOS shell. Has good doc.
REGISTER.ARC	25344	Check Register & Home Budget Control (Great)
SPS13.ARC	3712	Selective Screen Print - IBM
TD2.ARC	13312	Total Directory (DOS 2.x)
TICKTOCK.TXT	4352	DOS problems with filename "Clock".
V20FILES.ARC	23552	Format/Diskcopy For V20/V9000

572032 Bytes in      23 Files.      30720 Bytes free.

\* \* \*

(Continued from page 13)

However, Big Blue chose to go in a different direction with their format for the 1.2-megabyte High Density drives, first used on the PC AT. To do this, they had to contend with sectors laid out with boundaries along radius lines extending across all the tracks. This means that all the sectors closer to the center of the disk are shorter than the ones further out. To get the same "bits per inch" in each sector when some sectors are shorter physically than others, means that the sectors closer to the center must have their bits squeezed closer together than in the sectors further out. This is hard to do, using DSDD (Double Sided Double Density) disks. Big Blue, therefore, decided to get the floppy disk industry to make a special type of disk with a different magnetic coating on it. These are marked DSHD, standing for Double Sided High Density, rather than Double Density.

Most companies making disks produce both types today, but those that have the HD in their designation cannot be used on drives other than those especially designed for it. The magnetic material is 'too hard' (to change) for the write heads in regular Double Density drives. The drives in the V9000 are basically Double Density drives. They are used by Victor in a special way to provide greatly enhanced storage capacity for us lucky ones. But you get to buy the more widely made, cheaper (but not poorer quality - just different, and made that way for a reason) disks identified with the DSDD on the box. Maxell makes one of the best disks on the market, but with the improvement in manufacture over the last three years, many other brands are very close to the same quality. Too bad the markings on the boxes are so ambiguous.

\* \* \*

## CONTRIBUTORS

To facilitate readers contacting authors, with this issue we are starting the policy of providing author's addresses and phone numbers (unless requested otherwise). Here are this issue's contributors:

George Anderson,  
1571 El Camino #9,  
Mt. View, CA 94040  
(415) 962-9555

Franz Hirner,  
1493 Beach Park Blvd, Box 180,  
Foster City, CA 94404  
(415) 349-3602

John Knobelsdorf,  
Route 2, Box 103,  
Bertram, TX 78605  
(512) 355-2084

Andrew Rudiak,  
132 Villa Court,  
Lafayette, CA 94549-2042  
(415) 932-5580

\* \* \*

## FROM THE BOARDS

We regret that, even with a larger-than-normal 28-page issue, space was not available for FROM THE BOARDS.

\* \* \*

## VictorTalk

### Address inquiries to:

BAY AREA VICTOR USERS GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

### Subscriptions: Carol Hirner

Editor: Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

**VictorTalk** is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in **VictorTalk**. Copyright 1988 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

**VictorTalk** is an independent publication not affiliated in any way with Victor Technologies, Inc. **Victor** is a registered trademark of Victor Technologies, Inc.

**Submit Articles** on MSDOS disks in ASCII or Spellbinder format (either single-sided or double-sided, in Victor or IBM format), with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

**BAVUG BBS** is on line 24 hours a day: (415) 574-7914, Franz Hirner, Sysop.

## Yes, I want to join.

Enclosed is a check/money order for \$35.00 (foreign membership, \$40.00 in U.S. dollars) for my 1988 Victor\*Group membership.

Name \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

Make checks payable to:

## VictorTalk

1493 Beach Park Boulevard, #180  
Foster City, CA 94404

City \_\_\_\_\_

State \_\_\_\_\_ ZIP \_\_\_\_\_

Country \_\_\_\_\_

Phone \_\_\_\_\_

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
State _____	
City _____ & Zip _____	
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No. Disks	Vol. No. Disks	Vol. No. Disks	Vol. No. Disks
4 2 61-64	5 1 78-81	5 5 87-90	6 3 96-97
4 3 65-71	5 2 82-84	5 6 91-92	
4 6 72-77	5 4 85-86	6 2 93-95	

- |    |    |    |    |    |    |    |    |    |    |    |    |    |     |                               |     |                               |    |    |    |  |  |
|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-------------------------------|-----|-------------------------------|----|----|----|--|--|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14  | 15                            | 16  | 17                            | 18 | 19 | 20 |  |  |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34* | 35*                           | 36  | 37                            | 38 | 39 | 40 |  |  |
|    |    |    |    |    |    |    |    |    |    |    |    |    |     | *(34 & 35 are a two-disk set) |     |                               |    |    |    |  |  |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54  | 55                            | 56* | 57*                           | -- | 59 | 60 |  |  |
|    |    |    |    |    |    |    |    |    |    |    |    |    |     |                               |     | *(56 & 57 are a two-disk set) |    |    |    |  |  |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74  | 75                            | 76  | 77                            | 78 | 79 | 80 |  |  |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94  | 95                            | 96  | 97                            | —  | —  | —  |  |  |

1988 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disks ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, sets of 6, \$12.00 /set ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_

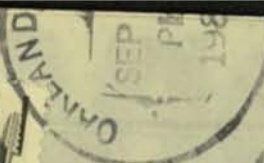
1988 membership fee \$35 (receive all 1988 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR\*GROUP

1493 BEACH PARK BLVD, #180

FOSTER CITY, CA 94404



FIRST CLASS MAIL

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP

1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
State _____	
City _____ & Zip _____	
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Un-archived versions are available where "A", "AB", or "ABC" appears below number. If originally released (archived) version is desired, cross out corresponding "A", "AB", or "ABC". (Original releases of all other disks contain no archived files.) Note that pricing below is per "disk number" rather than per "disk" (re-released unarchived "disk numbers" may be comprised of one disk (A), two disks (AB), or three disks (ABC). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No. Disks	Vol. No. Disks	Vol. No. Disks	Vol. No. Disks
4 2 61-64	5 1 78-81	5 5 87-90	6 3 96-97
4 3 65-71	5 2 82-84	5 6 91-92	6 4 98
4 6 72-77	5 4 85-86	6 2 93-95	

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56*	57*	--	59	60
													A						AB
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
												A					AB	AB	AB
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	—	—
AB	AB	AB	AB	AB	AB	AB	ABC	ABC	AB	AB	ABC	ABC	AB	ABC	AB	AB			

\* 34 & 35 are a two-disk set; 56 & 57 are also a two-disk set

1988 Victor\*Group member: No  Yes  Date joined/renewed \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, sets of 6, \$12.00 /set ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_

1988 membership fee \$35 (receive all 1988 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_



## EXPLORING MS-DOS, Part 3 / George Anderson

In this installment we are going to examine external commands.

What we mean by 'external' is any file in DOS with an extension of .COM, .EXE or .BAT. All .COM, .EXE or .BAT files must be named to invoke them, and require access to the directory in which they reside (directories are in DOS 2.0 or above). Otherwise, the access must be directly available to the operation. If access is not set up, you will get one of DOS's famous error messages, in this case: 'Bad command or file name'. It will not be necessary to use the extension when typing the name of any of the external commands, just as it is not necessary to use the extension when invoking internal commands.

In this article, we will concentrate on the .COM and .EXE commands included in DOS, and wait until the final article to discuss .BAT files, since these files are the ones most commonly generated by the user, and therefore are more easily described separately. One point should be made clear now to avoid confusion when using external commands.

It is possible to have .COM, .EXE and/or .BAT files with the same name, and there is nothing wrong with having the same name. However, it is helpful to note that if you do have files with the same name, DOS will run only one of them at a time. The DOS default is to scan the directory for names that are the same, and to run them in the following precedence:

- 1) .COM
- 2) .EXE
- 3) .BAT

Therefore if you have a SORT.COM and a SORT.EXE file, DOS would run only the SORT.COM since it is first in the line of precedence listed above. Even if you added the extension to the file name, to differentiate it from a file having precedence (and both are located in the same directory), DOS will run the file having precedence. If both files are NOT located in the same directory (i.e. at the same level) DOS will run the one specified in the current directory, provided no path exists to the file having precedence. So, by proper location in a directory, with no path to a file of the same name, a file having lower priority can be invoked, if you desire.

DOS internal commands, we found, were read into RAM at the time the computer is booted; but DOS external commands are stored on disk, and must be read into RAM when needed. For DOS to place the command into RAM, it

must be available in or through the current directory. If the command is not in the current directory, a PATH command is needed to tell the computer where to look. The PATH command may be entered manually, or it may have been entered in a general PATH command stored in AUTOEXEC.BAT, and invoked when the computer was booted. For example, if the CHKDSK command is needed, and it is in the \DOS subdirectory of disk C:, the proper PATH command to invoke CHKDSK from C:\ manually would be:

EXAMPLE 1: C:\DOS\CHKDSK

Where: C: defines drive C:, \DOS defines the path as being through the subdirectory named DOS, and CHKDSK is the external command file desired.

EXAMPLE 2: To define the same path from an AUTOEXEC.BAT file, enter this line in the .BAT file:

```
PATH C:\DOS
```

Where: The C:\DOS means the same as it did in example 1, above. Done this way, we would enter only the file name CHKDSK when calling for this command. The PATH command, stored in AUTOEXEC.BAT would furnish the rest of the required path data to the computer.

NOTE: As an alternative, the PATH command shown in Example 2 could be entered directly from the DOS command line prior to entering CHKDSK (also on the command line). It would then look like this:

EXAMPLE 3: To enter the path command directly on the command line, type:

```
PATH C:\DOS  
CHKDSK
```

### REDIRECTING COMMANDS

In DOS, input usually comes from the keyboard and output usually goes to the screen. It does not have to work this way. You have the option to redirect either the input or the output (or both). For example, you can take input from a file, perform an operation on it, and send the output to

the printer; or you can get input from a file and send the output to another file. The two command signs you will use to redirect input and output are '<' and '>'. These are called the lesser-than and greater-than signs, or sometimes just the left and right corner brackets. The left corner bracket is used to get input from someplace. It is placed to the left of the name of the someplace. The right corner bracket is used to send output to its destination. It is placed to the right of the named source. An example will help understand how to use these signs with the proper commands.

EXAMPLE 1: To sort a list of names already in files on disk, and have the results show up on the screen, type the following:

```
SORT < NAMEFILE
```

Where: SORT is the command to sort the specified material, and < NAMEFILE is the source of names in a file on disk to be sorted. Note that the '<' sign means 'take the source material from NAMEFILE'.

EXAMPLE 2: The operation of Example 1 may be expanded to send the sorted results to a file saved on disk. To do this, type the following:

```
SORT < NAMEFILE > SORTNAME
```

Where: The '>' sign sends the output of the sort to a file on disk named SORTNAME. DOS opens a new file with the name SORTNAME if it does not already exist. If it does exist, it will write the sorted names to the existing file, and the existing file is overwritten in the process.

EXAMPLE 3: To obtain a printed copy of the directory, redirect the output from the screen to the printer by typing the following:

```
DIR > PRN
```

EXAMPLE 4: If you want a permanent record of the directory on disk in a file named DIRECTORY, type the following:

```
DIR > DIRECTORY
```

EXAMPLE 5: To add the directory information to an existing file DIRECTORY, type the following:

```
DIR >> DIRECTORY
```

Where: The second right corner bracket makes it a command to APPEND the directory to the file rather than to overwrite it. If the file does not exist, DOS creates it.

## PIPES AND FILTERS

Pipes are a way of putting more than one command on a line, and getting them to perform a sequence of operations. It is like a string of commands, with the PIPES symbol '|' linking them together. One common use for PIPE is to limit the data sent to the screen to a screenful at a time, as in the TYPE command. The command limiting the data to a screenful is MORE.

EXAMPLE 1: The PIPE symbol, '|', links the TYPE command with the MORE command to give the action desired. It looks like this:

```
TYPE NAMELIST | MORE
```

Where: The TYPE command produces the typing on the screen, '|' links commands, and MORE causes the typing to pause after each screenful. More than one command may be linked with '|'.|

EXAMPLE 2: To produce a sorted directory and send it to the file DIRECTORY on disk, try the following:

```
DIR | SORT > DIRECTORY
```

Filter commands are:

```
FIND Searches files for specific text
MORE Limits screen displays to one screenful at
a time
SORT Sorts file contents alphabetically
```

We will take these up in more detail later.

With these few preliminaries out of the way, let us proceed to look more closely at the DOS external commands:

NOTE: The DOS version containing the commands listed are shown by an 'X' in the appropriate column:

```
Column 1 = DOS 1.25
Column 2 = DOS 2.11
Column 3 = DOS 3.1
```

DOS Version 1 2 3

ASSIGN		X	
ATTRIB		X	
BACKUP		X	
CHKDSK	X	X	X
COMMAND	X	X	
COMP		X	
CONCAT	X	X	
CPM		X	
DCOPY	X		
DISKCOMP		X	
DISKCOPY	X	X	
EXE2BIN		X	
FGREP	X	X	
FILCOM		X	
FIND		X	X
FORMAT	X	X	X
GRAFTABL		X	
GRAPHICS		X	
JOIN		X	
KEYBFR		X	
KEYBGR		X	
KEYBSV		X	
KEYBUK		X	

DOS Version 1 2 3

LABEL		X	
LS		X	X
MODCON		X	
MODE		X	
MORE		X	X
MV		X	X
PARK		X	
PORTSET		X	
PRINT		X	X
RDCPM	X	X	
RECOVER		X	X
RESTORE		X	
SDCOPY		X	
SEARCH		X	X
SELECT		X	
SHARE		X	
SORT		X	X
SUBST		X	
SYS		X	X
TAIL		X	X
TREE		X	
UNIQ		X	
WC		X	X

BACKUP  
JOIN  
FORMAT

RESTORE  
SUBST  
DISKCOPY

LABEL  
PRINT

Also, since DOS is evolving, ASSIGN will have limited use in the future. For those who want to stabilize their use of commands so that they will have to make fewer changes in the future, use SUBST in place of ASSIGN, and you will be compatible with planned changes in future DOS as well as having a SUBSTitute for ASSIGN now.

### ATTRIB

The ATTRIB command, by itself, displays the attribute for the file named in the command. However, this is the least that ATTRIB will do for you. By adding + or - r, or + or - a, ATTRIB will set the read only status bit on or off, and it will set the archive status bit on or off.

What does all that mean? It means that you have control over the status of your files, and have an easy way to manipulate or change that status as you need. For example, you want to make a clean backup copy of your WordStar file disk, but you don't want those pesky .BAK files anymore. One simple way to delete the .BAK files while copying all the pertinent files is as follows:

### ASSIGN

ASSIGN substitutes a drive letter designation you select for the existing letter designation of a drive. This is most commonly done when a program makes assumptions about the drive designators to use for program and data, and these assumptions do not match your operating configuration. For example, some programs assume that the program disk will be in drive A:\, and the data disk will be in drive B:\. When a V9 has an internal disk drive, it is usually called drive A:\, and the floppy drive is called B:\. If you want both program and data to run from the hard disk, use ASSIGN to temporarily redirect calls for the B:\ drive to the A:\ drive.

EXAMPLE 1: To change the drive designator temporarily, type:

```
ASSIGN B = A
```

This will redirect the computer to look on drive A:\ every time the program calls for drive B:\.

NOTE: ASSIGN makes no note of what kind of device the assignment was made on. It cannot distinguish when certain commands will work on the new assignments and when they will not. To avoid trouble, do not use the following commands with ASSIGN:

EXAMPLE 1: To set the archive bit on the files to be saved from disk A:\ to disk B:\, and to 'unset' the archive bit on the .BAK files, type:

```
ATTRIB +A A:*. *
ATTRIB -A A:*.BAK
COPY A: B:
```

Where: The +A sets the archive bit ON, and the -A sets the archive bit off. The COPY command then copies those files with the archive bit set ON.

EXAMPLE 2: To set the READ ONLY bit ON, on a file you want to protect from inadvertent overtyping, type:

```
ATTRIB +R A:PERSONAL
```

Naturally, when you do want to update the file, just rerun the command, but with a ^- instead of a ^+ in front of the R.

### BACKUP

The BACKUP command is used to back up files from the fixed disk to a floppy disk or to another hard disk. Depending on the switches used, this command can back up all files or selected files. This command will tell you when a

floppy disk is full, and prompt you for another disk with which to continue the backup. BACKUP will 'splice' together sections of the last file recorded on the current floppy disk with the balance of that file on the next disk used. This is not done when using the COPY command, so watch it. The BACKUP command also takes care of keeping track of which files are in which subdirectories, and the names of the subdirectories while making its copies. It can then recreate the same subdirectories with their proper files installed.

One drawback to this command is that the backed-up files cannot be read directly from the backup disks with normal DOS commands. There are some utility programs which will read them directly and allow you to use them without having to RESTORE them to another disk or to the hard drive. However, this is not the intended way to use the BACKUP/RESTORE command pair. Normally one would have RESTORED the BACK(ed)UP files to their original location, and would proceed from there to use them in exactly the same way one had before the BACKUP procedure was used.

The basic command format looks like this:

EXAMPLE 1: To BACKUP the A:\ drive to floppy disks on drive B:\, type:

```
BACKUP A: B: (optional switches go here)
```

Where: A: is the SOURCE drive (drive containing the files to be backed-up), and B: is the TARGET drive (where the BACKUP files will go).

A more practical example might be:

EXAMPLE 2: To BACKUP all files in A:\UTIL to B:\, type:

```
BACKUP A:\UTIL B:\
```

A great deal of flexibility is gained with BACKUP by use of the optional switches. Let's look at what those switches can do for us. The switches available, and their functions are:

- /A Adds the files to be updated to the files already on the backup disk without erasing the existing files.
- /D Backs up only those files having a date later than the date you enter.
- /L If with file name, creates a log entry of the backup in the file specified - if without, creates a backup log file (BACKUP.LOG) in the root directory of the disk being backed up.

- /M Backs up only those files that are new or have been changed since the last time a BACKUP was performed.
- /P Packs the maximum number of files onto the floppy disk. It does so at the possible expense of compatibility with BACKUP/RESTORE.
- /S Gets everything, from the level you specify, down, including the subdirectory names and contents - if you do not specify a starting subdirectory, DOS default is the whole hard disk.
- /T Backs up only those files having a time later than the time you enter.

EXAMPLE 3: To backup all the files on a hard disk, A: to floppies on drive B:, and to preserve the subdirectory divisions as well, type:

```
BACKUP A: B: /S
```

Where: /S preserves the subdirectory organization of the hard disk, A:, so that a subsequent RESTORE command will place everything back on the hard disk in exactly the same order as before.

The balance of the switches should be easy to understand from Example 3; and the description of their function, above.

Files named BACKUP.LOG (created using the /L: switch) contain several bits of potentially useful information. The first line tells you when the backup took place (date & time); and each backed-up file is listed by file name and location (which backup disk).

NOTE: If the commands ASSIGN, JOIN or SUBST have been used on the back up source drive, RESTORE may not work when you put the files back on the source disk. To avoid trouble, do NOT use BACKUP if the commands above have been used.

When BACKUP finishes, it produces an Exit Code. These codes are what are called 'errorlevel codes'. Such codes are useful in advanced DOS applications for instigating some further action or processing, based on which code was returned. Other than that, the code is an interesting sidelight on the resources DOS offers (something for everyone, sort of). The exit codes are contained in your DOS manual, should you want to dig into them.

#### CHKDSK

The CHKDSK command looks at the specified disk, and checks to see that the format of the disk sectors and their

content are as expected by DOS. It reports on the number of bytes on the disk checked, the number of bytes used in hidden files, directories and user files; and the number of bytes available on the disk. It also reports on the amount of RAM memory in the machine and the amount currently available after subtracting all overhead functions occupying a portion of RAM.

CHKDSK will accept a file name after the command, and will then proceed to give a status report for both the disk as whole and for the file named. If you type 'CHKDSK \*.\*', it will give you a report on whether each file is or is not stored contiguously on the specified disk, and give the number of non-contiguous blocks involved for each file.

All of that reporting is pretty handy, but CHKDSK does even more. If a defect in the files is found, CHKDSK reports that, and gives you the option of trying to fix it. The most common error it reports are 'Lost Clusters'.

A cluster is a group of one or more sectors, usually two sectors. A Lost Clusters error message may occur if there are open files on the disk (not properly closed files). This usually occurs when someone has taken a disk out of the drive without giving the program (or DOS) a chance to perform the automatic closing routine. CHKDSK will report the problem (you may not be able to access files affected until a repair has been made), but will not automatically fix it. To fix the problem, re-invoke CHKDSK with an '/F' switch. This tells CHKDSK to proceed automatically to fix the error. When it has finished, CHKDSK reports the results of the 'fix'. To streamline the operation, CHKDSK accepts the '/F' switch at the time it is invoked.

CHKDSK also accepts a '/V' switch. This switch provides a history of the action CHKDSK takes during its operation.

NOTE: CHKDSK in DOS 1.x does not offer the /F and /V switch options.

There are other services performed by CHKDSK, but they are invoked only seldom. If an occasion occurs in which a crosslinked file is identified, the allocation (of disk space) is incorrect, or there was a disk error during read or write of the File Allocation Table (FAT), or allocation units become lost, you may find that reading the manual at that time will clarify your options for correcting the problem using CHKDSK. The main thing to remember is that there is help in this command for correcting certain types of problems if and when they arise.

## COMMAND

This command loads a new copy of COMMAND.COM into RAM. It is most often used in connection with batch files, and we will discuss COMMAND further with batch files.

## COMP

COMPare is a command which does just that - it compares two files to see if they are identical. The files can be anywhere; on any drive or in any directory or subdirectory. COMP will report the differences it finds with a message that says 'Compare error at offset nnnnnn', meaning that it found a difference at so many (nnnnnn) bytes into the file. Provisions have been made to report up to 10 differences on a pair of files. Beyond that, it gives up, and the next command action is performed. If the files are not the same length, COMP quits, and displays a message 'Files are of different sizes'.

If files are located on different drives, or in different directories, it is important to assure that a PATH has been specified to allow both files to be accessed. An example will show the most common PATH designation:

EXAMPLE 1: To compare two files - one on drive A:\, in subdirectory \CHARTS, and named TREASURE; the other on drive A:\, in the root directory, and named TREASON, type the following command:

```
COMP A:\CHARTS\TREASURE A:\TREASON
```

## CONCAT

The CONCAT command combines two or more files and displays them on the screen. This command might be used to combine a phone list with an address list - each kept in a separate file on disk. The combined files would be displayed, but would not be saved to disk. They may be saved to disk by adding the '>' symbol to redirect the output, and then specifying a file name to which to send the redirected output. The command would look like this:

EXAMPLE 1: To combine two files and view them on the screen, using example file names, type the following:

```
CONCAT A:\ASCII\ADDR.TXT A:\ASCII  
\PHONE.TXT
```

EXAMPLE 2: To send the same combination to disk instead, type the following:

```
CONCAT A:\ASCII\ADDR.TXT A:\ASCII  
\PHONE.TXT > COMBIN.TXT
```

Another use for CONCAT is to create a file. You would simply precede the file name you want to create with CONCAT >, and the new file with your choice of name will be created.

## CPM

CPM is an external command contained in DOS 2.x which invokes the CPM-86 Emulator, and runs CPM-86 programs under DOS. It requires that the CPM-86 disk be transferred to an MS-DOS disk using RDCPM. CPM is invoked by typing CPM, a space, the normal CPM-86 command line, and 'Return'.

## DCOPY

DCOPY is an external command used in DOS 1.x only. It is the approximate equivalent of the internal command DISKCOPY found in DOS 2.0 and above. Like DISKCOPY, DCOPY formats the disk while copying the entire contents of one disk to another. DCOPY will not work with a hard disk. An example shows the DCOPY format:

EXAMPLE 1: To copy a file named PHONES from drive A: to drive B:, Type the following:

```
DCOPY PHONES
```

If no source and target drives are specified, DCOPY prompts the operator to designate the source drive (in this case, A:). Pressing the space bar initiates the copy procedure, copying the file PHONES from drive A: to B:. This is the safer procedure with DCOPY. If the drives were identified, DCOPY would start before the operator could safely remove the system disk and replace it with the source disk.

On copy completion, DCOPY gives the operator the option to copy another. If you want to exit DCOPY, press 'Return'. When source and target drives are designated, as in Example 2, the file specified is copied, and you are returned to the DOS prompt.

EXAMPLE 2: To specify the source and target drives:

```
DCOPY A:PHONES TO B:
```

There are three option switches for use with DCOPY. They are:

/C Counts and gives a soft error total. Victor recommends manual retries if more than 15 soft errors occurred during the operation. DOS 2.0 and above provides for 'retries' to try several times to format the spot in question. If it doesn't succeed, it proceeds to format the rest of the diskette, and to store information identifying the bad spot so as to avoid using it later. It then reports the number of bytes involved in the bad

spot, at the end of the formatting. although the methods are different, it is safe to use disks formatted under either method, but the operator must decide how many soft errors he or she will accept. Using disks of recent manufacture, it is not likely to occur.

/E Gives a soft error total, together with the locations of the soft errors.

/Z Shows Zone information for the disk; i.e., zones associated with changes in the motor speed - there are eight of these. Gives size of tracks and gaps (the space in bytes between the tracks).

These switches are added after the drive designators are given.

## DISKCOMP

The DISKCOMP command is used to compare the contents of one floppy disk with the contents of another. If you want to compare the first sides only, use the '/1' switch after the command.

DISKCOMP may not work with the COPY command. If you will remember, a source file may be fragmented on the disk and not all blocks are contiguous. When the COPY command is used to copy such a source, it tries to place all the blocks in contiguous order. Even if it doesn't accomplish this, it is probable that the order and separation in the copy will be different than the source. When this happens, DISKCOMP will see the different order or spacing as a file that doesn't compare one-for-one with the source, and will produce an error message saying:

```
Compare Error(s) on  
Track nn, Side x
```

where nn is the Track number, and x is the side the error was found on (either 0 for side 1, or 1 for side 2). A successful comparison generates this message:

```
Diskettes compare OK
```

DISKCOMP may be used on a single drive system. It means shuffling disks to do it, but the results are the same.

## DISKCOPY

The DISKCOPY command has different switches in DOS 3.x than in DOS 2.x. They are described separately below. The DISKCOPY command copies the entire contents of one floppy disk to another floppy disk. It does this differently than COPY. COPY copies files, but it does this by looking at the internal record of locations on disk that each file contains, and takes each block of data in the order given, no matter in what sector the data block is located. On a clean disk this results in a file copied so that all blocks of

data are contiguous.

DISKCOPY, on the other hand, starts at the beginning of the disk and copies each sector in turn, regardless of its content. The result is a carbon copy of the source disk, with no rearrangement of the data blocks between source and target disks. For that reason, DISKCOPY will overwrite any data already on the target disk. Do not use DISKCOPY when you want to add files to the disk.

DISKCOPY offers prompt screens if you enter the command without parameters. In DOS 2.x, drive letters are not used; the choice of drives for source or target are given as 'Left' or 'Right' drive. Abbreviations are acceptable. You may substitute 'L' for 'Left' and 'R' for 'Right'. 'T' may be substituted for 'To'. An example shows the choices:

EXAMPLE 1:

DISKCOPY R TO L

DISKCOPY RTL

DISKCOPY RL

Each one of the command lines are equivalent.

In DOS 2.x, the option switches have the following definitions:

- /C Counts the total number of soft errors found.
- /E Gives location of soft errors together with the total of all soft errors found.
- /Z Shows disk Zone information (tracks and gaps).

The convention for specifying source and target drives in DOS 3.x is as follows:

0 drives named:

Single drive copy performed, swap disks from source to target until copy is completed.

1 drive named:

Named drive is source drive - if named drive is same as default drive, same as if no drives named.

2 drives named:

First drive is source drive, second is target drive.

3 drives named:

Used only with /O switch - designates overflow drive, which speeds up the copy operation.

In DOS 3.x, two HELP screens describing syntax and switches may be accessed by typing '?' after the command. The DOS 3.x switches for DISKCOPY are as described below:

- /F Invokes a required formatting of the target disk before copying.
- /K 'Kills' the prompts if operation uses two drives - sets errorlevel code on exiting.

/R Bypasses the 'Copy another? (Y/N)' message if no errors occurred - otherwise gives message.

/S Gives name, version number and date created for DISKCOPY command.

/V 'Verbose' switch, meaning that each step in the copying is described on the screen. Shows which sector is being copied, or which track is being formatted.

/O If you have a hard disk and two floppies, the third drive specified should be the hard disk (after the /O switch). This uses the hard disk for holding the source data that won't fit into RAM (the 'overflow'), and thus needs only one read cycle for the whole copy process - no disk shuffling! However, you must use the same floppy disk for the source and target disks for this to work. You may also write more than one copy of the source disk, if you want.

Of course, it is a good idea to see that the hard disk has enough room for the /O operation, since there is no checking done by this command.

DISKCOPY cannot change the format of the target disk from the format of the source disk. That is, it will not make a double-sided copy of a single-sided source disk. If that is what you need, first format the target disk as you wish, then use COPY to transfer copies of the files to the target disk on a file-by-file basis. An errorlevel code is returned at the end of the copying process. The codes are contained in the DOS 3.1 manual under DISKCOPY. A code of 0 means no errors occurred. Other possible codes range from 1 through 11.

## EXE2BIN

EXE2BIN is a command which automatically converts .EXE files to .COM files (.COM is the same as .BINary; .EXE files are generated using a linker working on .ASM (assembly) files. It's useful because .COM files take up less space on disk or in RAM, and because the .COM files load faster. There is a limit on which .EXE files will work with this command, since they must be smaller than 64K to convert successfully. This command is generally of little use unless you are doing your own programming, or want to dig into modifying existing programs.

## FILCOM

This command occurs only in DOS 2.x. Its purpose is to compare two files at the byte level or at the line level. The files may be in binary format or they may be source statements of a programming language. Remember that .COM files are in binary format. The byte comparison displays the bytes that are different, while the line comparison displays blocks of lines which are different.

FILCOM accepts path names in either or both of the named files to be compared. There are four option switches associated with FILCOM. They are:

- /# Requires that *n* lines (from 1 to *n*) match after an error has been found and before the text or data can be considered to match again. Default is 3.
- /B Does a byte comparison between the two files. No attempt to 'resynchronize' after an error. Prints the mismatches to screen.
- /C Will accept either case of letter - upper or lower.
- /W Counts tabs and spaces as one space for purposes of comparison. If spacing is different in the two files, but no space (of at least one space width) is missing, then a match is found.

Screen output can be redirected to a file, if so desired. It would be performed as follows:

EXAMPLE 1: To redirect output to a file, type:

```
FILCOM FILE1.DOC FILE2.DOC > FILE3.DOC
and press 'Return'.
```

For files larger than available RAM, FILCOM performs a complete byte-wise compare, if /B switch is used. Output report is the same as for a file which will fit into RAM. However, if a line compare is performed on a file too large for available RAM, and errors are found in the portion in RAM, FILCOM merely reports that 'Files are different'.

## FGREP

This command will find a specific text string in any list of files. It will output the file name and the line(s) containing the specific string. There are a number of switches intended to speed up the search process. Two of them, marked by **\*\*\***, are in DOS 3.x only. The switches are:

- /^ (That's a carat after the slash) Finds matches for text only if it is at the beginning of a line.
- /\$ Finds matches for text only if it is at the end of a line.
- \*\*\* /B Buffers the output (delivers the output to a portion of memory set aside for this purpose, called a 'buffer'). The default, if the output goes to disk.
- /C Counts the lines with matching text and prints the result to screen.
- /I Ignores the case of the letters in locating a match.
- /L Prints line numbers for the matching text found. It will be easier to find on a printout.
- /N Restricts output to the names of files in which a match was found.
- /R A recursive search (do it again from the top, with the changes just made) extending to the subdirectory level (and down). Normal search is

just the file or files in directories named with the command, but not into the files in subdirectories or sub-subdirectories.

- \*\*\* /U Unbuffered output (see /B, above). The default if output is sent to the screen.
- /V Inverts the match - print all lines with NO match.
- /W Matches text only for condition that text consists of one or two words with no underscore, letter or digit immediately preceding it.

An example of the command format follows:

EXAMPLE 1: To find the string "Peter" in the files C:\DBASE\NAMELIST and A:\NAPLIST, type the following:

```
FGREP Peter C:\DBASE\NAMELIST A:\NAPLIST
```

Notice that the name we are searching for is not in quotes. FGREP requires that we use no quotes unless there are DOS command line characters, hyphens or spaces in the string. Then we must enclose the string with double quotes, as in the example below:

EXAMPLE 2: To find a string with any of the 'exception' characters listed above, the command would look like this:

```
FGREP "A hole-in-one" C:\HELLO.TXT
C:\WFR.LST
```

In this case there are both hyphens, and a space in the string to be found, requiring that the string be enclosed with double quotes.

EXAMPLE 3: If you wanted FGREP to ignore upper/lower case during the search, and to print only the file names of files containing the search string, it would look like this:

```
FGREP /I /N C:\DBASE\YOURFILE
```

## FIND

The FIND command has a function similar to FGREP - to find a specified alphanumeric string within a file. FIND can only search one file at a time. In addition, it has fewer switches to work with, and its search is slower than FGREP. The switches available are:

- /C Only the number of matches found is output to the screen, not the lines containing the match.
- /N Displays the file line number in which the match was found, then the line containing the match.
- /V Sends all the lines NOT containing a match to the screen.



In FIND, you must enclose the string in quotes. If there are already quotes, then enclose the string with its quotes in a set of double quotes to define the string beginning and ending.

## FORMAT

This command is used often and therefore deserves attention in order to understand all that it does. Its basic purpose, of course, is to format disks so that they are usable for storing files. Formatting is like building streets. It creates a patterned area which is specifically set aside for an activity - streets for driving on, and formatting disks for storing files or data on. Without formatting, the computer doesn't know where to put the data, how long it should be in each sector, how or where to write a directory and other housekeeping information it needs in order to transfer the data to disk.

FORMAT works on floppy disks and on hard disks that have been prepared to accept formatting. FORMAT will work without specifying a drive. If no drive is specified, the disk on the default drive will be formatted. Imagine what that can do to a hard disk if what you really wanted was to format a disk on drive B:\ - disaster!

To avoid such a mistake, it is always a wise precaution to specify the drive to be formatted. True, there can be a warning message printed on the screen, but once we are used to the message, how often do we continue without reading the message again? A slight distraction, and you have a newly formatted hard disk; and the data stored there is gone. There are recovery programs to get us out of a fix like that, and they are better than losing the data entirely; but they take a lot of time, frustration and effort to use - and they do not guarantee to recover all the data, depending on what was done to the drive after the problem occurred. Try to do nothing to the disk until you have found out what is most likely to solve the problem.

FORMAT, for each of the DOS versions we are discussing, may be used on V9's using one of the other versions of DOS without a problem, but FORMATTING done on an MS-DOS machine of another type (IBM PC for instance) must be redone on a V9 before files or data can be stored on it.

When FORMAT is complete, DOS reports on how many bytes the disk should hold, and how many it actually will hold after formatting. If there is a difference in the numbers reported, the indication is that there are defective areas on that particular disk. These areas were not formatted, and information is stored on the disk to tell the drive not to write to those areas. This naturally will cut down on the storage space, but the rest of the disk is

usable. There is one exception to a disk being usable when defective areas are found during formatting. If the defect found was on Track 0 (the first track), the FORMAT program rejects the disk as unusable. This is because of the way the housekeeping data has been assigned to specific locations, one of them being Track 0, and damage to the disk in that area means that it is not available to store critical information needed by the computer to store and process data on the disk.

FORMAT has five switch options for use with the command. The options vary with the DOS level being used. \* means DOS 1.x, \*\* means DOS 2.x, unmarked means DOS 3.x. The switch options are listed below:

- /I Creates single sided disk format. This only works on a double sided drive.
- /C 'Clears' the disk of all stored data and files. It does not reformat the disk, but the old format is still valid. Trying it on an unformatted disk results in an error message, since there is nothing to erase.
- \* \*\*/C On DOS 2.x only, the /C switch displays the total number of soft errors. See also DISKCOPY.
- \* \*\*/D On DOS 2.x only, formats a double-sided disk. Use only with double-sided drives.
- \* \*\*/E On DOS 2.x only, the /E switch gives location of soft errors as well as total soft errors.
- \*\* /Fx Sets maximum number of soft errors acceptable (where x = 0-9).
- \* /I On DOS 1.x only, the /I switch clears the Directory and File Allocation Tables only - it does not perform the format routine. Therefore, it may only be done on an already formatted disk. Same as the /C switch on DOS 3.x.
- /M Displays track and head information during format.
- /S After FORMATTING is complete, installs the SYSTEM files at the beginning of the disk.
- /V Volume label or name (up to 11 characters) may be assigned to the disk at the time of formatting.
- \* \*\*/Z On DOS 2.x only, displays Zone information (tracks and gaps)

FORMAT does not work, in DOS 3.x, with ASSIGN, JOIN or SUBST and should not be used.

## GRAFTABL

GRAFTABL loads a table of graphics information. The loading of this table allows display of ASCII graphics characters 128 through 255. This is a terminate-and-stay-resident command, so it is available after loading and until the computer is turned off or rebooted. To load it, just type in the command, and press 'Enter'.

## GRAPHICS

This command allows printing of the display screen on printers which can accept these codes. To enter the command, merely type GRAPHICS, and press 'Enter' after booting with DOS.

## JOIN

This command concatenates or JOINS the directories on two different drives, effectively forming one enlarged directory out of the two. One drive is the 'controlling' drive, while the other drive is subsidiary to the first one. When invoking the command, the first drive specified is the 'control'. All of its directory is involved from the root directory on down. The second drive is the subsidiary drive, and it is specified with a path, which must lead to a directory which is one level down from its root directory. In addition, the directory on the subsidiary drive must be empty when the JOIN command is invoked, or it will not work. The feature that is different about JOIN is that the directories it joins together are displayed on the screen. Not so, on the PATH command. Both search across drive and directory barriers, but only JOIN displays the combined directory.

The format for JOIN is as follows:

EXAMPLE 1: to create an expanded directory covering both drives, A:\ and B:\, use JOIN as follows:

```
JOIN A:\ B:\SOURCE
```

If B:\SOURCE contains files when the JOIN command is invoked, a message is displayed saying "Directory not empty". Further, the path specified cannot be the current directory, or an "Invalid parameter" message will appear.

One switch is valid for JOIN: /D!. This switch invalidates the JOIN, and returns the drives and directories to their normal state.

If JOIN is typed without drives designated, DOS shows the drives and directories currently JOINed. There is no way to refer directly to the joined drive on the command line.

You must refer to the directory to which the drive is joined. Finally, JOIN gets very confused if you try to use it with a drive involved in a SUBST or ASSIGN command.

It also will not work properly if the following commands are attempted while JOIN is in effect:

```
BACKUP      FORMAT      DISKCOPY
DISKCOMP    RESTORE
```

## KEYB(FR,GR,SV,UK)

This command loads a foreign language keyboard - one of four. The default American keyboard is replaced by the foreign keyboard you choose. Since the alternate keyboard is being loaded from disk to RAM, rather than being stored in ROM as the default keyboard is, approximately 2K is added to the space DOS takes up in RAM. The four alternate keyboards are:

```
KEYBFR - French      KEYBSV - Swedish (Sverige)
KEYBGR - German     KEYBUK - British
```

Once the foreign keyboard is loaded, you may shift back and forth between it and the American keyboard. Use these commands:

```
CTRL-ALT-F1 - Switches from Foreign to American
CTRL-ALT-F2 - Switches from American to Foreign
```

If more than one alternate keyboard is loaded into RAM, DOS will access the last one loaded, but not an earlier one.

## LABEL

The LABEL command allows one to create, delete or modify a Volume Label on a floppy disk or on a hard disk. You merely type the disk letter, a colon, then the volume label you choose. You may use up to 11 characters for the label name. By typing LABEL alone, you may delete the current volume label. Answer 'Y' to the question "Delete current volume label? (Y/N).

## LS

The LS command displays a sorted directory listing for the disk or directory specified. LS will show hidden files in its display. There are seven switch options for use with LS, as follows:

- /B Inverts the sort, so the results are listed from Z-A, or 9-0.
- /E Groups the sort by file extension.
- /G Sort not by directory, but Grouped as a whole.
- /L Long format listing - lists file name, attribute, date, time and size; but does not sort.
- /R Lists each subdirectory together with its files. Recursive listing.
- /T Sorts by date and time, but not alphabetically.
- /I Lists one file per line.

That ends this installment. Next time we will finish the external commands.

\* \* \*

## HOW TO RECEIVE NEWSDISKS DIRECT FROM THE VICTOR CIRCLE DISTRIBUTORS [From the recently-received Victor Circle Newsdisk 29]

Some Victor Circle members from the western states who've read the last two issues of VictorTalk may have the impression that we've got big problems with our distributor structure — especially when they learn that the address for the individual who distributes to the far west is changing for the third time in almost that few a number of NewsDisks.

But nothing could be further from the truth. There's nothing wrong with our distributor network; those hard-working individuals are out there expanding CIRCLE.LST with the names of new Victor 9000 owners and NewsDisk readers at a terrific clip. The distributors are doing a fine job of filling requests for old NewsDisks, and they stand ready to get new NewsDisks out the door (on the rare occasion when one finally appears) within hours of initial receipt. If the old Victor Technologies, Inc. had distributors and dealers as dedicated, capable and responsive as Victor Circle distributors, they would STILL be selling as many V9000's as Compaq now does of the newer model Desk Pros. Unfortunately, although the distributors are great at generating new readers, they aren't quite so good at attracting volunteer editors and contributors to CREATE the NewsDisks. Despite what you may have read, our distributors remain committed and capable of moving the product — now all we have to do is give them product to move. In hopes that potential new Victor Circle members will know where to get OLD NewsDisks and that old members may be enjoined to help make some NEW ones, we publish these instructions to one-and-all on: "How to Receive NewsDisks Direct from the Victor Circle Distributors" (seems to me I've read that line before).

NewsDisks are free to members of the VICTOR CIRCLE. The requirement for membership in the CIRCLE is amazingly simple — send a card or letter to:

Dale Copps  
741 West End Avenue, Apt 5F, New York, NY 10025

and declare that you wish to be a member of VICTOR CIRCLE. Include your name, address, phone number (work and home number, if appropriate), and occupation. There are no membership or subscription fees required for CIRCLE membership.

VICTOR CIRCLE NewsDisk distributors have agreed to provide copies of future NewsDisks free of cost to Victor 9000 owners/users in their distribution area who provide formatted (SS) diskettes and pre-stamped diskette mailers, and who officially declare themselves members of the VICTOR CIRCLE.

Procedures for obtaining past NewsDisks vary; but you can't go wrong by sending your distributor sufficient floppy disks (formatted SS) and a stamped, self-addressed mailer large enough to hold all of the NewsDisks you are requesting.

Some distributors will provide floppies, mailers, and postage for past and/or future NewsDisks at some flat rate to cover expenses incurred.

This charge should not exceed \$250 per disk unless exceptional services are requested or additional postage fees are involved in forwarding the NewsDisks. Contact the appropriate distributor listed below for further information on alternative distribution procedures available for your specific area.

### NewsDisk Distributor List

CHRIS DeSALVO,  
19 Ridge Road, Boulder, CO 80303  
Colorado, Kansas, Montana, Nebraska, North Dakota,  
Oklahoma, South Dakota.

BILL DODGE,  
62 Beaudry St., Marlboro, MA 01752  
Connecticut, Maine, Massachusetts, New Hampshire,  
Rhode Island, Vermont.

ISAAC FELDHEIM,  
10 Waverly Place, Monsey, NY 10952  
New Jersey, New York.

JAMES HUGHES,  
P.O. Box 1146, Smithfield, NC 27577-1146  
Alabama, Arkansas, Florida, Georgia, Louisiana,  
Mississippi, North Carolina, Ohio, South Carolina,  
Tennessee, Texas.

TONY POMPA,  
Box 1352, Laurel, MD 20707  
Delaware, District of Columbia, Maryland, Pennsylvania,  
Virginia, West Virginia.

ANN REID,  
P.O. Box 3244, Scottsdale, AZ 85257  
Arizona, New Mexico, Utah, Wyoming.

BOB SWENDSEN,  
152 Aberdeen Drive, Dundee, IL 60118  
Illinois, Indiana, Iowa, Kentucky, Michigan,  
Minnesota, Missouri, Wisconsin.

JOSE VASCONCELOS, M.D.,  
The Brandon Clinic, P.O. Box 280, Brandon, Manitoba,  
CANADA R7A 5Z2.  
Canada.

MICHAEL WISHNIETSKY,  
98-809 Kaha'iea Place, Aiea, HI 96701  
Alaska, California, Hawaii, Idaho, Nevada, Oregon,  
Washington.

## ENLARGING YOUR ENVIRONMENT STRING SPACE / Franz Hirner

MS-DOS or PC-DOS has an area dedicated to the machine environment. What is the environment, you ask, and what should I know about it?

The DOS environment is used to pass information to DOS which gives you greater control over your computer. An example is the PATH command which most of us have included in our AUTOEXEC.BAT file.

```
REM AUTOEXEC.BAT

ECHO OFF
CLS
SET PATH=C:\;C:\DOS;C:\BIN
```

DOS creates an entry in the environment which defines the directories where either DOS or other programs should look for a program file. If you type a file name at the keyboard, DOS will first look in the current directory for the file you want to run. If it finds it, it will load the program and begin execution. If the program is not found, then DOS will begin a directory-by-directory search, using the path you have specified in your AUTOEXEC file. In the above example, DOS would first search in the C:\ root directory and then proceed to search through C:\DOS and C:\BIN looking for the file you specified. Only after failing to find the requested file in the current directory and all the directories specified in the path will DOS report an error.

A very important fact is that the existing environment from program is passed to the next program to execute and, on occasion, it would be nice to be able to change the environment for the new program.

You can check your environment at any time by entering the SET command without any parameters from the keyboard. SET <CR> will display the current environment. Also, the current environment can be redirected to a file by using the DOS redirection capability with the SET command - SET > filename.ext.

"This is very interesting", you say to yourself, but, "I can read about this in my DOS manual". Correct. But DOS doesn't give you many examples and certainly doesn't tell you about the limitations of the DOS environment or provide an answer on how to get around them. This, I will try to do.

Let's take a real life example for purposes of discussion. I am doing a lot of "C" programming and some assembly

programming using Microsoft "C" 5.0 and MASM 5.0, along with the excellent Codeview debugger which is a part of these packages. BRIEF is my favorite editor for creating programs. I wanted to integrate all three packages so that I could (1) create code in BRIEF, (2) compile the program with either MSC or MASM, and then (3) use Codeview to check the actual execution to be sure that the program is functioning the way I thought it would. All simple enough, except that I don't want to exit from BRIEF, compile, load Codeview to find any errors, and then reload BRIEF to correct any errors and start the process over again. I want to keep BRIEF resident in memory and have the capability to both compile and run Codeview. This saves a lot of time during the development process.

Both MSC and MASM suggest that individual directories are created for the .EXE files, LIB files, etc. In order to set the DOS environment so that the BRIEF, MSC and Codeview can find all the files they need I end up with a AUTOEXEC file which looks like:

```
PATH A:\;A:\DOS;A:\BIN;A:\BRIEF;A:\C5\BIN;
      A:\MASM
SET BPATH=A:\BRIEF\MACROS
SET BHELP=A:\BRIEF\HELP
SET BFILE=A:\BRIEF\STATE.RST
SET BBACKUP=A:\BRIEF\BACKUP
SET BCC="cl /Zi /Zp /W3 %s.c /"
SET BPACKAGES=c,h;s;asm,inc,m;r;doc,txt,p;wp
SET BFLAGS=M32klrzmJN
SET BHELP=A:\BRIEF\HELP\MSC5.HLP
SET INCLUDE=A:\C5\INCLUDE
SET LIB=A:\C5\LIB
SET TMP=A:\C5\TMP
PROMPT $P $G
DATE
TIME
echo off
cls
type menu
```

The file could be shortened by using shorter directory names, but I prefer the longer more descriptive names. So far, still no problem, but I have not entered the SETs to cover the various files for MASM yet and my environment is full. The DOS manual neglects to tell you that DOS 3.1 only allows 400 bytes total for the environment. To set up for MASM will take at least another 40 to 50 bytes and DOS will report an "environment error". There are programs supplied with MSC and MASM that will change the DOS

environment size for IBM DOS 2.x or higher; unfortunately, they will not work on the Victor version of DOS. Victor's DOS 3.1 is limited to 400 bytes total. There are occasions where you may want twice that amount of space, or more, in order to configure your system. There had to be a way and John Navas, once again, came to the rescue with his program PUTENV.

Following is the documentation and the source code for John's program.

PUTENV will allow you to load your environment from a data file and pass it on to the next program, in spite of Victor's limited environment table size.

PUTENV is designed for versions of MS-DOS that do not permit the amount of environment string space to be easily increased. It is an entirely well-behaved way to overcome that limitation.

PUTENV adds strings to the environment from a file, and then spawns COMMAND.COM. When COMMAND.COM is exited, the changed environment is released.

The string filename must be the first argument to PUTENV.

The file must contain ASCII strings in the classic STRING=VAL format, with each string in a separate line. This ARChive contains a sample file PUTENV.DAT which can be tested with:

```
PUTENV PUTENV.DAT
```

Type SET to display the changed environment. To release the environment, type EXIT.

If there are any additional arguments to PUTENV, they are passed to COMMAND.COM. For example:

```
PUTENV PUTENV.DAT /C SET
```

will display the altered environment and then immediately release it.

The MASM "C" Code and MAKE file follow. You can load and compile your own copy of PUTENV if you have MSC 5.0 or you can download PUTENV.ARC from BAVUG (415) 574-7914. The ARC file contains the "C" source, the PUTENV.EXE program file, and the MAKE file for compiling.

```

/*
**  ADD A FILE OF VARIABLES TO THE ENVIRONMENT AND SPAWN COMMAND.COM
**  Written in Microsoft C v5.0 by John Navas.
*/

#include <fcntl.h>
#include <io.h>
#include <process.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>

#define STDOUT 1
#define STDERR 2
#define WR_STR(f,s) write(f,s,strlen(s)) /* write any string */
#define SO_STR(s) WR_STR(STDOUT,s) /* any string to STDOUT */
#define SE_STR(s) WR_STR(STDERR,s) /* any string to STDERR */
#define WR_STRC(f,s) write(f,s,sizeof(s)-1) /* string constant */
#define SO_STRC(s) WR_STRC(STDOUT,s) /* string con to STDOUT */
#define SE_STRC(s) WR_STRC(STDERR,s) /* string con to STDERR */

void err_tag(char *tag, char *msg)
{
    SE_STR(tag);
    SE_STRC(": ");
    SE_STR(msg);
    SE_STRC("\n");
    exit(1);
}

```

```

int cdecl main(int ac, char **av)
{
    extern unsigned cdecl _amblksiz;
    int fil;
    long len;
    char *buf;
    register char *p, *q;
    int siz;
    static char prompt[128] = "PROMPT=('exit' to release environment)$ _$p$g";

    if (ac<2) {
        SE_STRC("usage: envarsfile [commandarg ...]\n");
        return 1;
    }
    _amblksiz = 512; /* allocate far heap memory in small chunks */
    if ((fil=open(av[1],O_RDONLY|O_TEXT))==-1 ||
        (len=filelength(fil))==-1L)
        err_tag(av[1], sys_errlist[errno]);
    if (len>32767 || !(buf=malloc((size_t)len+1)))
        err_tag(av[1], "Insufficient memory");
    if ((siz=read(fil, buf, (unsigned int)len))==-1)
        err_tag(av[1], sys_errlist[errno]);
    close(fil);
    buf[siz] = '\0';
    for (p=q=buf; (p=strchr(q, '\n'))!=NULL; q=p+1) {
        *p = '\0';
        if (trchr(q, '=')
            err_tag(q, "'=' is missing");
        if (putenv(q)==-1)
            err_tag(q, "Unable to add to environment");
        SO_STR(q);
        SO_STRC("\n");
    }
    if ((p=getenv("PROMPT"))!=NULL)
        strncpy(prompt+40, p, sizeof(prompt)-40-1);
    putenv(prompt);
    if (!(av[1]=getenv("COMSPEC")))
        err_tag("COMSPEC", "not in environment");
    if ((siz=spawnv(P_WAIT, av[1], av+1))==-1)
        err_tag(av[1], sys_errlist[errno]);
    SE_STRC("\nputenv: environment released\n");
    return siz;
}

```

#### MAKE FILE

```

putenv.exe: putenv.c putenv.mak
cl /Oals /Gsc /F 300 putenv.c /link /e /cp:1
del putenv.obj

```

Create your own environment DAT file:

```

AAA=BBB
CCC=DDD

```

\* \* \*

# FROM THE BOARDS

[FROM THE BOARDS is assembled from messages appearing on various (generally Victor-specific) bulletin boards, mainly The PUB and BAVUG, by John and Mary Jo Knobelsdorf. Only minor editing is done to correct obvious typographical errors and to eliminate some superfluous material. An effort has been made toward a degree of standardized formatting and the elimination of excess space to the extent practical without degrading legibility. Garbles that cannot be readily rectified are allowed to remain. Believing that considerable worthwhile material is contained in these excerpts, our intention is to include this feature in each issue of VictorTalk, as space will allow. - Ed.]

[1] #977 (of 979) MARTY, on 03-JAN-88 22:55 (Size: 1338 Bytes)  
Subject: In Praise of MW, etc.

## DEAREST CHUCK

Chuck Peddle was in the news recently. I've lost the clipping, but the point of the story was that Chuck appeared before some folks from a Fortune 100 company to demonstrate the new Tandon 30 MB removable HD pack, threw the thing on the floor, plugged it in, and watched in amazement as it failed to work.

[1] #979 (of 979) MICHAEL WISHNIETSKY, on 07-JAN-88 22:16 (Size: 496 Bytes)  
Subject: The Able Ms. Abell

PUB-crawlers will be happy to hear that the best friend the user groups ever had has secured a position with BINDCO - a company which provides packaging of various sorts to the microcomputer industry.

Rosemary Abell's new office phone number is (408) 973-9900. Old friends might wish to give her a call and wish her the best in her new job. If you'd like to drop her a line now and again, Rosemary's address is:

Rosemary Abell  
200 Condor Ave.  
Ben Lomond, CA 95005

[7] #85 (of 86) CHARLES VAUGHN, on 04-JAN-88 15:52 (Size: 927 Bytes)  
Subject: VT-100 and Crosstalk

## VT-100

Two years ago I set up my daughter to use CROSSTALK to access her college mainframe. The college DEC liked VT-100. I'm a bit hazy on the details, but I remember that her disk had a VT100.KB file on it. I suspect I must have done a MODCON before loading CROSSTALK and then selected the VT-100 emulation once in CROSSTALK. I think. I think.

At any rate, I find it hard to believe that CROSSTALK has a V9000 VT-100 translation table built into it that automatically takes over when you chose that emulation mode.

If needs be I can grab my daughter's disks and scout around on them to find what dinosaurs of a past era remain. She doesn't throw much away - deletewise that is.

PS: I never really tried MODCONing VT100.KB and NOT selecting the VT100 mode in XTALK, so I don't know what really happens in that case.

[7] #86 (of 86) TONY POMPA, on 06-JAN-88 15:39 (Size: 920 Bytes)  
Subject: Re: VT-100 and Crosstalk

As I remember it, the VT-100 keyboard that is used in XTALK on the V9000 doesn't really have anything in it that is incompatible with the other terminal emulations. It only reassigned a few keys, for the purpose of permitting some full-screen editing and the use of the numeric keypad as a clone of the VT-100 keypad. I \*\*DO\*\* recall that the VT-100 emulation would not work properly (for full-screen editing) if you did not MODCON the VT-100 keyboard prior to invoking XTALK.

One of the best VT-100 emulators that I ever used on the V9000 was the one from Zenithplan (a British company) that was available in 1983-1984. It did some really interesting things like allowing a real 132 column mode that actually worked. If anybody's interested in it, I have a legal copy, with manual available for \$12.00. Please note: I think the only file download capability that is supported is pure ASCII.

[8] #230 (of 231) JOHN DANKOWYCH, on 04-JAN-88 16:21 (Size: 9923 Bytes)  
Subject: Quickbasic, IBMBIOS

## THE GOOD NEWS:

Enter Ralf Kurschat! At the end of November I downloaded Kurschat's latest and greatest IBMBIOS emulator which now includes a Color Graphics Adapter emulator. After testing it out with a variety of IBM software including Sidekick, I decided to risk \$100 on MS Quickbasic 4.0.

## THE RESULTS:

MS Quickbasic 4.0 works under the IBMBIOS emulator (V 3.12). There are some features that I have not been able to test yet:

1.) The keyboard - The key assignments in IBMBIOS are not totally compatible with QB. I will have to move things around. The QB option that is most affected by this is the EDIT menu - cutting and pasting does not work as a result. IBMBIOS does include an assembler listing of the keyboard assignments and I should be able to make the changes needed.

2.) The mouse - I don't have one yet. I have a WAL V2 board which includes a mouse port and device driver. I don't know if this driver will work properly with QB. The mouse driver included with QB might work as well if the serial ports are properly emulated. (I seem to recall that they are (were) not.)

3.) The compile and link option does not seem to work if your source files are in a subdirectory or in a directory other than where QB resides. I'm not sure if this is an IBMBIOS or QB problem. I have been able to compile/link from within QB when everything is on one disk. Otherwise I have been compiling/linking from DOS.

## WHAT QUICKBASIC PROVIDES

1.) An interpreter (sort of). Each line of code is actually compiled as you enter it. I have not done any rigorous tests, but so far code seems to run as fast in interpreter mode as in compiled mode.

2.) Automatic syntax checking when you press <CR>. Online Help.

3.) Subroutine support. Modularity - structure.

4.) Inter-language support. C, asm etc.

5.) 8087/80287 used for floating point calculations (if present).

6.) Full screen editing. Multiple editing/viewing windows.

7.) QB seems to be using MS Windows. It does not include any windows software for the user. These presumably come with MS Windows or the Windows development kit.

8.) Extensive debugging - break point setting. Single line stepping. Fast subroutine stepping. Output screen view. Execution continuation after error and correction.

9.) Separate batch compile and link. These are actually used from within QB as well.

10.) Support for most IBM compatible graphics boards: CGA, HGC, EGA, VGA and MCGA. I tested the IBMBIOS CGA emulation with the TORUS program. It works great.

11.) Extensions to the language:

Binary file support

User data types

Recursive subroutines and functions

32 bit integers, 8087/80287 support

Control flow statements: IF...THEN...ELSE, DO

UNTIL, etc.

12.) Code and buffer size no longer limited to 64kbytes.

Quickbasic has many of the "structured language" features touted by fans of C and Pascal.

Will any programs written, compiled & linked in QB run on the V9000 in V9 mode (ie not under the emulator)? I don't have a definite answer but I think the answer is yes if you don't do any CGA graphics calls (or anything else IBM specific). I think that one should be able to call V9000 specific graphics subroutines, eg: Bik Chung Yung's or possibly the newly released Grafix.lib.

## IBMBIOS 3.12

Great stuff! Neat! Wow! Can EGA emulation be far away?

It is NOT TRUE that IBMBIOS will not work with the WAL V2 board. IBMBIOS needs 896k to be present in your Victor 9000. Many V9000 users have the WAL V2 ram/SCSI board. It definitely can be used to run IBMBIOS. What is true is that the WAL RAMDRIVE will not work when IBMBIOS is invoked. You should remove the WAL ramdrive from your config.sys file if you plan to run IBMBIOS. The integral WAL SCSI hard disk controller/driver also works under the IBMBIOS.

## MSDOS 3.xx

NOTE! All of the above tests of IBMBIOS and Quickbasic were performed under MSDOS 3.xx. I happened to have it



and reasoned that the later DOS would be most compatible with IBMBIOS and QB. Both will likely work as well under DOS 2. Note also that some of the WAL device drivers need to be updated to DOS 3 versions (available from WAL).

#### A GOOD REASON TO UPGRADE TO MSDOS 3.xx

A recent review of MSDOS 3.xx led me to regret the purchase of a hard disk optimiser. Apparently Microsoft has performed major surgery to the file sector allocation algorithm in DOS 3. The result is much less file fragmentation. This means that if you rebuild your hard disk under DOS 3 from scratch that you won't get much more performance by running a disk optimiser.

#### Brooklyn Bridge and MSDOS 3.xx 16 bit FAT entries.

I recently bridged an IBM AT to a V9000. The AT was running DOS 3 and the V9000 DOS 2. Copying files under AT control from the V9000 floppies worked fine. Copying files under V9000 control from the AT (AT hard disk formatted under DOS 3) seemed OK until the copied files were examined. They were garbage. Checking the BB manual I discovered a warning to this effect. I repeated the experiment with DOS 3 on the V9000 and everything is now OK going both ways. This suggests that, contrary to what I had read on the PUB or on a Newsdisk, the Victor MSDOS 3 supports (at least the reading of) 16 bit FAT entries. I don't know if the Victor MSDOS 3 (hard/floppy) disk formatting routines write/allow 16 bit FAT entries. Recall that MSDOS 2.xx uses 12 bit FAT entries.

#### WRITING MSDOS DEVICE DRIVERS by Robert S. Lai / The Waite Group

This is an excellent book. It includes everything you need to know to write device drivers. The author offers a diskette with all of the examples listed in the book: ramdrive, clock, console and printer drivers.

Has anyone heard of Robert S. Lai. He is obviously a V9000 user since he mentions it in several places in the book. I plan to write to him for the disk and ask for V9000 versions of the drivers. It seems to me that Robert would be the ideal person to work on the now dormant MS Windows conversion project.

#### WAL RAMDRIVE under IBMBIOS

I miss my 1 Megabyte ramdrive while under IBMBIOS. I am in the process of reverse engineering the WAL ramdrive device driver. I think I understand how most of it works. The ramdrive uses the top 64k segment D0000H as a window

into the ramdrive memory. This window needs to be moved to some other segment since IBMBIOS need 896k to run. I propose to use the segment at A0000H while installing an 896k main memory system. So far this does not work. The system hangs during ramdrive installation. WAL manual states that 832k is the maximum main memory allowed with the ramdrive. This I assume is because the ramdrive window is at 832k. Once I patch the window location to A0000H (640k) the 832k restriction should be gone. If I ever get this going I'll upload the required patch to the PUB. I did call WAL 3-4 weeks ago but they failed to respond.

#### IBMBIOS - IO2NMI

My colleague, David Lewis, informs me that it is possible to implement the IO2NMI hardware via jumpers on at least one version of the V9000 motherboard. This is not the best way to do it since a different set of instructions would be required for each version of the motherboard. David also has an improved version of the circuit to suggest.

[9] #68 (of 70) TONY POMPA, on 02-JAN-88 09:50 (Size: 299 Bytes)  
Subject: Re: Plus-PC

Bruce may be correct in stating that PLUS-PC can't boot PC DOS above 3.2. I know for certain that the VI's will boot DOS at least as high as 3.1 but I can't categorically state that I've ever done it on a PLUS PC. Maybe Marty knows, since he acquired a PLUS-PC only a few short months ago.

[9] #70 (of 70) BILL WITOWSKI, on 07-JAN-88 12:20 (Size: 872 Bytes)  
Subject: +PC Serial Ports

After buying the Brooklyn Bridge I was pleased to learn that it operates on my +PC V9 in BOTH modes, which leads to the following observations and questions:

Using both computers (V9 and a Zenith laptop) via Brooklyn Bridge effectively doubles the RAM available for processing. By using a memory splitter-toggler like Program Shift you can effortlessly hop between applications via hotkeys. No, it's not multi-tasking, but it is a LOT faster than a shell exit loading command.com and whatever program you want.

The Victor version of Brooklyn Bridge includes an "Express Lane" which is a hardware attachment to the V9 serial port to make it work at speeds higher than the standard 38,400 maximum (it runs at 115,200). Could this same mod be used to increase port speed past 1200 in IBM +PC mode?

\* \* \*

TEXT REVERSING PROGRAM / R. W. Johnson

[Editor's Note: Andy Rudiak, Fontfreak and VictorTalk Editor Emeritus, has long tickled/bedeviled local members of Victor\*Group with his cryptic monthly meeting notices.]

Given the propensity [Andy Rudiak's] for cute meeting cards, I thought the following reversing text program might prove useful. The text file can be made using Wordstar and STRIP, or Wordstar in a non-document mode, or in other ways. And, of course, printer routines can be added, or simply use the reversed file for the file to be printed.

```
100 'REVERSE IS A PROGRAM TO PRINT A TEXT
110 'BACKWARDS FROM THE END TO THE BEGINNING.
120 CLEAR
130 DIM L$(200)
140 OPEN "I",1,"TEXT"
150 OPEN "O",2,"REVTEXT"
160 J=1
170 IF EOF(1) THEN 210
180 LINE INPUT #1,L$(J)
190 J=J+1
200 GOTO 170
210 J=J-1:CLOSE #1
220 LPRINT CHR$(27)+CHR$(77):'FOR PRINTER SET
225 'TO 12 CPI (CHANGE AS REQD.)
230 FOR J=J TO 1 STEP -1
240 L$=L$(J):GOSUB 400
250 L$(J)=L$
260 PRINT L$(J)
270 PRINT #2,L$(J)
280 LPRINT L$(J)
290 NEXT J
300 CLOSE #2
310 LPRINT CHR$(12)
320 STOP
400 'SCAN EACH LINE AND REVERSE IT*****
410 P$=""
420 L=LEN(L$)
430 FOR K=L TO 1 STEP -1
440 I$=MID$(L$,K,1):P$=P$+I$
450 NEXT K
460 L$=P$
470 RETURN
```

Running the program with the above paragraph as TEXT, gives for REVTEXT:

.detnirp eb ot elif eht rof elif desrever eht esu ylpnis ro ,dedda eb nac senituor retnirp ,esruoc fo ,dnA .syaw rehto ni ro ,edom tnemucod-non a ni ratsdroW ro ,PIRTS dna ratsdroW gnisu edam eb nac elif txet ehT .lufesu evorp thgim margorp txet gnisrever gnivolof eht thguoht I ,sdrac

gniteem etuc rof ]s'kaiduR ydnA[ ytisneporp eht neviG

COMPUTERS FOR SALE

Victor 9000, 256 K RAM,  
double-sided floppy/20Mb hard disk,  
Epson MX-100 printer  
Software includes: MS-DOS 2.11, Lotus 1-2-3,  
Multiplan, Wordstar, Peach Text.  
\$750

John Maciazek  
1838 Sumatra Ave.  
San Jose, CA 95122  
(408) 926-3726

Victor 9000, 384 K RAM,  
single-sided floppies,  
Plus PC installed  
Diablo 620 serial printer  
Software includes: MS-DOS, Lotus 1-2-3 & 1A,  
Benchmark word-processing version 3  
Asking \$1500; other offers considered

Bill Linsman or Katie Colvin  
Linsman Film  
6930 E. 1st St.  
Scottsdale, AZ 85251  
(602) 949-9008

4 Victor 9000, dual floppy disks, 256K RAM  
1 Victor 9000, floppy/10Mb hard disk, 256 K RAM  
1 Televideo personal terminal  
1 Novell 47Mb hard disk  
1 Novell File Server 6 station  
includes Netware 6B & cards

Lotus 123, Supercalc, Word Perfect, dBase II, Wordstar, Open Systems, Multiplan, and various utilities. Many extras including manuals and extra keyboard. All for a low price of \$3000. or OBO.

Contact: Carol, Cascadian Farm, Star Rt., Rockport, WA  
98283 (206) 853-8175

# HELP FORMULATE VICTORTALK IMPROVEMENTS

[Two members of the Victor\*Group Steering Committee have composed this plea for help from you, our readers, in shaping future issues of VictorTalk. Please heed their plea! - Ed.]

This is a letter to ask each of you readers of VictorTalk what subjects you would like to see covered in the year to come. We know that each of you has used your computer for projects that are of interest to you. We know also, that there are times when the stubborn, idiosyncratic thing just will not do what you want it to do - it persists in doing what you tell it to do, and that's not very nice.

We have thought of a number of changes to the content of Victor\*Talk to help you in getting the most out of your computer, be it V9000, Victor PC or what. But we need your help in setting our goals for next year's editions to make it the most interesting and to-your-point ever.

If you want it, there could be a question and answer column in which the editors would scour the ranks to get a clear, complete response to your needs. We can have more tutorial articles on basics or on popular application programs. There can be a tips-'n-tricks corner with short programs to do useful things for you. Some of you may want to start programming or to advance your capabilities to a higher level. Some may like to look at different languages, such as the C language. Want to do more than a birthday list with dBASE III+? How would you like to be able to make your mailing list work FOR you instead of against you? Are you getting all the help you can from your computer and are still doing more than half the job yourself? Rise up! Defend your right to have the computer do these tasks! New, more powerful word processor programs have appeared. The latest will even import high quality artwork, and incorporate it into the body of the text. A good start on Desktop Publishing - more than adequate for many tasks, including makeup for advertising for a small business. Want to know more about this? Tell us.

IBMBIOS is a program to emulate the IBM PC on your V9000. It has been upgraded greatly since it came to us from Germany. It has gotten better, and is getting better yet. It covers text output now with great things coming for handling graphics for an even wider variety of needs. Want a hard disk, but don't want the hassle of trying to find DMA cards and Controllers to make it work with your computer? We are close to being able to tell you how to use a low-cost IBM compatible hard disk with the V9000, which will also allow adding a floppy disk fully compatible with IBM,

without giving up your present Victor drives. This means you can read IBM format disks directly, and can use them directly, under IBMBIOS emulation. If this sort of information interests you, let us know so we can put the right articles together for you.

VictorTalk has been a great newsletter. A lot of effort has gone into it, and the results have shown. It stands head and shoulders above many other newsletters for content and for professionalism in format and finish. But it can be even better with your desires incorporated. Take some time to think about this, then take up the trusty word processor, and bang out a thoughtful answer. Although "Keep up the good work" is really appreciated by the staff, this time it won't cut any ice - and more important - it won't get you the type of articles you wish we would publish! So, a word to the wise: get your comments in early to be sure they will be read and acted on for early publication. Do it while you are still thinking about it. We're waiting!

\* \* \*

---

## PRO-YAM COMMUNICATIONS PROGRAM FOR VICTOR 9000

Franz Hirner reports the following possibility: If sufficient interest is shown, Chuck Forsberg of Omen Technology, Inc. (Portland, OR) has indicated that they would make a Victor-specific conversion of Pro-Yam. Pro-Yam (also known as Pro-Comm) is a full-capability IBM communications program with extensive script file capability. The anticipated price would be in the vicinity of \$130. If interested, drop a note to Franz, or leave him a note on BAVUG BBS. (See page 24 for address.)

\* \* \*

---

## ADA COMPILER

A note from Bob Crissman (16911 Colombine Dr., Pflugerville, TX 78660) states the following: "The new Dept. of Defense language, ADA - runs on the Victor 9000 - using the Janus Ada compiler from R&R Software - Madison, Wisc. I have it up and running on my Victors (hard disk)."

\* \* \*

# ELECTRONIC GREMLINS / John Knobelsdorf

On Tuesday, May 31, 1988, I was logged onto the PUB bulletin board in Maryland. I was using VICMINI for a communications program and an AT&T line, as is my standard procedure, and was working from a subdirectory on my internal harddisk.

I was also using Vicmini's [C]apture Text command to capture all of the transmission, and was using ASCII protocol.

I noticed a lot of line noise as I was laboriously typing a brief mail message in [S]end mode to Gratz Roberts (the Sysop), but continued to backspace and re-type as necessary to correct both my typing errors and get rid of the funny looking characters that kept appearing on the screen.

After I finished the message and sent it to Gratz's mailbox, I proceeded to check the Boards and do a [B]atch download. After that, I checked the [F]ile sections, did a [R]ecent listing and then a [G]oodbye.

I then [S]topped collecting text, [Q]uit to DOS, changed the blinking cursor to the usual block cursor, and was ready to review the captured text.

"So what!?", you ask.

So., I found that I had problems similar to those experienced by Caryl Conner on March 29, 1988.

The subdirectory I was working from was in shambles, with a bunch of gobbledy-gook listed as files. The DIR command produced a listing of the strangest filenames I ever saw, made up of ASCII characters from various parts of the captured text, mostly of more than the normal maximum eight character length, and with wierd file creation dates. The SDIR command produced a similar listing, but with fewer files. I was to later find that some of the DIR command listed files were being treated as subdirectories within my working subdirectory.

I promptly checked the rest of my harddisk and found that the problem, to lesser degree, existed in four other of my subdirectories, including that (\BIN) where I store most of the utility files that I use. Continued checking revealed that I could still invoke some of the utilities, such as

LIST550, but not others, such as FORMAT. When trying the files that could not be used, I would see a "sector not found - can't access drive A" error message. It appeared to me that the File Allocation Table(s) were partially corrupted, if that is possible.

Fortunately, I maintain a set of backup disks, one for each of the subdirectories on the harddisk. It was relatively easy, but rather exasperating to delete all of the files in each subdirectory, remove the subdirectory (for added safety), create a new subdirectory of the same name, and copy the files from the backup floppies to the respective "new" subdirectories.

It was relatively easy, that is, except for the communications subdirectory that I had been using for the BBS activity. Before I could delete the "files" in this subdirectory, I had to use CHKDSK with the "fix" switch (/f) to get the gobbledy-gook into one big file, which could then be deleted with the DEL command.

There were times when I thought "Oh rats!, why not just do an HDFORMAT and an HDSETUP, and quit messing with this?", but then I would recall how good it would feel when I finished the job and "quit beating my head against the wall".

I attribute all of the problems to line noise during transmission, but I would be the last person to proclaim that to be the case. I think that the problem extended to the subdirectory \BIN because that is where I stored VICMINI (but no longer do so), and I think that the problem with the other affected subdirectories resulted from problems that were created in the file allocation table(s).

If misery loves company, Caryl Conner will be relieved to know that she is not the only person pursued by the electronic gremlins. If there is a lesson to be learned from all of this, it is to actively and continually maintain current backup copies of all your files, whether on harddisk or on floppies. If anyone else has experienced similar difficulties, let us hear from you. If anyone else has a good, or better (or best) explanation of the cause of the problem, please share that information with us.

\* \* \*

# THE LIBRARY REPORT / John Knobelsdorf

This edition of the Library Report will announce several major changes that have been made in the format of the V\*G public domain disks.

The number of public domain disks is approaching three digits. To allow for this, each disk will now be referred to as Disknnn, where the nnn will be the number of the disk (such as Disk098, Disk099, Disk100, Disk101, etc.).

The DISKID, or descriptive information file for each new disk will have a similar connotation. The filename will be DISKnnn.TXT, where again the nnn is the number of the disk (such as DISK098.TXT, DISK101.TXT, etc.)

Beginning with DISK098, files and programs will no longer be released in Squeezed, Librared or Archived form.

Each new disk will be setup using a Root Directory and Sub-directories. The use of Sub-directories is necessary to avoid the problems associated with the unarching and storing on disk of different archive files when each contains an individual file with differing content but duplicate filename (such as README, DISKID, etc.)

The Root Directory will include the name of the descriptive information file for the disk (i.e. DISK098.TXT), and the names of any other individual files that were obtained, not as part of a librared or archived set of files, but as individual files and subsequently "published" on the referenced disk. All individual files will be in UnSqueezed form.

The Sub-directories will each individually contain the contents of a single Librared file or a single Archived file. The name of each Sub-directory will be that of the Librared or Archived file, but without the .LBR or .ARC extension.

An illustration of the new format will be found at the end of this article in the listing of DISK098.TXT from our most recently released public domain disk.

The new public domain disk format with sub-directories will require that users of the disks have and use MS-DOS 2.11 or higher. This requirement also existed for the unarchiving of .ARC files when using PKXARC.COM or ARC.COM.

The Editor of VictorTalk, Jack Prichard, has volunteered to re-create those public domain disks having archived files (PD Disks #078 thru #097, plus a few earlier ones) using the

new format with sub-directories. That will be a time consuming labor of love, and Jack is to be complimented on his interest in your behalf. It will require two or three disks to hold the individual files now contained on one disk of archived files. The revised disks will be issued in sets, and will be identified by the addition of an "A", "B", or "C" to the original disk number as necessary (i.e. DISK89A, DISK89B, DISK89C and DISK90A, DISK90B etc.). Thus, disks 054, 060, 073 and 078 thru 097 will be available either as single disks with the original .ARC format, or as sets of one, two or three disks in the new format. The revised disk sets will be sold at the same price as the original single disks (with no additional charge for the extra one or two disks in the set). Also, the price remains unchanged at \$5.00 per disk number (set or single) to V\*G members and \$10.00 per disk number to non-members. A revised order form which details the new ordering options and explains the disk nomenclature has been included in this issue of VictorTalk.

Speaking of VictorTalk, a minor clarification is in order concerning an editorial comment on page 3 of Vol. 6, No. 3, in the article "IBMBIOS". The public domain does include the file HXTBIOS.COM. That may not have been readily apparent because of the previous practice of publishing archived files in archived form. HXTBIOS.COM is part of the archive XTBIOS.ARC which may be found on DISK087. When DISK087 is reissued, you will be able to find HXTBIOS.COM listed in the sub-directory named XTBIOS. Coincidentally, updated versions of XTBIOS.COM and HXTBIOS.COM are included on DISK098, which has just recently been released; they are listed thereon in the sub-directory named XTBIOS3.

This issue of the Library Report will also announce the release of one more public domain disk.

On July 6, 1988 we released DISK098. This disk is in the Chapter set and is available for copying at the monthly meetings. It also may be ordered thru the mail in the usual manner. Please use the new order form which is included in this issue (Vol.6, No.4) of VictorTalk.

The files on the new disk came from the PUB and BAVUG bulletin boards. Some of the files will require the use of a Victor VI, Plus-PC, a PC-Compatible or one of the various emulators. Be sure to read the DISKnnn.TXT information file on each disk and check the descriptions before you try to use a particular program. Also, remember that some of the files on the diskettes will be listed in various sub-directories.

Following is a list of files on DISK098:

Program Name	Ext.	Size (bytes)	Additional Information
DISK098	.TXT	7680	Information about the files on this disk.
DSZ0406			X/Y/ZMODEM file transfer (4/6/88 version)
2A-PIF	.DVP	416	
2S-PIF	.DVP	416	
2T-PIF	.DVP	416	
DS	.BAT	44	
DSZ	.COM	50200	
DSZ	.DOC	63140	
FIX	.BAT	278	
MAILER		4	
QEDT202A			Latest version of QEDIT - use IBM or Emulator
CHUNKER	.EXE	8880	
HINTS	.DOC	838	
ORDERFRM	.DOC	3370	
PRESS	.DOC	3942	
Q	.EXE	47984	
QCONFIG	.DAT	6759	
QCONFIG	.EXE	30208	
QDISK	.EXE	1834	
QEDIT	.DOC	65079	
QHELP	.TXT	2050	
READ	.ME	3090	
SHOWKEY	.COM	2569	
UPDATE	.DOC	2659	
RSLDR201			RSL Ramdrive v.2.01 for WAL v2, use IBMBIOS
DISKID		376	
RAMDRIVE	.COM	3012	WARNING!! Do NOT use these files!
RAMDRIVE	.DEV	2462	There are bugs in here!
RAMDRIVE	.DOC	10440	Use the RSL Ramdrive v.2.02 on DISK099.
RAMDRIVE	.LOG	245	This information was learned after the
RSL200	.MSG	8429	DISK098 had been released.
TOLLFREE			List of toll-free (800) BBS numbers.
TOLLFREE	.BBS	11514	
VDIFF			Compares two files, marks differing lines; V9 OK
VDIFF	.C	26386	
VDIFF	.EXE	13060	
VDIFF	.TXT	7808	
XTBIOS3			09/30/87 Add'l. Keyboard changes for (H)XTBIOS Emul
HXTBIOS	.COM	8128	
README		2252	
VF	.COM	11648	
VF	.DOC	4736	
XTBIOS	.COM	7760	
XTBIOS	.DOC	23040	
ZDEL50			File delete utility, good options/wildcards
READ	.ME	1346	
ZDEL	.ASM	23856	
ZDEL	.COM	1556	
ZDEL	.DOC	403	
ZDELREAD	.ME	1346	



## CONTRIBUTORS

To facilitate readers contacting authors, we are continuing the policy of providing author's addresses and phone numbers (unless requested otherwise). Here are this issue's contributors:

George Anderson,  
782 Jordan Ave.,  
Los Altos, CA 94022  
(415) 941-3029  
(New address and phone)

Franz Hirner,  
1493 Beach Park Blvd, Box 180,  
Foster City, CA 94404  
(415) 349-3602

R. W. Johnson,  
2820 Grant Street,  
Concord, CA 94520  
(415) 687-7620

John Knobelsdorf,  
Route 2, Box 103,  
Bertram, TX 78605  
(512) 355-2084

\* \* \*

## VictorTalk

### Address inquiries to:

BAY AREA VICTOR USERS GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

**Subscriptions:** Carol Hirner

**Editor:** Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

**VictorTalk** is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in **VictorTalk**. Copyright 1988 by Victor\*Group. Contents may be reproduced by permission only. **Permission is only a post card away.**

**VictorTalk** is an independent publication not affiliated in any way with Victor Technologies, Inc. **Victor** is a registered trademark of Victor Technologies, Inc.

**Submit Articles** on MSDOS disks in ASCII or Spellbinder format (either single-sided or double-sided, in Victor or IBM format), with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

**BAVUG BBS** is on line 24 hours a day: (415) 574-7914, Franz Hirner, Sysop.

## Yes, I want to join.

Enclosed is a check/money order for \$35.00  
(foreign membership, \$40.00 in U.S. dollars)  
for my 1988 Victor\*Group membership.

Name \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

Make checks payable to:

## VictorTalk

1493 Beach Park Boulevard, #180  
Foster City, CA 94404

City \_\_\_\_\_

State \_\_\_\_\_ ZIP \_\_\_\_\_

Country \_\_\_\_\_

Phone \_\_\_\_\_



PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date \_\_\_\_\_ Office use only:  
Name \_\_\_\_\_ Order rec'd \_\_\_\_\_  
Company \_\_\_\_\_ Member check \_\_\_\_\_  
Address \_\_\_\_\_ Comments \_\_\_\_\_  
City \_\_\_\_\_ State & Zip \_\_\_\_\_  
Country \_\_\_\_\_ Phone \_\_\_\_\_ Order shipped \_\_\_\_\_

Circle disk numbers ordered (all are single-sided). Un-archived versions are available where "A", "AB", or "ABC" appears below number. If originally released (archived) version is desired, cross out corresponding "A", "AB", or "ABC". (Original releases of all other disks contain no archived files.) Note that pricing below is per "disk number" rather than per "disk" (re-released unarchived "disk numbers" may be comprised of one disk (A), two disks (AB), or three disks (ABC). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No.	Disks	Vol. No.	Disks	Vol. No.	Disks	Vol. No.	Disks
4	2 61-64	5	1 78-81	5	5 87-90	6	3 96-97
4	3 65-71	5	2 82-84	5	6 91-92	6	4 98
4	6 72-77	5	4 85-86	6	2 93-95		

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54 A	55	56*	57*	--	59	60 AB
61	62	63	64	65	66	67	68	69	70	71	72	73 A	74	75	76	77	78 AB	79 AB	80 AB
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	—	—
AB	AB	AB	AB	AB	AB	AB	ABC	ABC	AB	AB	ABC	ABC	AB	ABC	AB	AB			

\* 34 & 35 are a two-disk set; 56 & 57 are also a two-disk set

1988 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, sets of 6, \$12.00 /set ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1983-1984, set of 12, \$1.00 each ..... \$ \_\_\_\_\_

1988 membership fee \$35 (receive all 1988 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR\*GROUP

1493 BEACH PARK BLVD, #180

FOSTER CITY, CA 94404



## FIRST CLASS MAIL

POSTAGE WILL BE PAID BY ADDRESSEE

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP

1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date \_\_\_\_\_  
Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ & Zip \_\_\_\_\_  
Country \_\_\_\_\_ Phone \_\_\_\_\_

Office use only:  
Order rec'd \_\_\_\_\_  
Member check \_\_\_\_\_  
Comments \_\_\_\_\_  
Order shipped \_\_\_\_\_

Circle disk numbers ordered (all are single-sided). Un-archived versions are available where "A", "AB", or "ABC" appears below number. If originally released (archived) version is desired, cross out corresponding "A", "AB", or "ABC". (Original releases of all other disks contain no archived files.) Note that pricing below is per "disk number" rather than per "disk" (re-released unarchived "disk numbers" may be comprised of one disk (A), two disks (AB), or three disks (ABC). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No.	Disks	Vol. No.	Disks	Vol. No.	Disks	Vol. No.	Disks
4	2 61-64	5	1 78-81	5	5 87-90	6	3 96-97
4	3 65-71	5	2 82-84	5	6 91-92	6	4 98
4	6 72-77	5	4 85-86	6	2 93-95	6	5 99

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56*	57*	--	59	60
													A						AB
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
												A					AB	AB	AB
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	—
AB	AB	AB	AB	AB	AB	AB	ABC	ABC	AB	AB	ABC	ABC	AB	ABC	AB	AB			

\* 34 & 35 are a two-disk set; 56 & 57 are also a two-disk set

1988 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, sets of 6, \$12.00 /set ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1988 membership fee \$35 (receive all 1988 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR<sup>®</sup> TALK

Bi-Monthly Magazine for Victor 9000 users

VictorTalk, Vol. 6, No. 5, Sep-Oct 1988

## NEW EDITORIAL BOARD FOR VICTORTALK / Dan Mardesich

The Victor\*Group Steering Committee has proposed a new structure for the editorship of Victor Talk and is calling for volunteers.

About two years ago, Jack Prichard agreed to edit two issues of Victor Talk, and now having now done twelve issues (by the end of the year), he is understandably ready to move on to other things. He has done a superb job of collecting and editing material, and most importantly, writing a great deal of it himself. We all appreciate his efforts in indexing and cataloging VictorTalk articles and public domain disks and programs as well as the many other contributions.

John Knoblesdorf took over the managing of the Public Domain Library some three years ago and has since retired from his business and moved to a spread in West Texas. He has continued to produce new public domain disks, to maintain the catalog of programs on the various disks, and to distribute the disks by filling orders from all over the world. John, too, wants to move on to other things and is ready to retire as the Public Domain Librarian.

The Steering Committee, on behalf of the Victor community, wishes to express sincere thanks to both Jack and John for all the time and effort they have expended in their dedication to keeping the Victor 9000 a viable option in computing. Without them we would all be poorer.

So we need a new editor and public domain librarian. Rather than lay all the responsibilities of the editorship on one individual, the suggestion has been made to form an Editorial Board to distribute the work load. The breakdown of duties would be something like the following.

### EDITORIAL BOARD

**Managing Editor:** who will be responsible for finding people to write articles and seeing that regular contributors get their copy in on time. He or she will transmit the copy to the..

**Executive Editor:** who will do the actual "editing" of the copy submitted, putting it into readable form, adding comments, checking accuracy, etc. He or she will transmit the final copy for the issue along with a hard copy to the..

**Paste-up Person:** who will print out copy, strip in headings, footers, tweak the text, paste up the pages and take it to the printer. From there the mailing group will pick up the newsletters, stamp, label and mail them.

In addition to the above crew, we are looking for some regularly contributing editors:

**"From the Boards" Editor:** download messages from the bulletin boards and edit them into interesting dialogs concerning current interests. (Help is available for your phone bill.)

**Public Domain Review Editor:** assemble a column for each issue discussing one or more PD programs.

**"Back to BASIC" Editor:** assemble a column for each issue on programming in BASIC or other languages.

**Editor for "Hints & Kinks", or "Q&A", or "Networking", or some such column.**

**Editor for a column about getting started with a V-9000.** This column would be the genesis of a pamphlet for people who buy or inherit orphan computers that have no documentation. It would include hints on getting started with a Victor, where to go for help or maintenance, what to do until the doctor comes, that sort of thing.

### PUBLIC DOMAIN LIBRARIAN

Collect programs from other public domain disks, bulletin boards, etc. and assemble them into new releases for the PD library. Write disk ID for each new disk. (A 1200-baud modem and some help on the phone bill is available.)

Maintain the catalog of disks, programs, and disk IDs.

Fill orders for PD disks: copy disks and mail them out.

Write "The Library Report", for each issue of VictorTalk, describing the latest disk release(s).

— — —  
George Anderson has consented to become the Executive Editor, even though he expects to be moving to the San Diego area shortly. Andrew Rudiak will be our paste-up and tweaker person and the Moore boys will continue to be the mailing crew. The rest of the positions are up for grabs. If you are interested, or know someone who is, please contact:

George Anderson  
782 Jordan Avenue  
Los Altos, CA 94022  
(415) 941-3029

\* \* \*

## THE VICTOR 9000 POWER SUPPLY / Gratz Roberts

[The following article by the PUB SysOp is reproduced, with thanks, from Victor Circle NewsDisk 20. - Ed.]

The following information is based on my experience gained after the disastrous failure of my Victor 9000 power supply and my subsequent experience gained while repairing other Victor 9000 power supplies. The recommended test set up and repair techniques are my own, not those of the Victor Corporation. Likewise, the suggested replacements are parts that I would use in the repair of a defective unit, not necessarily those that would be used by a Victor Repair Facility; however, I am confident that the suggested parts will provide adequate service.

Note -- the following discussion and symbolic references refer only to the Compower Corp. power supply and Victor Power Supply Schematic (Drawing Number 100701). Although the Aztec replacement supply is probably quite similar in design, the schematic and reference numbers are not directly applicable to that supply.

The Victor 9000 power supply is a fairly sophisticated, switching type supply that is capable of operating over a much wider range of input voltages and line frequencies than that of the conventional brute-force supply found in "that other machine"; consequently, the Victor 9000 power supply is both somewhat harder to trouble shoot and slightly less reliable than the conventional supply.

The Victor 9000 supply provides four different outputs:

- +12 volts regulated at approximately 1.1 amp
- 12 volts regulated at approximately 0.2 amp
- +5 volts regulated at approximately 5 amp
- +12 volts unregulated at approximately 1.5 amp

It should be noted that these figures for output currents are estimates on my part, as I have no specific figures from Victor.

The input rectification process of this supply utilizes a full-wave voltage doubler when operating from 110 VAC lines and a full-wave bridge when operating from 220 VAC lines. In either case, the DC output voltage of the filter section is about 308 volts ( $110 \times 1.4 \times 2$  for the voltage doubler or  $220 \times 1.4$  for the bridge). To change the supply from 110 VAC operation to 220 VAC operation, simply remove the jumper wire across C109.

The output rectifiers are typical half-wave circuits followed by pi-type LC filters with the +/- 12 volt regulated

outputs utilizing standard three-terminal commercial regulator devices. Protection from an excessive +5 volt regulated output is provided by the silicon controlled rectifier (SCR) A103 (a "crowbar" circuit) that senses any excessive voltage and shorts the +12 volts unregulated output to ground, thereby causing the supply to fail. However, should this action occur, the power supply will need servicing, both because of the excessive voltage problem and because one or more additional parts will have been destroyed by the shut-down process. So, while the Victor 9000 power supply may be fail-safe, it certainly doesn't fail safely. Also, note that the output rectifiers are quite different from the rectifiers on the input side of the supply and should be replaced with only low-forward-drop, fast-recovery diodes. Conventional rectifiers will not work in this application.

The heart of the Victor 9000 power supply consists of the oscillator, the switching circuit, and the feedback circuit that maintains the +5 volts at the proper level. If the +5 volt level changes, the feedback circuit causes the output of the switching supply to change in such a manner that the +5 volt output is brought back up to the proper level. Note that when the +5 volt supply is provided with more (or less) drive, the input voltage to the other three supplies will also increase (or decrease); but, because the +/- 12 volt supplies have their own internal regulators, the output of those supplies will remain unchanged. Thus, if the +5 volt supply is at the proper level, all the rest of the supply voltages must be correct, unless there is a defective component in one of the supplies. This self adjustment capability of the +5 Volt supply tends to keep the other three supplies operating at the proper voltages. This allows the Victor 9000 to continue operating during "brown outs" down to about 90 VAC, a level that would undoubtedly cause "that other machine" to unceremoniously cease operations. In fact, as noted below, when operating under light test loads, the Victor 9000 supply will operate down to about 60 volts; however, it should be noted that long term operation at low voltages can be hazardous to the fan in the power supply, inasmuch as it is not very tolerant of low line voltages.

The Victor 9000 supply has two completely separate ground systems. The input side of the switching circuit has an AC ground (NOTE THAT THIS IS NOT THE GROUND FROM THE GROUND PIN OF THE POWER CORD); whereas, the output side of the circuit utilizes standard chassis ground. In the interest of safety, it is recommended that the unit under test be isolated from the rest of the

computer and connected to an isolation transformer (as shown at the end of this article) and that test equipment be connected to only one of these grounds at a time. Thus, if you use an oscilloscope to trouble shoot the supply, be sure to connect the common scope terminal only to the AC ground when working with the oscillator circuit or power transistor and only to the chassis ground when working with one of the four DC supplies. If you do otherwise, you are courting disaster.

Perhaps you are now wondering how the supply can possibly work if one half of it is isolated from the other half. Power from the switching circuit is transferred to the four output supplies via the transformer. Information about the level of the +5 volt supply is fed back to the oscillator of the switching circuit via opto-isolator A101, which has a LED (light emitting diode) in its input circuit and a light sensitive transistor in its output circuit. Since there is only a beam of light connecting the input of the opto-isolator to the output of the opto-isolator, isolation capabilities of several thousand volts between input and output are obtainable using these devices. Thus, a feedback path has been established between the two sides of the Victor 9000 supply even though there isn't a direct connection.

If my experience with this supply is any indication, any time the supply fails, you will have to replace R106 (1 Ohm resistor) and Q103 (SJ6942 power transistor) plus any other parts that may have precipitated the failure. Since you are unlikely to find the SJ6942 in a local parts store, I have listed some recommended commercial replacement parts for this and other items at the end of this article.

Replace any defective components you find, then hook up the power supply printed circuit card as shown below. Using the Variac, gradually increase the voltage. When the input AC reaches about 25 Volts, the lamps on each output should begin glowing, indicating the presence of voltage. Continue to increase the voltage up to about 60 Volts and the lamps should now be glowing about as bright as they will get. If the +5 Volt and the +12 Volt unregulated supply bulbs begin to glow brighter as you continue to raise the voltage, back off on the Variac immediately, as you have a problem because the feedback circuit is not providing proper voltage control. (If you continue raising the Variac voltage under this condition, you will almost certainly destroy both R106 and Q103.)

If you have raised the input voltage and none of the lamps indicate the presence of voltage (and you know for sure that there is about 300 VDC on the collector of Q103), you probably either have a defective oscillator circuit or one (or more) of the half-wave rectifiers (D112, D115, D118, or D121) is shorted. To determine which

problem exists, connect an oscilloscope to the cathode (band end) of D110 (i.e., the base of Q103) and slowly bring the Variac voltage up to 110 Volts. If the oscillator circuit is dead, you should see a straight line on the scope, whereas you should see some oscillatory action (about one-half volt peak-to-peak) trying to take place if the problem is an excessive reflected load caused by a short in one of the rectifier diodes.

Obviously, if one of the lamps does not light, you have a problem in that particular output supply. One sneaky problem I have found in this area was a cold solder joint at the point the transformer winding to that particular supply connects to the printed circuit board.

I hope you now understand why I recommend the use of a lamp for each supply as an indicator of its output. The lamps are certainly much cheaper than four separate meters and, in my opinion, it is much easier to spot problems with the lamps than with meters.

I have found that C111 (a small electrolytic capacitor in the emitter circuit of Q103) was defective on every inoperative supply that I have worked on. Because of the low value of the parallel resistor R114, this capacitor must be removed from the circuit for proper testing. Do not rely on an in-circuit test to check this capacitor.

I have found that some of the defective dental tools cast off by my friendly dentist to be excellent tools for use in printed circuit board repairs. After I have unsoldered a component, I use a straight dental tool with a tapered point and while my soldering iron melts the solder around the hole the component came out of, I gently push the tool through the hole (from the foil side to prevent lifting the foil from the board) and then remove the iron. This leaves a nice hole for the replacement component, because the solder will not adhere to the stainless steel dental tool. These tools come in many shapes, but for this operation, select a small straight one, and if necessary, break or grind off the end so that you have a tool suitable for the job. You can also purchase a tool that will do the job, but I find the small dental tools to be better than some of the tools you can buy for this purpose.

A word of caution about working with the Q103 power transistor. This device is seated on a heat sink which helps it dissipate the heat from the transistor. Between the transistor case and the heat sink is a silicon compound that helps transfer the heat from the transistor to the heat sink. When replacing the transistor, be sure to place a thin film of similar material on the bottom of the transistor before it is mounted onto the heat sink. CAUTION -- DO NOT GET ANY OF THIS MATERIAL IN YOUR EYES AS

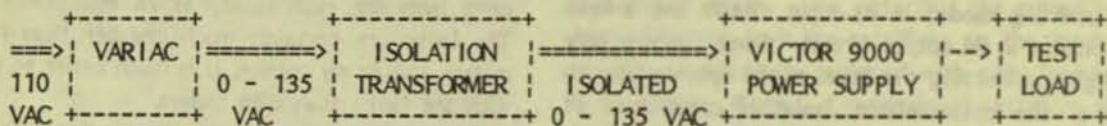
THERE IS NOTHING THAT WILL DISSOLVE IT WITHOUT HARMING THEM. And remember that the screws holding the transistor to the heat sink also transfer power to the transistor; therefore, do not attempt to apply power to this circuit before these screws fully secure the transistor to the heat sink. To do otherwise, could destroy the transistor. Also, DO NOT touch the heat sink or output transistor while this supply is running as they are HOT, both from electricity (approximately 300 VDC) and temperature.

wattage soldering iron of about 37 watts or less for general repair work. You may find, however, that you'll need about 100 watts for some large parts such as the transformer terminals. But, do not use anything larger than the 37 watt size for general repair work on the printed circuit board. Also, use only rosin core solder and use it sparingly, making sure it doesn't spread to adjacent terminals.

A final thought -- don't blindly do anything without first thinking about the consequences that may result. And DO BE CAREFUL, as the voltages on the input side of this supply are LETHAL.

A word of caution about soldering. Use a low

RECOMMENDED SETUP FOR TEST & REPAIR OF VICTOR 9000 POWER SUPPLY



I use the following connectors to connect the isolation transformer to the power supply and to connect the power supply to the test load: 8 Pos. DIP Solder Connector (0.156 Centers) GC #41-208 or MOLEX #09-52-3083 or Equivalent

Output Voltage	Pin #	Lamp # & Rating	Resistor	Wattage	Current
+12 Reg	1	382 (14V @ 80 ma)	40 Ohms	5 Watts	300 ma
+ 5 Reg	2,3	328 (6V @ 200 ma)	4 Ohms	10 Watts	1.25 Amp
+12 UnReg	7	382 (14V @ 80 ma)	20 Ohms	10 Watts	600 ma
-12 Reg	8	382 (14V @ 80 ma)	None (Lamp load is sufficient)		
Common	4,5,6				

Note 1: Lamp numbers are recommendations only. Object is to provide lamps of approximately equal brightness for each voltage output. Cheaper lamps may be substituted for those listed.

Note 2: Resistor values and wattages are recommendations only. Object is to provide a load of approximately 20% to 30% of rated output. After the supply has been repaired, it should be checked out at loads that more closely approximate the rated value.

Note 3: I recommended the following commercial replacements for:

Q101 (SU374) unijunction: replace with 2N4871

Q103 (SJ6942) power transistor: replace with SK3467 or ECG283

C108 (.0033 uf @ 600V) capacitor: replace with .0033 uf @ 1600V or a 600V capacitor having low dielectric losses, otherwise the replacement may be destroyed by internally generated heat

R106 (1 Ohm) resistor: replace with 1 Ohm, 1/4 W flameproof res.

D115, D118 (80SQ035) diode: replace with ECG580 or similar low-forward drop, fast-recovery diode. Can also replace D121 if a cheaper diode is not readily available. DO NOT SUBSTITUTE REGULAR POWER DIODES as replacements for these units!

A101 (?) opto-isolator NPN transistor: replace with 4N25

\* \* \*

# EXPLORING MS-DOS, Part 4 / George Anderson

In the last issue, we examined the first portion of DOS's external commands. This time we will deal with the second portion of the external commands. To pick up where we left off, we will begin with the MODCON command, found in DOS 2.X only.

## MODCON

The MODCON command is usable in all major versions of DOS (i.e. DOS 1.X, 2.X and 3.X). It was originally not a part of DOS, but was found in the Programmer's Toolkit (and perhaps in other software packages for the V9000). In DOS 2.X MODCON was included, but in DOS 3.X it again disappeared. The command name stands for Modify Console, and is used to modify the Victor keyboard (all keys are programmable) and the character set printed to the screen. It does not modify the characters printed on the printer. Those are defined by the printer ROM, or by software designed to access the 'downloadable' font memory in the printer.

MODCON deals with two variables: the named source, and the named file to be saved as it is replaced by the source file. The source file may be a keyboard (.KB) file or a character set (.CHR) file, or both.

Keyboard files are used to change the letter or symbol typed when a specific key is pressed. For example, a .KB file may assign the symbol '@' to the 'K' key. With this keyboard file (or a character set file) installed through the use of MODCON, each time the 'K' key is pressed, the '@' symbol is typed. Keyboard files are found in SYSGEN in the Programmer's Toolkit II. Existing keyboard files may be modified using KEYGEN, found in Programmer's Toolkit II or in Graphics Toolkit II.

Character set files contain different fonts for display on the screen. If a character set file is involved, the letter font or style showing on the screen will vary from the standard. There are two types of character sets, one to be found in the Programmer's Toolkit II, and the other to be found in CHARGRAF in Graphics Toolkit II. EFONT is a character font editor, and may be used to create new characters or to modify existing characters. EFONT is found in both of the Toolkits mentioned above. While these sources are not available to all, there are some .KB and .CHR files available on the Victor\*Group public domain disks. By getting the disks, one can make use of some of the options for change we are describing here.

When MODCON is invoked, a source file without an extension will load both the keyboard file and the character set file with the same name. To load only a .KB file or .CHR file, simply add the appropriate extension to the file name desired. The same principle applies to files to be saved. No extension saves both types of files, while addition of the proper extension will save only that file type. The example below shows how MODCON is invoked:

EXAMPLE 1: To load a keyboard modification file named BOXSET.KB, and to save the current keyboard to a file named SAVKEYS.KB, type:

```
MODCON BOXSET.KB SAVKEYS.KB
```

Naturally, you must pay attention to where the source file is located; and, if using subdirectories on your disks, provide a path to the proper subdirectory or DOS will not know where to look.

## MORE

MORE is a pipe command that delivers one screenful of data at a time to the monitor screen. To use it, be sure that MORE is on the current drive or that there is a path to it from the drive in use. Pipe commands use that interrupted vertical line, '|', to connect two commands to be performed in sequence, and that is the way one uses MORE. A command, followed by the vertical line (be sure you use a space on either side of the vertical line - sometimes it will not work if you don't leave the space), followed by MORE will present the data appropriate to the command, a screenful at a time.

EXAMPLE 1: To show all (and only) the directory names from the directory, type the following:

```
DIR | FIND "DIR" | MORE
```

Where: Two pipes are used - one to FIND the string sequence "DIR" in the directory (DIR), and the other to invoke MORE to limit the resulting presentation to one screenful at a time.

## MV

MV combines the actions of COPY and DELete. First it copies the source file to the location you specify (a disk or subdirectory). Then it automatically DELetes the source file. This leaves a copy of the source file at a new



location. The effect is as if you had physically moved the source file from its original position to the new position. If you named two files, the first is treated as the source, and the second as the destination. If data exists in the destination file, it is overwritten with the copy of the source file, just as would be the case using COPY. Wild cards may be used. Be sure the path to the new location is specified either with a separate PATH command, or by including the path in the file name statement. An example is given below:

EXAMPLE 1: To move a file named REQUEST.COM from the root directory of drive C:\ to the subdirectory named BUSINESS on drive C:\, type the following:

```
MV C:\REQUEST.COM C:\BUSINESS
```

Where: MV is the Move command, C:\REQUEST.COM is the source file, and C:\BUSINESS is the subdirectory into which REQUEST.COM is to be moved. If the current directory is already the root directory of drive C:\, you do not need to specify the C:\ for REQUEST.COM.

The MV command works between any drive and any other drive.

EXAMPLE 2: To move all .COM and .EXE files from drive C:\ to drive B:\, type:

```
MV C:\*.COM C:\*.EXE B:\
```

MV also will perform the function of renaming a file.

EXAMPLE 3: To rename a file (CHUCK1) on drive C:\, to CHARLES1 while placing it on drive B:\, type the following:

```
MV C:\CHUCK1 B:\CHARLES1
```

## PORTSET

PORTSET is an external command which allows resetting of the serial port parameters. With it you can change the default settings which were placed in memory at bootup with PORTA and PORTB commands. Refer to these latter commands for the format for changing the baud rate, bits per character, parity and number of stop bits. With PORTSET it is not necessary to call up PORTA or PORTB, change it, and then reboot in order to change the settings for the serial ports.

## PRINT

PRINT is an external command which 'lines up' one or more files to be printed, and proceeds to print them while DOS continues with another task. The PRINT command, therefore, looks almost like a multitasking command in that you continue doing other tasks while the printing is going on. It is limited to placing in the queue only files on the current drive or in the current directory. While printing is taking place from the queue, DOS cannot access the printer for alternate tasks. Alternate tasks which do not require the printer, however, will continue to perform.

The first time after bootup this command is used, a message is displayed on the screen:

```
Name of list device [PRN]:
```

Of course, PRN means Printer; and you are to name your printer as LPT or COM1 or COM2 in response. Instead, you may press RETURN, since PRN is the same as LST, which is defined in your CONFIG.SYS file. When you respond with a valid option, DOS returns another message on the screen:

```
"Resident part of PRINT installed."
```

Otherwise it responds with:

```
"List output is not assigned to a device."
```

meaning that the response you gave was not a listed device assigned and recorded for the computer to access and check.

There are three option switches available for use with PRINT. They are /C, /P and /T, standing for Cancel, Print and Terminate, respectively. The PRINT command is invoked as follows:

EXAMPLE 1: To list three files to print, type:

```
PRINT TEXT1.DOC IHC13 WALLTR8
```

If you decide to cancel the PRINT command for only the first of the three, but print the other two, it may be done by inserting /C after the first file. However, this is a toggle, meaning that it will remain in force until action is taken to change it. In this case we still want the second and third files printed, so we will use the /P switch as well:

EXAMPLE 2: To list four files, cancel the first two and print the balance, type:

```
PRINT TEXT1.DOC /C IHC13 IHC14 /P  
WALLTR8
```

Where: The /C switch cancelled the preceding file (the first file) and the next file after the /C switch as well (it would continue to cancel subsequent files except for the /P switch); and the /P causes the file before it and all files after it to print until a further switch is invoked. The placement of the switches is important. The rule to remember is that the switch operates on the file before it and all files after it until another switch takes over.

The /T switch may be used at any time during the printing to dump the queue and terminate printing. No other parameters should be used with the /T switch, nor are they needed, since this switch terminates the previous action.

### RDCPM

This is an external utility program designed to read a CPM disk from MS-DOS. The format is:

```
RDCPM (filename.ext)
```

Using this format, the specified CPM file would be copied to the default directory of the default drive. If a drivename precedes RDCPM, the file will be copied to the default directory of that drive. By inserting DIR after RDCPM, the CPM directory will be displayed. Wild-card characters may be used in the normal manner.

### RECOVER

The RECOVER command is supposed to recover a file from a bad disk. For example, there may be an area which has gotten scratched and has lost its magnetic media in the scratch area (and the data stored there) after having written to the disk. Be careful with this command. It 'RECOVERS' by performing a sector-by-sector copy of all the readable data in the file. IT ALSO RENAMES the file.

I recommend making a DISKCOPY (also sector-by-sector copy) of the whole disk having the lost data problem, then working ONLY on the copied disk. If you have to, you can still go back and make another DISKCOPY of the bad disk and try again. Using the DISKCOPY procedure also means that you do not lose the file name. RECOVER is better than nothing, but barely! At best, some portion of the original file (the part in the damaged or defective area) will be lost.

RECOVER will mark the damaged sector to prevent writing to it again - if you do choose to use it on the original disk. It is invoked by typing the command name and the drive. If a single file is bad, then type the command name, the path

(if any) and the file name to be recovered. The renamed file has the naming convention:

```
FILEnnnn.REC
```

Where: nnnn is a sequential number for the recover session, starting with 0001.

### RESTORE

RESTORE is used in conjunction with BACKUP, and copies all files and subdirectories onto a hard disk from a BACKUP disk. There are two option switches available with this command. The /P switch asks for confirmation before writing over files having a Read-Only flag or files which have been modified since the backup was made. The /S switch must be used with RESTORE if it was used on BACKUP. /S RESTORES all files in their original subdirectories so that everything is where it was before BACKUP was performed. /P does not, since it was designed to work with individually BACKedUP files.

EXAMPLE 1: To RESTORE files from a floppy disk in drive A:\ to drive C:\ and preserve the subdirectories (/S was used at BACKUP), type:

```
RESTORE A: C:\ /S
```

EXAMPLE 2: To RESTORE files from and to the same drives as those of EXAMPLE 1, but with the Prompt option to avoid overwriting certain files, type:

```
RESTORE A: C:\ /P
```

Where: A: is the source drive, C:\ is the destination drive - root directory, and /P invokes the option to prompt each file restoration so that, if restoration for that file is not desired, RESTORE will skip that file.

### SDCOPY

SDCOPY is an external command to copy from one disk to another disk in the same drive. (It is intended for use on machines having only one floppy disk drive.) Its format is simply SDCOPY followed by the drivename of the floppy disk drive. Its use requires swapping between source and destination disks one or more times (depending upon size of disk contents and available memory).

### SEARCH

The SEARCH command searches files with options to

specify the following:

- \* Where to search (Domain) [files or directories to be searched]
- \* Which things to search for [matching constraints]
- \* What action(s) to take [lists matching files if no action specified]

SEARCH covers a lot of ground. (This description covers almost four pages!) It may be of most application with a hard disk; however, it may be used with floppies only. It is a multi-purpose command with which one may use option switches to:

- \* Move files
- \* Change file attributes
- \* Copy files
- \* Sum size of groups of files
- \* Delete files
- \* Delete empty directories
- \* Archive files

Further option switches allow one to:

- \* Invoke prompts
- \* Exclude specified subdirectories from the search
- \* Copy 'entire subdirectory hierarchies'

As with DOS commands, either upper case or lower case will do.

#### WHERE TO SEARCH (Files or Directories)

SEARCH will search specified files or directories. It can do this either specifically or generically. If done specifically, wild cards may be used in place of the actual names. If done generically, the search begins at the actual current directory on the drive specified, unless told differently.

For example, if drive A: is specified, and A:\BATCH is the current directory, the search will start at A:\BATCH and work downward through the rest of the subsidiary subdirectories. If you want to start at the root directory, specify A:\. Individual files may be intermixed with subdirectory specifications, but you must be sure that the path to each is clearly spelled out.

EXAMPLE 1: To specify a search for a subdirectory on drive A: and two files on drive B: when the default drive is B:, you would type:

```
SEARCH A:\SUBDIR4 MYFILE YOURFILE
      (then add parameters for Which and What)
```

Where: The first Where parameter is A:\SUBDIR4, a subdirectory on drive A:. Any subdirectories below

A:\SUBDIR4 will also be searched. The two files on default drive B: to be searched are MYFILE and YOURFILE.

#### WHICH (Matching Constraints)

The Which part of the three parts of SEARCH consists of several classes of constraints. They are:

- o The class naming files
- o The class dealing with file attributes
- o The class dealing with file comparisons
- o The class dealing with file date comparisons
- o The class dealing with file size comparisons

#### Naming Files

The class naming files consists of /NAME, /^NAME, /FNAME and /^FNAME. The caret mark '^' stands for 'NOT'; so we have NAME and NOT NAME, FNAME and NOT FNAME in this class. To separate file names in the specification, use an exclamation point (!!) with no spaces around it.

/NAME specifies files from the domain (remember that 'domain' is the 'Where' constraint we selected first (above)) which match the file names specified by it. The selected files will then have action taken, based on the 'What' portion of the SEARCH command, which comes last.

/FNAME differs from /Name in that it specifies files and their path names, rather than just file names. As with /NAME, action will be specified in the 'What' portion of the SEARCH specification.

/^NAME and /^FNAME are the 'NOT' versions of their counterparts, above. /^NAME specifies all files which do not match the file name(s) following it. /^FNAME specifies all files whose full path names (including file name) do not match those following it.

EXAMPLE 2: /NAME is invoked to specify all batch files on drive A: and all data files on drive B: - type the following:

```
SEARCH A:\ /NAME *.BAT B:\ /NAME *.DAT
```

Where: SEARCH first defines where it wants to search; i.e., the root directories of both drive A: and drive B:, and down through the hierarchy from the root directory on both drives. Next, it names what it wants to search for; i.e., the names specified after

/NAMES. Then it states the names it wants to match - all files ending in .BAT on drive A: and all files ending in .DAT on drive B:.

With the four naming switch options, there are several wild card options that may be applied:

- ? Provides a single character match.
- \* Matches a character string of any length. The \* acts differently from its use in DOS commands. There it stands for a string of any length. Here it stands for an explicit string that you supply. The string is contained between two asterisks.
- /. Usually used with an asterisk. When combined as '\*/.', it extends the match criteria to the extension of the file name.
- \/ Used to match the full path of subdirectories, rather than a generalized path using an asterisk for a wildcard subdirectory name. If you wanted to match three levels without knowing that three was the right number, using \/ would automatically search the three existing levels.
- [ ] Gives a multiple choice search. By placing the letters inside the brackets that represent the first letter of all the files you want selected, SEARCH will pick them out for you. This means that if the file names desired started with S, R, F, and G; then placing them in brackets: [SRFG]\*.DAT would match any .DAT file whose name started with S, R, F, or G. Notice the similarity to a wild card calling for all .DAT files: \*.DAT. Here, by placing letters inside the brackets, and then specifying that those letters apply to \*.DAT files, the search is restricted to file names starting with the letters in brackets, for .DAT files only.

There are many variations on this. By combining the various switches, above and below, one can do amazing things - often with a great amount of effort. The key to success with this one, if you want to use its complex combinations, is to practice with it often, until you do not have to look up the combinations you find most useful.

#### Attributes

The next class deals with file attributes. File attributes are things you are told about, but seldom see. There are five of them - each defined, for purposes of the SEARCH command, by a letter. The five are:

- A Archive status bit is set 'ON', meaning that the file has been edited since the last backup.
- D Subdirectory.
- H Hidden file.

- R 'Read-Only' file.
- S System file.

File attributes are specified following the /ATTR switch. There can be more than one attribute letter in the string. To specify more than one letter, use the '&' sign between them. There are times when you may want to specify one OR another. To do this, place an '!' symbol between the letters. To specify that you do NOT want a particular letter, use the '^' in front of the letter representing the condition you do not want in the files specified.

EXAMPLE 3: To select all files with an archive bit set to ON, except for the Read-Only files, on drive A:, type:

```
SEARCH A:\ /ATTR A&^R
```

Where: SEARCH A:\ determines the drive to be searched, starting at the root directory (\ for root); A specifies that you want to select files with the archive bit ON; '&' says you also want another specification to apply to the search; '^R says the other condition specified is that the files selected should NOT be marked as Read-Only.

If you should want to inspect the current file attributes, use the DOS LS command with the /L switch to give you a long version listing. This will include the attributes. As an alternative, you may use the SEARCH command with the /LS action switch.

#### Comparisons

The /COMP switch, followed by a directory or subdirectory name, will cause SEARCH to compare all the files in the domain selected (the SEARCH area specified in the 'Where' portion of the command) with all the files named in the /COMP switch. The selection will then consist of the files whose names are identical, and whose content is equivalent. If differences are found, they are displayed. To preserve the tree structure of the files being compared, use a /T switch. An example will show how these switches are invoked.

EXAMPLE 4: To search drive A: starting at the current directory (not necessarily the root directory), and compare it with the file path starting with B:\MESSAGES, type:

```
SEARCH A: /COMP B:MESSAGES /T
```

Where: the search will take place on all subsidiary levels of both drive A: and B:\MESSAGES.

The /<sup>^</sup>COMP switch makes its selection if one of two conditions are found:

- o No file of the same name is found.
- o A file with the same name is found, but the content of that file is not the same as the specified file.

### Exists

The /EXISTS switch is similar to the /FNAME switch, but does not use the wild card options available to /FNAME.

### Time

The time related switches are /LE, /LT, /GE, /GT, /LEDATE, /LTDATE, /GEDATE, and /GTDATE. The meaning of the first four are:

- /LE = Less than or Equal to
- /LT = Less Than
- /GE = Greater than or Equal to
- /GT = Greater Than

The other four, with DATE appended to the switch name, restrict the switch to selecting on the basis of date only, ignoring time. The four without DATE use both date and time in the selection.

### Size

The size comparison option switches specifies on one of two criteria when invoked. The /SIZE n switch specifies files at least as long as n bytes. The /SIZEB n switch specifies files whose length is at least n blocks long, where a block is 512 bytes.

Some examples of SEARCH commands with 'Where' and 'Which' option switches follow:

EXAMPLE 5: To match any file on drive C: except for .EXE files, use this phrase:

```
SEARCH C:\ /^NAME *.EXE
```

Where: SEARCH C:\ starts searching at the root directory of drive C.; /<sup>^</sup>NAME says all files except those named following; and \*.EXE names all .EXE files.

EXAMPLE 6: To match files, with those on drive C:, beginning with \ACAD and continuing through any subsidiary subdirectories (if there are any), whose file names do not begin with A (for ACAD) or C (for CAD), use the following command:

```
SEARCH C:\ACAD /^NAME [AC] *.*
```

There are two lines of search using /FNAME. The first line of search, if started at the root directory level, will follow a path by searching all first level subdirectories, but not those below the first level.

EXAMPLE 7: To search a complete path specification to the first level subdirectories using /FNAME, type:

```
SEARCH B:\ /FNAME [ALRD]*
```

EXAMPLE 8: To search subdirectories at all levels on drive B:, type:

```
SEARCH B:\ /FNAME [ALRD] \\^*
```

These examples should suffice to show the principles involved. SEARCH is so flexible in its capabilities that all combinations and permutations would fill a book by itself. Perhaps this glimpse will give courage to try a few of the options available. Nothing can get hurt if you use dummy files, or copies of good files made up just for this exercise. Good hunting on this one.

### WHAT (Actions)

The optional action switches perform some action on the files selected. They can set attributes ON or OFF with the /ATTR+ and /ATTR- switches, respectively. Several archive actions may be taken using /AUTOTARC, /TARA, /TARC and /TARX action switches. /TARC creates a TAR (tape archive) file with the specified file name and extension. Do not be concerned that this switch will only operate on tape drives. It will work perfectly well on disk drives. /AUTOTARC creates a file similar to /TARC, but with path information preserved and included. /TARA appends to a /TARC or /AUTOTARC file. /TARX extracts files from /TARC or /AUTOTARC files. /TARC and /AUTOTARC are mutually exclusive - you cannot use both at the same time.

/BACKUP copies files with the archive bit set to ON, then resets the archive bit to OFF after the copy. Use a /T switch with /BACKUP if you want to preserve the tree structure of the subdirectories.

/CP followed by a directory name copies all the files matching the specification into the specified directory. This switch can use the /T option to preserve tree structures.

`/LS` lists a specified directory in long format, giving the pathname, the attribute status, size, date and time for each file selected.

`/MV` followed by a directory name moves the files selected into the directory specified. It deletes the files at the source location.

`/PRINT` is a default for `SEARCH`. It prints the names of matching files selected by `SEARCH` to the screen.

`/RM` removes the matching source files. This can be used to delete duplicate files.

`/RMDIR` removes all directories which are empty. If a directory was emptied with a `/MV` switch, then `/RMDIR` takes place after all the files have been moved.

`/SUM` adds up the total number of bytes (not rounded off) in a group of files matching the `SEARCH` specification.

#### MODIFIERS

(Action and/or Constraint)

- `/A` matches files if their attribute status bit is ON.
- `/N` means NO ACTION. This modifier will display what would have taken place if the `/N` had not stopped it.
- `/NR` means NOT RECURSIVE; i.e., the search is limited only to the directories specified, and not to the subsidiary subdirectories as well.
- `/P` sends prompts to the screen and allows you to avoid the proposed action, if appropriate. Works with `/CP`, `/MV`, `/RM`, `/RMDIR`, `/TARA`, `/TARC`, `/TARX`, `/BACKUP`, `/ATTR+`, and `/ATTR-`.
- `/S` is silent. It displays nothing. You might not want anyone to see what is going on!
- `/T` tree structure is maintained. Works for `/MV`, `/CP`, `/COMP`, `/EXISTS`, `/TARX` and `/BACKUP`.
- `/V` keeps both volume and drive positions for `/TARX`'d files. They will be put back in the same order they were originally.
- `/.` allows wild cards to include the file extension separator (the dot)
- `/\` allows wild cards to include the directory separator (the backslash)

Some do's and don'ts for using `SEARCH`:

- o Remember that `SEARCH` always starts at the current directory unless you specify the root directory.
- o You can copy modified files only, with `/BACKUP`

and avoid the time copying files which have not changed.

- o Watch out for the restrictions placed on certain switches. Those that are abbreviations for a longer string of switches and commands cannot violate the rules for using the longer version.
- o Files may be searched for by date, or by date and time. See section dealing with date and time comparisons.
- o Hidden and or system files will list with the proper switches.
- o You may find the exact size of a group of files to help determine if enough room is available before starting to move, copy or archive files.
- o `SEARCH` manages end-of-disk space for `/TARC` or `/TARA`.
- o `SEARCH` will copy entire subdirectories, helping you to rearrange subdirectories.
- o Empty directories may be removed, using several commands and switches on the same command line.
- o Duplicate files may be found and removed.

Overall, `SEARCH` is a very powerful, flexible tool. It duplicates many of the other DOS commands within its structure, but offers a method to perform the equivalent of many separate commands on a single command line. The cost is its complexity, and therefore the time it takes to become proficient with it. There are a few things that it will do that other commands are not available to do.

#### SHARE

The `SHARE` command allows the sharing of files in a networked group of computers. It is of no use in single computer installations.

#### SORT

The `SORT` command is used in conjunction with other commands to sort data accessed by those commands. The most common usage is in producing a sorted directory. This may be done by:

`DIR | SORT`

Where: The directory data is piped to the `SORT` command, gets sorted alphabetically, and then is sent to the screen.

If one wanted a sorted list of subdirectories only, one would have to get the directory data, find the subdirectory entries, sort them, and then display them on the screen. This can be done by:

## DIR | FIND "< DIR >" | SORT

Where: The source of the data and the output of the data to SORT are indicated by the '<>' pair around the "DIR" portion of the command line.

To reverse the order of the sort, invoke the /R option switch. SORT works on either alphabetic or numeric data. To start a SORT in a certain column, use the /+n switch, with n being the column number where the sort is to start. The output of SORT may be directed to a printer by:

DIR | SORT > LST (the list device, the printer)

## SUBST

The SUBSTitute command substitutes one drive identifier for another. For example, if a program makes a fixed assumption that all program files are on drive A:, and you have those files on a hard disk named drive C:, the program bombs. SUBST to the rescue! Follow the command with the new drive identifier (C:, in this case), then the old drive identifier (drive A:, here). It looks like this:

SUBST C:\WORD A: (if that is the proper path on C:)

The /D switch disables the substitution. It needs the first or new drive identifier to perform this feat of derring-do.

You cannot use ASSIGN, BACKUP, FDISK, DISKCOMP, DISKCOPY, FORMAT, LABEL, RESTORE or JOIN while SUBST is invoked. Further, use CHDIR (CD), MKDIR (MD), RMDIR (RD), or PATH only with caution. Look to see if the results make sense in that situation, before you accept them.

## SYS

SYS transfers the DOS system files from the disk in the default drive to the disk in the drive specified. It is invoked with the drive identifier for the drive on which you want to install the system files. The default source drive takes care of itself.

## TAIL

TAIL prints the end of a specified file to the screen. If no other specifications are included, TAIL prints the last ten lines of the file. To print more or less than that, add the /n option switch, where n is the number of lines at the end of the file that you want printed. If you want the display to show n bytes instead of n lines, then use the /B switch after the /n. If n is not specified, then 512 bytes is the default.

## UNIQ

This command filters the input and compares adjacent lines. If you SORT the file first, then use UNIQ, any adjacent lines that are the same will appear in the output. There are three switches that may be used with UNIQ:

- /U lists only lines that are not duplicated.
- /D lists the duplicated lines, but does so only one for each duplication.
- /C lists each line. It gives a count of how many times it occurs, thus allowing finding all occurrences.

## WC

Word Count is a command that counts words, or lines, or words and lines in one or more files. It outputs the word count or line count and the file name. If more than one file was specified, it does this for each file, then gives a total word count or line count. There are two switches for use with WC:

- /R includes the counts for all files in all subsidiary subdirectories from the start directory.
- /W omits word count to speed up the operation. Counts lines only.

That ends the discussion of external commands for this issue. Next time (in Part 5) we will address batch files, how they are done, what they are good for, and examples you can use.

You may notice that four external commands that were listed (in Part 3 of this series) as being a part of MS-DOS 3.1 were not covered here. They will be covered in Part 5, prior to the discussion of batch files, along with a rationalization of the discrepancies in DOS on disk versus DOS documentation, versus DOS as seen through the eyes of DISKID on the DOS disks.

\* \* \*

- - -

## ERRATA

In "EXPLORING MS-DOS, Part 3" (VictorTalk, Vol. 6, No. 4, Jul-Aug 1988), on page 6, under DCOPY, DISKCOPY was erroneously referred to as an internal command. It should have been referred to as an external command.

\* \* \*

# BACK TO BASIC / Bob Elford

[Herewith are presented two short programs in BASIC. They may be useful just as they stand; but, perhaps more important, they can be used as simple programs that you can manipulate to gain experience in BASIC and customize to your desires. The programs were written in VBASICA (which comes with MS-DOS 3.1); however, they will run perfectly well in GW-BASIC; and, with the deletion of the CLS (CLear Screen) command, will also run in MS-BASIC (which comes with both MS-DOS 1.25 and 2.11).

Some suggested changes to gain experience:

1. Delete the line-spacing LPRINT statements (to save paper by eliminating double-line spacing).
2. Change LPRINT statements to PRINT statements temporarily (to save even more paper while making changes, by directing output only to the screen).
3. Align comment statements vertically, to facilitate reference.
4. Align initial quotation marks vertically, so format can be more easily visualized from listing.
5. Change the format statements to add some desired information, make the output form more esthetically pleasing to you, or better conserve space, etc.
6. Modify MILAGE.BAS to avoid garbage being printed if less than the specified six check points (markers) are entered.
7. Make other changes that will make the programs more useful to you, or easier to read.

If you need help with BASIC, dig out your back issues of VictorTalk. ("MS-BASIC, Part 1" through "MS-BASIC, Part 5", VictorTalk, Vol. 5, No. 1, Jan-Feb 1987 through

Vol. 5, No. 5, Sep-Oct 1987.)

A note on units: In WATER.BAS, meter readings are given in 100's of cubic feet, while usage is given in gallons/day. In MILAGE.BAS, "average realized speed" is in miles/hour; the output table has a complex format wherein the table is diagonally divided in two parts. In the upper right part, the units are miles (cumulative from the specified marker in the left column); in the lower left part, the units are hours and minutes (cumulative from the marker at the column heading).

Let us know if programs of this sort are considered useful, either for their intended purposes, or as exercises in BASIC program manipulation. A column of this sort could become a regular feature of VictorTalk. Toward this end, contributions that might be suitable for such a column would be most welcome. - Ed.]

## A TABLE OF WATER USAGE

Those of us faced with water rationing on a meter often find it helpful to read the meter at intervals to see whether we are keeping within the limits set by our Water Company. To help with this checking I have worked up a short program that gives me a printout of the maximum meter reading permitted under a given value of water usage. If I like I can print tabulations for several "goals," so I can see if I am able to keep within some lesser usage I may set for myself. Perhaps the program will be helpful to you, too.

```
10 CLEAR
20 '-----WATER.BAS----- (Bob Elford)-----
30 CLS
40 PRINT "WATER.BAS prints the maximum permissible meter reading for each day"
50 PRINT "of the billing period."
60 '-----9-11-88----- (VBASICA)-----
70 PRINT:PRINT "Enter, on prompt, meter reading at beginning of period, date of"
80 PRINT "that reading, number of days in the billing period, name of present"
90 PRINT "month, number of days in present month, name of next month, number"
100 PRINT "of days in next month, name of third month, and number of gallons"
110 PRINT "per day permitted (See Water Company directives)":PRINT:PRINT
120 DEFDBL D:DEFSNG A:DEFINT I:DEFSTR S          '-----Define variables
130 I4=0          '-----Counter for form feed
140 INPUT "Initial Reading <###.##>";DINITIAL
150 INPUT "Beginning date<##>";IDATE
160 INPUT "No. days in billing period<##>";I3
170 INPUT "Name of present month";S1:INPUT "No days this month";I1
180 INPUT "Name of second month";S2:INPUT "No days second month";I2
```



```

190 INPUT "Name of third month";S3
200 INPUT "Gals/day permitted <###> (Use <-9> to terminate)";I5
210 IF I5=-9 THEN 390
220 DUNITS = (I5*13)/748      !-----To convert to meter units
230 S5=      "\      \ ## \      \ \      \ ## \ \ #### \      \"
240 LPRINT USING S5;"WATER USAGE for",I3,"days beginning",S1,IDATE,"for",I5,"gals/day"
250 LPRINT:LPRINT "Date      Max.Rdg.  Actual Difference":LPRINT
260 FOR I=1 TO I3
270 IDATE1=IDATE+I:SDATE=S1+STR$(IDATE1)  !-----Advance one date
280 IF IDATE1>I1 THEN IDATE1=IDATE1-I1:SDATE=S2+STR$(IDATE1)  !---Set month
290 IF IDATE1>(I1+I2) THEN IDATE1=IDATE1-(I1+I2):SDATE=S3+STR$(IDATE1)
300 DREADING=(INT((((DUNITS*I1)/I3)+DINITIAL)+.005)*100)/100!---Compute reading
310 PRINT USING "\      \ ###.## \      \ \      \";SDATE,DREADING,"      ", "      "
320 LPRINT USING "\      \ ###.## \      \ \      \";SDATE,DREADING,"      ", "      ":LPRINT
330 I4=I4+1      !-----Advance counter
340 IF I4=29 THEN LPRINT CHR$(12):I4=0      !-----Form feed
350 NEXT I
360 LPRINT CHR$(12)      !-----Form feed
370 CLS
380 GOTO 200      !      To set new gals/day value or terminate
390 END

```

#### A TABLE OF MILEAGE AND RUNNING TIME

How far can I go before I have to refuel? Where would be a good place to plan on stopping for lunch? Do we want to try to go on to the next checkpoint before we get a motel? Such questions come up regularly when I'm travelling, and I don't want to take time to look up mileage on the map and estimate the driving time. I've found it helpful to set up some tables before starting out on the trip;

a quick glance at the table will provide answers to the questions above. The table will show the mileage between any two points on the table and the running time based on whatever assumed speed I have entered. After entering the check points and mileages I usually run several tables for different speeds. I can then refer to whichever table is most appropriate. I hope the program will prove as helpful to you as it has to me.

```

10 CLEAR
20 !-----MILEAGE.BAS----- (Bob Elford)-----
30 CLS
40 PRINT "MILEAGE.BAS prints inter-marker mileage and driving time."
50 PRINT
60 !-----9-11-88-----BASICA-----
70 PRINT "Enter, at prompt, six check points beginning with the starting"
80 PRINT "point and ending with the termination point. If more check points"
90 PRINT "are desired the route may be subdivided and treated as two or more"
100 PRINT "segments. Also enter mileage from starting point and average "
110 PRINT "realized speed."
120 PRINT
130 DEFINT I,J,K,L:DEFSTR F,S:DEFDBL D,T      !-----Define variables
140 DIM J(70):DIM IT(70):DIM IH(70):DIM IM(70)  !----- Dimension arrays
150 FOR I=1 TO 6
160 PRINT I;
170 INPUT " Marker <Oakland> (Up to 8 characters)";S(I)
180 INPUT "Miles from start";J(I)
190 NEXT I
200 INPUT "Average Realized Speed <55> (Use <-9> to close)";JS
210 IF JS=-9 THEN 530      !-----To close
220 FOR I = 1 TO 6      !-----To create two-way table

```

```

230 FOR K = 1 TO 6
240 L = I*10+K
250 J(L) = J(K) - J(I)
260 NEXT K
270 NEXT I
280 FOR I = 1 TO 6
290 FOR K = 1 TO 6
300 L=I*10+K
310 IT(L) = J(L)/JS*60
320 IH(L) = IT(L)\60
330 IM(L) = IT(L) MOD 60
340 NEXT K
350 NEXT I
360 '-----Set format-----
370 SA = "\ \ \ \ \ \ \ \"
380 SB = "\ \ ### \ \ \ \ \ \ \ \"
390 SC = "\ \ ## ## \ \ \ \ \ \ \ \"
400 SD = "\ \ ## ## \ \ ## \ \ \ \ \ \ \"
410 SE = "\ \ ## ## \ \ ## ## \ \ \ \ \ \ \"
420 SF = "\ \ ## ## \ \ ## ## \ \ ## ## \ \ \ \"
430 SG = "\ \ ## ## \ \ ## ## \ \ ## ## \ \ ## ## \ \"
440 LPRINT "ROUTE: ";S(1);" to ";S(6);" via ";S(2);S(3);S(4);S(5);" Speed: ";JS:LPRINT
450 LPRINT USING SA;"Marker",S(1),S(2),S(3),S(4),S(5),S(6):LPRINT
460 LPRINT USING SB;S(1),J(11),J(12),J(13),J(14),J(15),J(16):LPRINT
470 LPRINT USING SC;S(2),IH(12),IM(12),J(22),J(23),J(24),J(25),J(26):LPRINT
480 LPRINT USING SD;S(3),IH(13),IM(13),IH(23),IM(23),J(33),J(34),J(35),J(36):LPRINT
490 LPRINT USING SE;S(4),IH(14),IM(14),IH(24),IM(24),IH(34),IM(34),J(44),J(45),J(46):LPRINT
500 LPRINT USING SF;S(5),IH(15),IM(15),IH(25),IM(25),IH(35),IM(35),IH(45),IM(45),J(55),
    J(56):LPRINT
510 LPRINT USING SG;S(6),IH(16),IM(16),IH(26),IM(26),IH(36),IM(36),IH(46),IM(46),IH(56),
    IM(56),J(66):LPRINT CHR$(12)
520 GOTO 200
530 END

```

\* \* \*

VICTOR 9000 COMPUTER FOR SALE

640 K RAM, 10Mb hard disk, CPI Memory Board,  
 Audio Preamp Board and microphone.  
 Software includes: MS-DOS 2.11, White Crane Systems  
 RAM Drive, Lotus 1-2-3, dBASE II,  
 Audio Toolkit, Programmer Toolkit I and II,  
 Graphics Toolkit, Turbo Pascal, WCS Disk Utilities,  
 Crosstalk, Async, and all manuals.  
 \$600 or best offer (plus shipping charges)

David L. Carpenter  
 The Carpenter Consulting Group  
 7732 Haskins  
 Lenexa, Kansas 66216  
 (913) 631-1303

\* \* \*

VICTORTALK AD POLICY

VICTORTALK will publish, free of charge, ads for Victor (or Victor-related) hardware and software from either Victor users or dealers. The principal requisites are that the ad be of potential value to Victor users and that space can be made available. The normal publication deadline is the first of even months (December, February, etc.). The Editor reserves the right to edit submitted material as believed appropriate. If requested before deadline, ads will be repeated. Submit material (hard copy only is acceptable) directly to the Editor at the address shown in the masthead.

\* \* \*

# FROM THE BOARDS

[FROM THE BOARDS is assembled from messages appearing on various (generally Victor-specific) bulletin boards, mainly The PUB and BAVUG, by John and Mary Jo Knobelsdorf. Only minor editing is done to correct obvious typographical errors and to eliminate some superfluous material. An effort has been made toward a degree of standardized formatting and the elimination of excess space to the extent practical without degrading legibility. Garbles that cannot be readily rectified are allowed to remain. Believing that considerable worthwhile material is contained in these excerpts, our intention is to include this feature in each issue of VictorTalk, as space will allow. This issue's FROM THE BOARDS is totally concerned with IBMBIOS and closely related topics. - Ed.]

[1] #986 (of 991) LLOYD LEBLANC, on 09-JAN-88 22:50  
(Size: 1207 Bytes)  
Subject: EMULATION

To all spreadsheet freaks: Borland's QUATTRO (new, about \$130 discount) will run very nicely on the Victor using HXTBIOS or IBMBIOS emulators (mono mode, of course), and naturally on the PlusPC and VI in CGA emulation OR mono modes. Why care? Cause QUATTRO looks like a damn good substitute for an 8087 chip or a faster IBM machine.....keep your Victor. QUATTRO significantly speeds up typical Lotus or Multiplan operations, including long recalcs. I mean SIGNIFICANTLY. Especially with a NEC V20 in your Victor.

I think someone was saying that they could not get the CGINIT.COM or IBMBIOS.COM to load properly on a V9 with a loaded WA Escort board. Well (forgive me, Marty), you can't unless you unload from MULTO, but then, you can. Even tho the docs say you must have 896k RAM in the Victor, you can get by with -- I tried it -- just under 800k. Set your Escort to load the full 896k (that turns into 832k the way WA does it), leave MULTO or other "resident" programs, and run CGINIT and IBMBIOS. It will usually work pretty well.

[1] #987 (of 991) CHRIS UMBRICH, on 10-JAN-88 01:02 (Size: 816 Bytes)  
Subject: Coexisting Valid & WA RAM

I just called VTech for some information about their RAM-cache software and especially to find out how it would work with my checkered RAM (Victor, Valid & Western Auto.). I don't know if this is common knowledge that I missed, but it turns out VT had heard about the RAM conflict of their

board with the WA board, and engineered a PROM that allows one to install the full RAM complement on their board, while allowing the WA board to access the 64K below the 896K it needs. (I had to remove some chips from the VT board, leaving me with only 640K system RAM and 1.5MB Ramdrive on the WA board. This is probably not sufficient to run the emulation software).

Am I correct in interpreting the recent messages to indicate that the emulators will run on 832K RAM?

[1] #990 (of 991) LLOYD LEBLANC, on 10-JAN-88 18:41  
(Size: 1741 Bytes)  
Subject: EMULATION/RAM

To answer Chris and others re: emulation with IBMBIOS and RAM limits:

It looks to me, after some experimenting, that you can run IBMBIOS.COM "only" and have it function as the IBM mono adapter (text), with the WA Escort cards set for the 896/832 RAM in the V9. The effective RAM I see on my machine with CHKDSK is about 797k after the OS loads. (Remember, you must vacate MULTO to run HXTBIOS and IBMBIOS).

You CAN load CGINIT.COM, and then run IBMBIOS -- but I find that, although IBMBIOS reports that it's functioning as the CGA adapter, you get no CGA "color" screen writing. I suspect this is a function of not having quite enough RAM as the program author wanted -- between 768k and the full 896k. Unless you want the CGA effect, or the program you have is CGA-only, no problem; but, it would be nice to be able to get both modes and keep the WA Escort board in the machine.

As I mentioned, you can run HXTBIOS as well, skipping the CGA question. I use the IBMBIOS mainly 'cause people like Franz Hirner have fixed up the keyboard codes and it seems smoother than the HXTBIOS I have.

I can run a lot of things I thought difficult. The latest is QUATTRO. I also have run JAVELIN PLUS, PC-WRITE, and some other things that use IBM memory in different ways. IBMBIOS handles them all. Sometimes the screen writes are a little messy (overwriting etc.) but usually it's acceptable. Graphics, of course, is out without the true CGA emulation. I don't know how successful anyone has been in getting Lotus 2.1 graphics right under IBMBIOS.

Anyone up to modifying the WA Escort device driver to shift that damned 64k RAMDISK "window" to higher than 896k? Hmmm, someone, please?

[4] #236 (of 236) JIM HUGHES, on 08-JAN-88 03:42  
(Size: 1325 Bytes)  
Subject: Accounting Packages

I have had a request from a Victor Circle member to locate an inexpensive accounting package for use on the Victor 9000. Does anyone out there know of any packages that are Victor specific (i.e., DAC Easy, is it still available for V9K?), or other packages that will work with XTBIOS.

My experiences are all limited to more expensive packages (RealWorld Payroll 4.0 runs under XTBIOS, and I am preparing to attempt to port the latest version of DataWrite to V9K form IBM Diskettes). There were several packages reviewed in PC Magazine Vol. 6, No. 15, September 15, 1987, but I do not know if any of these will work under XTBIOS.

[1] #93 (of 109) LLOYD LEBLANC, on 15-JAN-88 00:43  
(Size: 667 Bytes)  
Subject: EMULATION

Marty: Thanks for the information on the WA board's segments.

Franz Hirner out here wonders how I can get "mono" emulation with my 832k nominal RAM (actually after loading DOS 2.11 HD I see about 767k -- out of MULTO) -- and load CGINIT -- but not get the CGA emulation. Any ideas? Anyone out there getting CGA emulation with the WA Escort boards? Anyone out there PERIOD?

Marty -- have you seen CGA emulation via IBMBIOS? Is it up to (better than?) the PlusPC CGA screen output? I'd like to have it for some programs where the CGA screen is more "interesting" (a la JAVELIN).

[1] #96 (of 109) LLOYD LEBLANC, on 16-JAN-88 16:10  
(Size: 1027 Bytes)  
Subject: IBMBIOS - MORE

To Marty -- Mainly

The problem with combining IBMBIOS and the WA Escort 1.5 board is that the WA with its 832k RAM limit cuts JUST ENOUGH out of the clock IBMBIOS wants for CGA manipulation -- that it fails. But sufficient space remains (luckily in this case) to get the mono mode (or IBM MDA emulation). I've just finishing testing this with a number of

IBM packages in both the MDA and CGA modes. Lotus and Quattro and others will continue to run OK -- so far -- in the mono mode with the WA set for 896k RAM.

(PS: This with only 128k or 256k on the V9 motherboard. I don't know if using another memory board before the Escort causes another set of problems). Al Bullock out here sez that WA is "working on the drivers".....this reminds me of the response you got when you talked to WA about the DOS variations and the driver fixes needed.

Keep me posted on this one and I'll do the same. I'd like to get that CGA emulation for some applications. Just to ease eyestrain if nothing else.

[1] #114 (of 115) SAM MARTIN, on 24-JAN-88 10:02  
(Size: 2936 Bytes)  
Subject: Emulating the Rich and Famous

Now that, thanks to 128k board from Charlie Vaughn, I have finally achieved a Full House in the world of DOS memory and that has led to putting IBMBIOS to work, with the preliminary results I report below.

Unless you want REAL graphics (i.e. you play a lot of those arcade games), or what passes for real graphics in the Big Blue world, there is no need to install the CGA initiation file, which latches on to 40k of the available RAM and has to be installed early after boot, in the autoexec, before anything else has a chance to use the space. There is an advantage to not reserving the 40k unless you are going to use the graphics, in that you can work with long files in certain programs (such as Edix and Brief).

Annoying problem: IBMBIOS balks at installing itself if you are using the SEARCH.COM enhancement of the D Path command; merely disabling ("search /d") is not enough, you have to get it out of memory. Solution: either "TSR" mark and release set up appropriately, or perhaps some other solution that tells Path you are interested in non-executable files as well as those it recognizes; I will explore the terrain further and report back, if not eaten by cannibals along the way.

Momentarily disturbing: the repeating keys (cursor, backspace, etc.) just trudge one pace at a time. Solution: hold down the RPT key while keeping the cursor key depressed, and all is the way God (or was it Chuck Peddle) intended it to be.

Good news: you can Efont to your heart's content (BEFORE entering the IBM mode) and use any weird set of characters you like. Bad news: Keygen is out, because

the emulator takes over the keyboard. Solution: Use Smartkey to adjust the 265-character set around, depending on the application program (read: word processor) you are using. You can use the latest version of Smartkey under the emulator, and it is well worth getting, with many more features than the one available for the Victor.

More good news: the latest version of Edix works beautifully. I can get up to 560+ k of working memory -- i.e. text in RAM, divided into as many as 20 open files (I never get beyond 6) and viewable in a nicely configured 1-to-4 windows. That's about 128k more than ever available on an IBM 640-maximum set-up. Using both the V9 and the AST 268, with Brooklyn Bridge between, I can work with a full meg of text(s) before my eyes, which I find very useful. BRIEF will probably be similar, when I get around to checking it out.

Incidentally, if you have problems with IBMBIOS, use HXTBIOS instead; it seems to be a little more tolerant, though perhaps a bit less efficient. Again, without the CGA graphics.

Final note: I had problems with DOS 3.1, but they may have been due to my ineptness. Since my system has been on DOS 2.11, I didn't pursue the matter; I don't see much advantage to DOS 3 anyway, at least for my purposes.

[1] #119 (of 128) SAM MARTIN, on 03-FEB-88 21:10  
(Size: 2932 Bytes)  
Subject: IBMBIOS

#### Notes on using IBMBIOS:

1. If you are using Guy Gordon's ramdrive, it should be set to zero ("Ramdrive /s=0"), since it seeks the highest possible memory, which is where IBMBIOS does its thing.

2. The DOS Path replacement "Search.com" (which treats all files with equal respect, even if they are not executable), will interfere EVEN IF DISABLED. It must be totally out of the memory before IBMBIOS will initiate. You can put it back in once you are into the IBM mode, and it works as usual, but once you leave for the Victor mode ("e") you can't return to the IBM mode, because the resident Search.com has latched on to something the initiation needs. Unfortunately the otherwise praiseworthy creator of Search.com failed to include a disinstall function (only switches to disable/reenable). So I have reluctantly given the program up. In its place I have beefed up my Path settings and supplemented that with DPATH, which permits access to data files.

3. Scrnsave does not behave as usual and probably should be ditched. It also creates problems with Brooklyn Bridge, which is apt to stick on the Victor side, so you can't escape back to DOS, without rebooting. I have decided to do my

screensaving "the old fashioned way" -- repeated taps on the dimmer keys.

4. With the current keyboards, the cursor keys (and backspace) lack the repeat attribute. You hold them down and the cursor just sits there. But if you depress the RPT key while holding one of the cursor keys (or the backstop) the rapid skip across the line can be done as nicely as before. The scrolling is another matter; the RPT key only will give you a double-page scroll. Not too satisfactory. I wonder if our keyboard wizards can't improve upon this situation.

5. Speedy (4) can be installed before IBMBIOS and works fine. (Dunno whether it can be installed while already in IBM mode.) We can modcon to any character set we like (very handy!) without worrying about interference. But Keygen is out, unfortunately. Use Smartkey instead. While in the IBM mode the latest version of Smartkey can be initiated and you can switch individual ASCII assignments or put strings/macros in place of one of the characters. (But there is no good way I have found to go directly to the hardware and make new assignments to the keys, as Keygen permits in Victor mode.)

6. The latest IBM version of List works fine.

7. The latest version of EDIX works very nicely and, because of Victor's larger DOS, permits a greater RAM workspace (for text) than on the IBM, with its 640k limitation on DOS. With IBMBIOS on the Victor, you can have close to 570k of text workspace if you have not used the CGAINIT option (before IBMBIOS), 40k or so less, if you have.

I hope these remarks will be helpful to others who want to try the new emulator. I am quite pleased with it, so far.

[1] #128 (of 128) DON WACKER, on 07-FEB-88 23:24  
(Size: 799 Bytes)  
Subject: IBMBIOS & TurboTax

The version 3.12 of IBMBIOS works slick as a weasel with TurboTax, a really excellent income tax program written for the you-know-what. It not only is a really great program, but it's price isn't out of sight -- \$75. Considering what the idiots in Congress did to us this time, Turbotax is one way to attempt to remain somewhat sane while filling in eight million stupid forms.

Also, the Family Research Manager, a Dollarhide genealogy program, runs on the emulator. I don't know how it is for drawing the pedigree charts and such. My luck with graphics with the emulator is less than good; it either hangs before drawing or hangs after and won't return to the calling program. Still, my hat is off to anyone who can make the good 'ol Victor run the other program bare so.

[8] #237 (of 240) CHARLES VAUGHN, on 08-FEB-88  
10:13 (Size: 210 Bytes)  
Subject: @)(\$#(\*#\*)@ V9CSCRN.ARC access.

IBMBIOS and CGA.

I don't understand what all the fuss is about. I have tested the CGINIT CGA emulation from within Quickbasic 4.0 under MSDOS 3.1. No WAL ramdrive installed of course - IBMBIOS won't load if total system memory is less than 896k - not user available memory, but physically present - i.e., the number displayed by the universal boot ROMS. I ran the TORUS and MANDELBROT graphics demos, which both search for a CGA card. The graphics is displayed with shading. I also compiled the MANDELBROT and ran it outside of QB, but still under IBMBIOS.

IBMBIOS and SERIAL PORTS.

In my opinion, this should be the next area targetted before EGA emulation.

[1] #137 (of 152) BOB REUTER, on 14-FEB-88 21:21  
(Size: 750 Bytes)  
Subject: WAL V2 + IBMBIOS

This subhacker doesn't even have any industry substandard software to test IBMBIOS with, but I downloaded it anyway. After reading about all of the problems with the V2 board and getting this thing loaded I tried a little experiment.

I did a before and after CHKDSK and found that if I booted my machine from the 3.1 MS/DOS (copy) disk that I bought without my AUTOEXEC file or any of the other goodies I normally load, like MULTO, that I got back that 64K. Of course, I did not have my ram disk but I did have the full 896K ???

Now I know this solution seems too simple, so maybe some of you hacker types out there can check it out or tell me why it won't work before I really screw something up.

[1] #142 (of 152) LLOYD LEBLANC, on 18-FEB-88 14:42  
(Size: 651 Bytes)  
Subject: QUATTRO ON V9

I can't remember whether I reported this earlier here, but Borland's QUATTRO (new spreadsheet, \$130 discounted) runs very nicely on the V9000 using the IBMBIOS emulator -- and does NOT need color emulation, so you CAN use your WA Escort boards set for 896(832) and use QUATTRO. Despite some carping in the trade press, I

find it an excellent program, sometimes faster than Lotus 2.0 and with nice popup menus, a little easier to use. Good manual, too. So, if you're running a V9000 and want a "state of the art" spreadsheet/database, try QUATTRO. Install for the IBM mono (text) screen and run it with IBMBIOS (without color setup).

[1] #143 (of 152) ED KING, on 19-FEB-88 08:34 (Size: 313 Bytes)  
Subject: Lotus 2.01/V9000

I couldn't help but notice Quattro being used on V9 with an emulator. Does my memory serve me correctly and that Lotus 2.01, unprotected, can be run thru the emulator on V9? What are the particulars. I know 2.0 Lotus on a +PC V9 works, if you do some tricks and don't do graphing. Is the emulator better?

[1] #144 (of 152) SAM MARTIN, on 20-FEB-88 09:25  
(Size: 1475 Bytes)  
Subject: IBMBIOS and Smartkey

A note on using Smartkey under IBMBIOS. It works very well (and I find it essential), but be sure to UNLOAD it before returning to the Victor BIOS. If you don't, it stays in memory, but you can't get back to it (even to unload it) when you return to the IBMBIOS; the crucial "smartkey" just doesn't work. There are probably other resident programs that will cause similar problems, so it would be well to make a habit of unloading such programs when leaving the IBM mode, even temporarily, and then reloading them when reentering it. On the bright side: I find we CAN use MODCON from within the emulator as well as before entering the IBM mode, so it is easy to change character sets at any time. On the negative side: the keyboards have not coded the cursor and scroll keys to repeat when kept held down, the way they normally do on both the Victor and the AT (dunno about the pee-sea). It should be possible to remedy that in a new version of the keyboards, I hope. Meanwhile, though it is a nuisance to have to use both hands, you can hold the RPT key down and it will simulate the cursor repeat (more or less) and give you a bigger swatch of the text when you scroll, in a jerky sort of way.

Does anyone know of a separate program to speed up the cursor (etc.) on the V9? (Something like KEYREPET-.COM, which is apparently hardware specific.) Also, is there a keystroke buffer (type-ahead) program that will work on the V9?

[1] #145 (of 152) LLOYD LEBLANC, on 20-FEB-88 14:35  
(Size: 386 Bytes)  
Subject: LOTUS/IBMBIOS

Lotus 2.0-2.1 runs as does Quattro under the IBMBIOS setup (mono mode is quite all right). I haven't seen the graphics potentials yet -- since I don't use them in the CGA mode -- but I think they both work. I would not say they run "better" under IBMBIOS....but they do run. Screen handling not as nice as in the PlusPC setup. CGA definitely inferior to the PlusPC setup.

[1] #147 (of 152) DON WACKER, on 21-FEB-88 21:07  
Subject: IBMBIOS

I finally found a program IBMBIOS can't handle - a portfolio manager.

Efforts to run work for some output, but input gets the message:

C LEGAL: err translating a char: Error 0

A serious error has taken place -- Restart program.

Does this give a clue?

This is the first of about 8 IBM programs IBMBIOS can't handle.

[1] #149 (of 152) BRAD CHASE, on 23-FEB-88 10:18  
Subject: My bi-monthly upload.... (partial)

.....  
MARTY, LLOYD, JOHN DANKOWYCH, and other W.A./emulator people:

Here's the scoop on IBMBIOS and why it needs 896K. On the Victor, we have 4K of video RAM mapped in at F000:0000 to F000:0FFF. We also have other hardware mapped in in the addresses above E000:0000. It turns out that E000:0000 is also 896K, which is why we have that number as a limit for available system RAM. (By the way MARTY, this is also why you can't shift the WA RAM window above 896K.)

On the PC, IBM (in it's infinite wisdom) decided nobody would need more than 640K of RAM, and reserved everything above that for hardware mapping. The addresses we happen to be particularly interested in at the moment are B000:0000 and B800:0000, which are the base addresses of the video RAM for the monochrome display adapter (MDA) and the color graphics adapter (CGA), respectively. These addresses turn out to be 704K and 736K. The CGA needs 16K of memory, so it uses address range 736K to 752K.

One more thing we need to know is that the Victor operating system (MS-DOS) takes up about 64K of space.

Now, when the system boots, it puts the operating system as high as possible into memory. On a 896K system, this would be at 832K to 896K. On a 768K system this would be at

704K to 768K. (Note that the EGA board uses the space between 640K and 704K too.)

Now, most PC programs write directly to video RAM. When we're trying to run them on the Victor in emulator, they still try and write to where they think the video RAM is. As we can see, on anything less than a 768K system, this will be in the middle of nowhere. On a 768K system this will be in the middle of the operating system. Only on an 896K system is the block of memory the PC uses for video RAM available and free on a Victor.

What I \*think\* is happening with the WA boards is that WA is taking the top 64K for paging its extra memory into the system address space. MS-DOS then takes its chunk, and

then device drivers eat up enough to get down into the video RAM area. I don't have a WA board, so I'm not sure. Try running CHKDSK with a WA installed. If it says less than 752K available, then this is what's happening, for sure. If the number it gives minus the size of CGINIT is less than 736K, I'd say it was too little memory also. LLOYD, this latter might explain why you can get MONO but not CGA.

I haven't looked at IBMBIOS much either, but I can suggest a couple of possibilities to make the whole thing work. (1) For people with 768K of memory, write a "reboot" program that fools the system into thinking they only have 704K of memory when it reloads the operating system and everything. This would hold that video RAM window open for emulator use. Offhand, I doubt this is worth the effort. Let people buy the extra 128K! (2) For people with 896K and WA boards, it might be possible to write a device driver that grabs the necessary RAM range (704K-752K) and holds onto it for the emulator. It might also be possible to modify the WA code to reserve this space.

In either case, you \*would\* lose that RAM for use by programs, but how many people really use it for anything but a RAM drive?

.....

[1] #150 (of 152) BRAD CHASE, on 23-FEB-88 15:03  
Subject: IBMBIOS

SAM and IBMBIOS users: You might want to look into a program called REFEREE for helping you unload your memory-resident IBM-mode programs. It would probably be possible to modify IBMBIOS to automatically load the memory resident stuff you wanted when you run it and \*automatically\* unload it all upon exit. Shouldn't even be that difficult really. It should be possible to modify

IBMBIOS to add repeat to the cursor/scroll keys. I have doubts about speeding up the Victor cursor - certainly don't know of a program to do it beyond SPEEDY, the general screen speed up.

JOHN: It's going to be real tough to emulate the IBM serial port access. These ports use the in and out instructions on the I/O bus. Victor doesn't use that bus at all, but instead memory maps its ports. See my previous message for some ideas on emulating, but it's not going to be easy.

[1] #153 (of 156) CLINT CHAPLIN, on 24-FEB-88 22:29  
Subject: IBMBIOS

It may be possible to emulate the serial port of the IBM using the IO2NMI card that is suggested in the documentation for IBMBIOS, along with appropriate software behind it, but it'll be slow. This is the method that PC+ and VI uses, and they can only go at 1200bps. Need I say more? Actually, some smart person could make a card that has on it the 8250 and the baud rate generator exactly as IBM has, and bring out another serial port connector. With the real hardware behind the IBM serial port, there'll be no limitation on speed.

[1] #160 (of 163) DAVE EPPESTINE, on 06-MAR-88 00:15  
Subject: IBMBIOS

I've been futzing around with the IBMBIOS emulator and it works very well for most things, but has anyone tried to use it with a benchmark thing like the good old flight simulator? (unprotected, of course)... and has anyone done anything more with development of that IO2NMI generator the IBMBIOS documentation talks about building? Can you actually buy a breadboard from Radio Shack or some such that will plug right into a Victor? If so, what's it called or is there a part number for that rascal? What does this IO2NMI generator really give to you? I thought it might be the shading features talked about, but back a few messages ago, I read where someone used IBMBIOS with a Quickbasic 4.0 demo and got shading (no mention of an IO2NMI generator).

It might be of note that I used IBMBIOS (896k Valid Tech, CGINIT installed) with R:base System V configured for an IBMPC-XT with CGA card and got NO SHADING effects even when I brought up a color palate to change foreground/background colors. Have I done something wrong?

I do find the screen handling to be somewhat better than HXTBIOS, but I still prefer good old EMIBM for the R:base

programs, just much more natural action.

Only one thing... does anyone know of a way to get rid of that rotten blinking underline cursor within any emulation? Perhaps a PD cursor change program for the ibm world?

[1] #162 (of 163) MICHAEL WISHNIETSKY, on 06-MAR-88 21:09  
Subject: Ports and Bridges

With many V9000 owners cranking up their systems to 896K and starting to use Ralf Kurschat's IBMBIOS, there's renewed interest in the various methods for porting IBM software over to the Victor environment.

I've edited Ralph Grabowski's earlier NewsDisk article on the subject, added the various messages from The PUB concerning Guy Gordon's Brooklyn Bridge, and included Jack Slates recent BRAVO newsletter article reviewing the Bridge in the file PORT.ARC now in Sierra Hotel (File Section #6 for the FNG's).

The Victor 9000 version of the Brooklyn Bridge, which sells for \$199, is available to VICTOR CIRCLE and PUB members at a special user group price of \$169. See PORT.ARC for Brooklyn Bridge description, reviews, ordering info, etc.

Another mini-service brought to you by your friendly VICTOR CIRCLE crowd in hopes that you won't even notice that there may never be another NewsDisk!

[1] #166 (of 179) SAM MARTIN, on 12-MAR-88 09:23  
Subject: IBMBIOS

When in the IBM mode you can do most DOS things, but don't use the "CLS" command to clear the screen. It seems to be hardware-specific (and is an internal command, so not easily replaced). Has anyone figured out a good way around this?

Another shortcoming of IBMBIOS: Kurchat did not (could not?) include in the coding of the cursor (and backspace, delete, etc.) the REPEAT attribute which is there both for the Victor and the IBM machines (at least the AT, I have not checked the obsoleted itty-bitty pee-sea). Brad Chase, in a response to my calling attention to that, had a message on this board that it should be possible to code the keys for the repeat function. Meanwhile, as I think I mentioned, a stopgap solution is to keep the RPT key depressed while using the cursor keys. Not quite the same effect, and it does tie up both hands, but that is better than repeated hitting of the key to move across the line. The emulator is really quite remarkable, so I am grateful for all it does so well.

\* \* \*



# THE LIBRARY REPORT / John Knobelsdorf

On October 5, 1988 we released DISK099. It is in the Chapter set of disks and available for copying at the monthly meetings, or for ordering thru the mail in the usual

manner. Please use the new order form which is included in this issue (Vol. 6, No. 5) of VictorTalk.

Following is a list of files on DISK099:

```

=====
Program Ext.   Size           Additional Information
Name          (bytes)
=====
DISK099 .TXT    7424   Information about the files on this disk.
PK361  .EXE   119680  08/02/88 version of PKWare's archiving utilities
    
```

NOTE: PK361.EXE was obtained as a "single" file, but it "sees" itself as an archive. It cannot be handled as a normal archive until it is modified by using the file PKARC36.DIF which will be found listed in the sub-directory PK36PAT on this disk. If, however, PK361.EXE is invoked with the command line PK3610<ret>, the file will un-archive itself and provide all of the program and documentation files for version 3.61 of PKARC, PKXARC and PKSFX.

Because of this, and the need for proper correlation and indexing, the un-archived contents of this file are also on this disk, and are listed in the sub-directory named PK361.

```

AFILT180.ARC           Improved AFILTER but needs IBMulator
  AFILTER .DOC         1706
  AFILTER .EXE        14240
BROKER .ARC           Amortization Program/Report gen. Use EMIBM
  L-BROKER.COM        54633
  READ .ME            1152
IBMBIO43              (IBMBIOS.A43) 7/11/88 Rev.to IBMBIOS - Runs WP 5.0
  CGINIT .COM         342
  DELTA .DOC          8377
  IB006 .ASM          29815
  IBMBIOS .COM        27035
  IBMBIOS .DOC        28800
  IBMPC .CHR          8320
  README .1ST        12961
  RGBTEST .COM        4388
MARC520 .ARC          Merge ARCHives (from DEA see ARC DOC) v5.20
  MARC .EXE           13087
PK361 .EXE            Version 3.61 of PKWare's archiving utilities
  APPNOTE .TXT        3695
  MAKESFX .COM         876
  MANJAL .DOC         91137
  ORDER .DOC          9693
    
```

PKPAK .EXE	21720	
PKPAK JR .EXE	21702	
PKUNPAK .EXE	15112	
PKUNPAK J .EXE	15030	
README .DOC	529	
RELEASE .DOC	2310	
SOFTDEV .DOC	1193	
PKARCE16.ARC		Utility to use PKXARC with ARC command set
ARCE .DOC	274	
ARCE .EXE	5745	
RSLDR202.ARC		RSL Ramdrive 2.02 for WAL V2, works with IBMBIOS
DISKID .	451	
RAMDRIVE.COM	3335	
RAMDRIVE.DEV	2812	
RAMDRIVE.DOC	10745	
RAMDRIVE.LOG	2789	
RSL200 .MSG	8429	
RSL202 .MSG	5439	

554976 Bytes in 35 Files. 10240 Bytes free.

-----

During the unarchiving of files and the preparation of re-release versions of various of our disks, Jack Prichard discovered several anomalies which are reported here for your interest:

1. PD 60: TYPESQ.DOC is 384 bytes, not 1142 as shown in DISKID60.TXT.

2. PD 60: ARC.DOC is 10792 bytes, not 10880 as shown in DISKID60.TXT.

3. PD 60: Contains ARC.BAK (10880 bytes), but it is not included in DISKID60.TXT. File is same as ARC.DOC with 4 added unprintable characters; therefore, it was not included in PD 060A.

4. PD 78: Four files (PTOOLDAT.LQR, PTOOLSCR.LQR, PTOOLTIM.LQR and PTOOLWIN.LQR) all give message "\_\_\_\_\_ is not a squeezed file" when processed by NUSQ103.COM, NUSQ110.COM or USQ.EXE from PD 73. (Looking at them with DISKTOOLS shows their beginning is different from those .LQR files that work ok; they all start FA FF, whereas, good files start 76 FF.) The other four PTOOL\*. files were unsqueezed with any of the unsqueeze files noted above. All eight files were successfully unsqueezed by ALUSQ11.COM from PD 73.

5. PD 80: DISKID80.TXT shows the four PC-TOUCH files with hyphen. In the directory, the hyphen was replaced by an underline. (This is confusing.) Corresponding files in

the unarchived PC-TOUCH.ARC use the hyphen; hence, in PD 080A the filenames were kept in the hyphen form.

6. PD 80: FREEWILL.DOC in FREEWILL.ARC fails CRC check (and has a garble in it).

7. PD 85: HIRES.ASM has 6656 bytes; DISKID85.TXT shows 6438 bytes.

8. PD 85: GRAPH.C has 8832 bytes; DISKID85.TXT shows 8340 bytes.

9. PD 86: SWITCH.ARC appears in DISKID86.TXT but it is not on disk.

10. PD 92: CMPTRUSE.RNF is 4992 bytes; DISKID92.TXT shows 4924 bytes.

11. PD 94: WP1087UP.TXT is included, but is omitted from DISKID94.TXT. It was included in DISK094A under PUB (assumed). The file appears truncated (at end of first line under "Thesuras").

We appreciate receiving this kind of feedback. It helps us do a better job of maintaining the V\*G public domain library. We wish that more readers would take time to report their findings. Errors and omissions will be corrected as may be possible.

\* \* \*

CONTRIBUTORS

To facilitate readers contacting authors, we are continuing the policy of providing author's addresses and phone numbers (unless requested otherwise). Here are this issue's contributors:

George Anderson,  
782 Jordan Ave.,  
Los Altos, CA 94022  
(415) 941-3029

Bob Elford,  
4097 - 39th Avenue,  
Oakland, CA 94619  
(415) 531-5755

John Knobelsdorf,  
Route 2, Box 103,  
Bertram, TX 78605  
(512) 355-2084

Dan Mardesich,  
27 Quarry Road,  
San Rafael, CA 94901  
(415) 453-4575

Gratz Roberts

\* \* \*

VictorTalk

Address inquiries to:

BAY AREA VICTOR USERS GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Subscriptions: Carol Hirner

Editor: Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

VictorTalk is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in VictorTalk. Copyright 1988 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

VictorTalk is an independent publication not affiliated in any way with Victor Technologies, Inc. Victor is a registered trademark of Victor Technologies, Inc.

Submit Articles on MSDOS disks in ASCII or Spellbinder format (either single-sided or double-sided, in Victor or IBM format), with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

BAVUG BBS is on line 24 hours a day: (415) 574-7914, Franz Hirner, Sysop.

VICTOR\*GROUP meetings are normally held at 7:30 PM on the first Wednesday of the month at the Holiday Inn, Powell Street, Emeryville, CA (adjacent to Highway I-80).

Yes, I want to join.

Make checks payable to:

Enclosed is a check/money order for \$35.00 (foreign membership, \$40.00 in U.S. dollars) for my 1988 Victor\*Group membership.

VictorTalk

1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Name \_\_\_\_\_  
Title \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_

City \_\_\_\_\_  
State \_\_\_\_\_ ZIP \_\_\_\_\_  
Country \_\_\_\_\_  
Phone \_\_\_\_\_

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date \_\_\_\_\_ Office use only:  
Name \_\_\_\_\_ Order rec'd \_\_\_\_\_  
Company \_\_\_\_\_ Member check \_\_\_\_\_  
Address \_\_\_\_\_ Comments \_\_\_\_\_  
City \_\_\_\_\_ State & Zip \_\_\_\_\_  
Country \_\_\_\_\_ Phone \_\_\_\_\_ Order shipped \_\_\_\_\_

Circle disk numbers ordered (all are single-sided). Un-archived versions are available where "A", "AB", or "ABC" appears below number. If originally released (archived) version is desired, cross out corresponding "A", "AB", or "ABC". (Original releases of all other disks contain no archived files.) Note that pricing below is per "disk number" rather than per "disk" (re-released unarchived "disk numbers" may be comprised of one disk (A), two disks (AB), or three disks (ABC). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No.	Disks	Vol. No.	Disks	Vol. No.	Disks	Vol. No.	Disks
4	2 61-64	5	1 78-81	5	5 87-90	6	3 96-97
4	3 65-71	5	2 82-84	5	6 91-92	6	4 98
4	6 72-77	5	4 85-86	6	2 93-95	6	5 99

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56*	57*	--	59	60
													A					AB	
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
												A					AB	AB	AB
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	—
AB	AB	AB	AB	AB	AB	AB	ABC	ABC	AB	AB	ABC	ABC	AB	ABC	AB	AB			

\* 34 & 35 are a two-disk set; 56 & 57 are also a two-disk set

1988 Victor\*Group member: No \_\_\_ Yes \_\_\_ Date joined/renewed \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, sets of 6, \$12.00 /set ..... \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1988 membership fee \$35 (receive all 1988 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR\*GROUP

1493 BEACH PARK BLVD, #180

FOSTER CITY, CA 94404

BULK RATE  
U.S. POSTAGE  
PAID  
LAFAYETTE, CA 94549  
PERMIT NO. 39

VICTOR TALK  
TIME TO RENEW!  
WARE AND DOCUMENTATION RELEASED TO PUBLIC DOMAIN

TIME TO RENEW!



# VICTOR<sup>®</sup> TALK

Bi-Monthly Magazine for Victor 9000 users

VictorTalk, Vol. 6, No. 6, Nov-Dec 1988

## V9000 SOFTWARE AND DOCUMENTATION RELEASED TO PUBLIC DOMAIN

In response to requests made by Victor user groups and various Victor 9000 users, Victor Technologies, Inc. has made available to the public domain "any software designed for the Victor 9000, associated support files and documentation . . . other than Victor MS-DOS Release 3.1 . . .". It also grants permission to copy certain schematic diagrams and installation instructions, as detailed in their letter reproduced below.

### VICTOR

Telephone (215) 251-5000  
396 Phoenixville Pike  
Malvern, PA 19356 USA  
Fax (215) 251-5049

November 7, 1988

Hugh T. Connelly  
13441 Point Pleasant Drive  
Chantilly, VA 22021

Dear Mr. Connelly,

Victor Technologies, Inc. is pleased to grant your request for permission to write user group newdisk and newsletter articles on how to build your own audio preamp board for the Victor 9000. Permission is also granted to copy or otherwise reproduce both the Victor schematic diagram for the audio preamp board and the installation instructions for the audio input option.

To enable Victor 9000 users to take advantage of all the features available to the Victor 9000, Victor Technologies, Inc. is pleased to grant permission to the Victor User Support Groups, Victor Circle Newdisks, and the remaining Victor-interest Electronic Bulletin Boards to copy, disseminate, and otherwise freely share with any of the remaining Victor 9000 users any software designed for the Victor 9000, associated support files and documentation in their possession, other than Victor MS-DOS Release 3.1, which is the only Victor 9000 software currently sold and supported by Victor Technologies, Inc.

Sincerely,

/s/ Paul Arsenault  
Victor Technologies

As a result of the permission granted, we anticipate adding a considerable number of worthwhile programs to our Public Domain library in the near future. Also, VictorTalk will feature related articles in future issues. Coming next month: Fossil Communications Drive, New Communications Software Available. These coming developments are just one more reason to renew your membership in Victor\*Group (and with it your subscription to VictorTalk). See the facing page.

DO IT NOW.

\* \* \*

### VICTOR 9000 COMPUTERS FOR SALE

MICRO COMPUTER APPLICATIONS, LTD is closing out their stock of Victor 9000 computers, accessories and software for the best offer.

#### HARDWARE:-

- 3 Victor 9000 computer, 128K RAM, dual single-sided floppy drives
- 2 Victor 9000 computer, 256K RAM, dual double-sided floppy drives
- 1 Victor 9000 computer, 256K RAM, dual single-sided floppy drives
- 1 Victor 9000 computer, 256K RAM, 10 Meg hard disk, single double-sided floppy drive
- 1 300/1200 baud Bell 212A/103 Modem
- 1 Interface Cable for Modem
- 1 128K RAM Memory Board
- 1 Audio Tool Kit
- 1 Audio Input System

#### SOFTWARE:-

- 1 Select/Victor Writer, CP/M-86
- 1 Report Manager (Victor Calc), CP/M-86
- 1 Graphics Tool Kit, MS-DOS
- 1 Level II Cobol, CP/M-86
- 1 Wordstar/Mailmerge, MS-DOS
- 1 dBase II, MS-DOS
- 1 Async-86 (CP/M-86 Communications)
- 1 Bisync-86 (CP/M-86)
- 1 Silversoft Accounting Package

Micro Computer Applications, Ltd.  
19 South Center Street,  
Marshalltown, Iowa 50158  
(515) 752-8845

# VICTORTALK INDEX – Volume 6, 1988

A cumulative index for VictorTalk, Volumes 1-5, 1983-1987, was published in VictorTalk, Vol. 6, No. 1, Jan-Feb 1988. An index for the year 1988 follows. It is published in the same format as the earlier index and consists of three tables: Table 1, the chronological listing of each published article with its category or categories; Table 2, a cross-index by author; and Table 3, a cross-index by category.

In each table, articles are referenced by a four-digit number (V/N/P) giving the volume, number, and page (as the 1st, 2nd, and 3rd and 4th digits, respectively). (In keeping with the four-digit format, zero is used for the third digit

with pages 1 through 9.) Although this system gives ambiguous references when two or more articles start on the same page, these ambiguities are easily resolved. For ease of reference, categories are listed alphabetically.

In Table 1, supplementary title-like information is given in square brackets in a few instances. Where no author is shown, the article has either been written by the editor or assembled from (usually unidentified) contributions. In Table 3, the categories and their descriptions are also identically those used in "Public Domain Library Index", VictorTalk, Vol. 5, No. 5, Sep-Oct 1987, page 1 (5501).

TABLE 1. INDEX (Page 1 of 2)

V/N/P	Title / Author	Cat.
6101	VictorTalk Cumulative Index	IND
6112	Prufread	WP:WP
6113	Useful Information for New Victor 9000 Users / Roger Folsom	MIS
6117	Victor Computers for Sale	SAL
6117	Victor Circle Newdisk Distributor Change	NEW
6118	Ciphers for Fun and Profit / Andrew Rudiak	BAS GAM
6119	V9 Floppy to HD Conversion	SAL
6120	Victor 9000 Hardware & Software	SAL
6121	From the Boards	CMM WP:SB
6123	Re: Users Group Update	NEW
6123	Victor 9000 Parts	SAL
6123	Victor 9000 and VI Hardware and Software	SAL
6124	Cleaning the Victor 9000 Screen / R. W. Johnson	TEC
6201	Exploring MS-DOS / George Anderson	DOS
6207	Spreadsheet News / Dan Mardesich	SPR
6208	Character Sets / Jack Prichard	CHR
6214	Benchmark Graphics: A Review / Roger Folsom	GRA
6218	Defeating the Fuzzy Wuzzies / R. W. Johnson	SCR

V/N/P	Title / Author	Cat.
6218	Maxell Disk Problem / R. W. Johnson	MEM
6219	From the Boards	CMM EMU HAR
6219	V9000 Hardware	SAL
6219	Victor Computers for Sale	SAL
6220	The Library Report / John Knobelsdorf	PD
6222	Technically Yours	MIS
6224	Victor*Group Address Change	NEW
6224	Re: Victor Circle Newdisk Distributor Change	NEW
6301	IBMBIOS / Franz Hirner	EMU
6304	Making Music on Your Victor / Andrew Rudiak	BAS MUS
6311	Keyboard Files / Jack Prichard	KB
6313	Disks - DSD0 vs. DSHD / George Anderson	MEM
6314	Exploring MS-DOS, Part 2 / George Anderson	DOS
6322	Victor Computers for Sale	SAL
6323	A RAMdisk That Survives Reset / Issue	MEM
6323	Power Supplies in the V9 / George Anderson	HAR
6324	Back-to-Back Paging with Spellbinder / Andrew Rudiak	WP:SB



TABLE 1. INDEX (Page 2 of 2)

V/N/P	Title / Author	Cat.	V/N/P	Title / Author	Cat.
6326	The Library Report / John Knobelsdorf	PD	6501	New Editorial Board for VictorTalk / Dan Mardesich	NEW
6328	Contributors	IND	6502	The Victor 9000 Power Supply / Gratz Roberts	HAR
6328	From the Boards	-	6505	Exploring MS-DOS, Part 4 / George Anderson	DOS
6401	Exploring MS-DOS, Part 3 / George Anderson	DOS	6512	Errata [Exploring MS-DOS, Part 3]	DOS
6411	How to Receive Newsdisks Direct From The Victor Circle Distributors	NEW	6513	Back to BASIC / Bob Elford	BAS
6412	Enlarging Your Environment String Space / Franz Hirner	DOS LNG:C	6515	Victor 9000 Computer for Sale	SAL
6415	From the Boards	BAS CMM DOS EMU MEM NEW	6515	VictorTalk Ad Policy	NEW
6418	Text Reversing Program / R. W. Johnson	BAS WP	6516	From the Boards	BUS EMU FIN MEM SPR
6418	Computers for Sale	SAL	6522	The Library Report / John Knobelsdorf	PD
6419	Help Formulate VictorTalk Improvements	NEW	6524	Contributors	IND
6419	Pro-Yam Communications Program for the Victor 9000	CMM	6601	V9000 Software and Documentation Released to Public Domain	NEW PD VTI
6419	ADA Compiler	LNG	6601	Victor 9000 Computers for Sale	SAL
6420	Electronic Gremlins / John Knobelsdorf	CMM:VMI	6602	VictorTalk Index, Volume 6, 1988	IND LST
6421	The Library Report / John Knobelsdorf	PD	6607	Issue Strikes Again	SAL WP:WS
6423	Special Notice for IBM Emulator Users	EMU	6608	Exploring MS-DOS, Part 5 / George Anderson	DOS
6423	New User Needing Help	MIS	6617	Victor 9000 Computer for Sale	SAL
6424	Contributors	IND	6618	From the Boards	MEM
			6620	V9000 Vendors	LST
			6623	Computers for Sale	SAL
			6624	Contributors	IND

TABLE 2. CROSS INDEX BY AUTHOR

Author	V/N/P (Vol/No/Pg)
Anderson, George	6201 6313 6314 6323 6401 6505 6608
Elford, Bob	6513
Folsom, Roger	6113 6214
Issue	6323 6607
Johnson, R. W.	6124 6218 6218 6418
Hirner, Franz	6301 6412
Knobelsdorf, John	6220 6326 6420 6421 6522
Mardesich, Dan	6207 6501
Prichard, Jack	6208 6311
Roberts, Gratz	6502
Rudiak, Andrew	6118 6304 6324

TABLE 3. CROSS INDEX BY CATEGORY (Page 1 of 3)

Category	Description	V/N/P (Volume/Number/Page)
ARC	ARChiving utilities and information about Backup and Restore, and file compression and expansion techniques for communications purposes. See also LBR and SQZ.	-
AUT	AUToexec.bat programs	-
BAS	BASic programs (including MS-BASIC, GW-BASIC, VBASICA, etc.) and tutorials on writing BASIC programs.	6118 6304 6415 6418 6513
BAT	BATch files and tips on writing BATch files.	-
BUS	BUSiness and accounting software including accounts receivable, accounts payable, checkbook accounting, etc.	6516
CHR	CHaRacter files and means of displaying and printing different character sets (fonts).	6208
COM	CoMMunications, including transfer protocol, modems, bulletin board systems (example, FIDO) and communication software (examples, Vicmini, Kermit, etc.). Some COM software is further identified:	6121 6219 6415 6419
COM:FDO	FIDO BBS	-
COM:KER	KErmit	-
COM:VMI	ViCMini	6420
COM:XMD	XMDem	-
CPM	CP/M utilities	-
CVS	ConVerSion of programs (IBM to Victor), DOS2 TO DOS3, and programs to convert output (Wordstar to ASCII, ASCII to hex, etc.)	-
D/T	Date/Time. Programs for displaying, modifying, etc. date/time information (both at system level, and within various application programs), keeping diaries, calendars, and time records.	-
DB	DataBase programs. Most DB software is further identified:	-
DB:db2	dBASE II	-
DB:db3	dBASE III	-
DB:FEX	File Express	-
DB:PC3	PC-FILE III	-
DIA	DIAGnostic programs to test and find hardware and software problems.	-
DIR	DIRectory programs for displaying or printing lists of files or otherwise manipulating (searching, finding, modifying, sorting, re-dating, deleting, etc.) files.	-
DOS	Disk Operating Systems such as MS-DOS, PC-DOS, CP/M, XENIX, etc. and shells such as Marti Moleski's MULTO.	6201 6314 6401 6412 6415 6505 6512 6608
DTP	Desk Top Publishing, primarily what has become to be known as desk top publishing but includes information of a general nature related to publishing.	-
EDI	EDItors, programs for line editing (such as EDLIN and similar shareware programs) but less than complete word processing programs (such as Wordstar, Spellbinder, etc.)	-
EDU	EDUcational programs, including typing tutorials, math problems, spelling flash cards, etc.	-

ISSUE STRIKES AGAIN

TABLE 3. CROSS INDEX BY CATEGORY (Page 2 of 3)

Category	Description	V/N/P (Volume/Number/Page)
EMU	EMulators, such as those emulating CP/M in MS-DOS, or emulating an IBM PC on a Victor 9000 computer.	6219 6301 6415 6423 6516
FIL	FILE manipulation programs.	-
FIN	FINancial programs for making business calculations (amortization schedules, present value, future value, interest, etc.), forecasting, planning and budgeting and investment programs (such as Wall Street Weekly).	6516
GAM	GAMES, including card games, board games, action games, narrative type games, ball games, crosswords, etc. (many of which include some graphics).	6118
GEN	GENealogy programs, some use BASIC, some need dBASE II.	-
GRA	GRAphics, programs in which graphics are an important element (such as certain games and financial programs, or programs to facilitate the production of graphics other than Knee Toe Graphics.) All programs and information related to Tony Casano's Knee-Toe Graphics are identified with KTG.	6214
HAR	HARdware, including V9000, +PC, and VI computers and computer system components such as CPU's, keyboards, monitors, disk drives, printers, plotters, the NEC 20 chip, etc.; and fixing hardware.	6219 6323 6502
HLM	HLMor related to computers and computer users.	-
IND	INDEXes, including cross-indexes, tables of contents, disk indexing utilities, etc.	6101 6328 6424 6524 6602 6624
KB	KeyBOARDS, including means of changing the characters or functions associated with the keyboard keys and .KB files containing these keyboard assignments.	6311
KTG	Knee Toe Graphics by Tony Casano.	-
LAN	Local Area Network software and related information.	-
LBR	LiBRary utilities for file compression and expansion.	-
LNG	LaNGuages excluding BASIC (.BAS) (which includes MS-BASIC, GW-BASIC, and VBASICA), whose programs and articles are sufficiently numerous to justify a separate category. Widely used and well represented languages are further identified:	6419
LNG:ASM	ASSeMblY language (.ASM)	-
LNG:C	C language (.C)	6412
LNG:FOR	FORtran (.FOR)	-
LNG:PAS	PAScal (.PAS)	-
LST	LiSTs of user groups, bulletin boards, vendors, software, etc.	6602 6620
MAT	MAThematics, including mathematical and statistical computations (including the use of the 8087 series math chip).	-
MEM	MEMory and off-line storage, including RAM, ROM, floppy disks, hard disks, ram disks, the V9000 memory map, etc.	6218 6313 6323 6415 6516 6618
MIS	MIScellaneous, including BBS chit chat and any subject that does not reasonably fit in any other category.	6113 6222 6423
MUS	MUSic and Sound, including audio output from the computer in any form (beeps, voice, noise effects, as well as music.)	6304

TABLE 3. CROSS INDEX BY CATEGORY (Page 3 of 3)

Category	Description	V/N/P (Volume/Number/Page)
NEW	NEws of a general nature, as opposed to news specific to another category (such as VTI, news specific to Victor Technologies, Inc.)	6117 6123 6224 6224 6411 6415 6419 6501 6515 6601
PAT	PATches and fixes, including information on the use of Disk Tools and DEBUG, etc, for restoring files and patching programs and the patch information itself.	-
PD	Public Domain program information, library catalogs, alphabetical listings of public domain files, etc.	6220 6326 6421 6522 6601
PRN	PRINters, including dot matrix, daisy wheel, ink jet, laser types, etc. and including plotters, as well.	-
QUS	QuesTionnaires.	-
SAL	SALes of both hardware and software, including listing of materials for sale, generally with price and other specifics.	6117 6119 6120 6123 6123 6219 6219 6322 6418 6515 6601 6607 6617 6623
SCR	SCReen utilities, including programs relating to the display of computer output and "screen dumps" (print screen on the IBM PC).	6218
SPR	SPRead sheets, including shareware spreadsheets programs such as ASEASY, version 2.8 (on PD #90). Widely used and well represented spreadsheets are further identified:	6207 6516
SPR:123	Lotus 123	-
SPR:MP	Microsoft's Multiplan	-
SPR:MIH	WordPerfect's Mathplan	-
SPR:VC	VictorCalc	-
SQZ	SQeeZe and unsqueeze utilities for file compression and expansion.	-
TEC	TECHnical notes, Victor's Additional Diskid's, tips and "how-to's" of all sorts that are not better suited for listing under a more specific category.	6124
UTL	UTILity programs of all sorts (examples: reset the computer from the keyboard, make the cursor blink, set and clear the 25th line, make menus for programs, etc.) other than those listed under a more specific category like QM, CVS, DIA, DIR, PRN, SCR, etc. (CP/M utilities are listed under CP/M.)	-
VTI	Victor Technologies, Inc. news, including information on general subjects not better listed under a more specific category.	6601
WP	Word Processing and utilities, including use of application programs such as Wordstar, Word Perfect, Spellbinder, etc. and utilities such as making mailing labels. Widely used and well represented application programs are further identified:	6418
WP:PMA	PMAt	-
WP:SB	SpellBinder	6121 6324
WP:WP	Word Perfect	6112
WP:WS	WordStar	6607

\* \* \*

# ISSUE STRIKES AGAIN

[This recently-received flyer (including its headline) may be of interest to a large number of Victor 9000 users. It is reproduced verbatim, but was retyped to fit the constraints of the VictorTalk page format. - Ed.]

Now for Sirius/V9000, Vicki Portable and VI owners

Make Wordstar run three times faster  
-----

Wordstar by MicroPro is the most popular word-processor ever with many Sirius (and Sirius compatible) users relying on this extremely robust and powerful word-processor every day. However, in the upgrade to more powerful DOS versions one shortcoming of Wordstar has become increasingly apparent - Wordstar is slow.

Recently Issue embarked upon a programming project with just one objective - speed up Wordstar. During the early stages of the project the results were uncertain, but with this primary objective firmly in mind our Dutch and English development team has generated an improved Wordstar which works just like ordinary Wordstar but with speed improvements of 300% and more.

STARPATCH is a 'patch' programme which takes a copy of Wordstar and extensively modifies and adds to it so that it writes directly to the Sirius/Vicki screen to give dramatic improvements in execution speed. Completely new code routines handle initialisation, character input and screen output.

The patcher is available to modify Wordstar versions 3.3 and 3.21 and both the patcher and the finished Wordstar will operate on the Sirius, Vicki Portable or Victor VI (in Sirius mode) under any version of MS-DOS. Using Starpatch couldn't be simpler: copy IPATCHWS onto a disc containing WS.COM, and run it. Starpatch is run once and once only, but it has to be used again after making any further alteration to Wordstar with WINSTALL. Cost of only US\$99.95 includes shipping by air mail and easy to follow instructions are supplied. Please enclose ordinary check in US\$, Post Office Money Order or Bankers' Draft with your order and state whether for Wordstar version 3.3 or 3.21.

SCROLL AT THE SPEED OF SOUND! DELETE LIKE LIGHTNING!

WATCH WORDSTAR WHIZZ!

Other Issue products for the Sirius/V9000, Vicki and VI include:  
-----

- \* The Clock+ Card, a real-time clock with 50 bytes of non-volatile, battery-backed RAM supplied with POP-UP software (Sirius and VI)
- \* A low cost 8087 adapter (Sirius, Vicki and VI)
- \* A high technology 512K RAM Board (Sirius)
- \* An advanced diagnostic ROM Set based on Universal ROMs but with many improved features (Sirius)
- \* A RAMDISC in software which, with Issue ROMs installed, survives reset (Sirius, Vicki and VI)
- \* A Public Domain Disc with games, the Issue Newsletters compiled and a full IBMulator in software (this requires 896K)

Data sheets on these products are available on request.

Issue Postbus 3707 1001 AM Amsterdam Nederland

\* \* \*

# EXPLORING MS-DOS, Part 5 / George Anderson

I said last time that there were four external commands which had not been included in the earlier installments of my DOS article, but which we would take up this time. We would discuss also the discrepancies between the DOS 3.1 disks, the DISKID file for DOS 3.1 and the DOS manual. So, let's get started.

The four commands listed as being in DOS 3.1 in Part 3 of the DOS article were: MODE, PARK, SELECT, and TREE.

These commands are truly in the DOS 3.1 manual, along with complete descriptions of how they work. However, it apparently has been a mistake in the editing of the manual. These commands are valid for the IBM PC Clone versions of MS-DOS, but they are not included in DOS for the Victor 9000.

PARK does have a close look-alike command in HDPARK (an unlisted external command), which performs essentially the same function as PARK (i.e., to park the heads of the hard disk over the inner cylinder or track as a safety precaution to avoid damage to data on the hard disk if anything should go wrong when the heads land upon stopping). MODE, SELECT and TREE do not have similar commands in the V9000 version of DOS.

Eight other commands were included in the listing and described functionally in Part 3, but should be deleted, since they are valid for the IBM PC Clones only. They, also, are described in the DOS manual, but do not exist in the V9000 DOS itself. The eight commands are: COMP, DISKCOMP, GRAFTABL, GRAPHICS, KEYBFR, KEYBGR, KEYBSV, and KEYBUK.

The command RDCPM was incorrectly shown in the listing in Part 3; it was also omitted from the 3.1 manual (although it does appear in the 2.11 manual). It should have been shown as contained in both DOS 2.11 and 3.1, but not in 1.25. RDCPM was described in Part 4.

Finally, there are several commands that appear on the DOS 3.1 disks, but not in the DOS manual. They were neither listed nor described in Part 3 or Part 4 of this series of articles. They are listed below, with a short description of their function:

132C - This command prepares the display for presentation of small letters in 132 columns across the screen.

132OFF - Turns off the 132 column mode.

132ON - Turns on the 132 column mode.

Note: While the three 132\_\_ commands do not appear in the DOS 3.1 manual, they are described in Appendix B.4 of the DOS 1.25 manual and in Appendix D.5 of the DOS 2.11 manual. However, the commands are included in all three versions of DOS.

ANSI - This utility emulates the ANSI escape sequences common to PC-compatible computers.

AUTOSET - This command aids in hard disk configuration. It partitions the disk in various ways to optimize access time for different types of usage. It is not important to the average user.

EXESIZE - This command is used to set the memory size for .EXE programs, and is seldom used by the individual. It would usually be used by a programmer, and then not too often. (The command description does appear in the 2.11 manual).

FF - This command means 'Find Files', and requires the path and the file name at the time it is invoked. It will then find a file of the name given, if one exists. It will continue to look for others of the same name, and display them if they exist. The default operation is to prompt for the next action unless the /L switch is used (for List). (Note: The FF command does not appear in all versions of DOS 3.1.)

GETSCRN - This command is used with BASICA; it allocates memory for graphics in BASICA.

KILLSCRN - This command is also used with BASICA to deallocate the graphics memory allocated with the GETSCRN command when finished with the graphics.

RENDIR - This is a utility to rename directories. It is used in the same way as RENAME is used for changing the names of files.

SCRNMX80 - This is a screen dump command to dump screen images to an Epson MX-80, FX-80, or a look-alike printer:

F10 selects the Graphics mode always, regardless of whether the screen is displaying text or graphics.

F9 is a Text/Graphics mode which will select text mode if text is displayed, and graphics mode if graphics is displayed.

If the screen dump command is invoked a second time, it uninstalls itself.

SCRNM84 - Another screen dump command - for the Okidata Micro84 printer. It operates in the same manner as the previous dump command.

VDIR - This is a volume directory utility. It lists Volume name, Type (e.g., MS-DOS), Capacity of the disk, and Directory size. The Volume name will default to the disk type if no name exists.

Because of the many changes required to correct the table given in Part 3 of this series of articles, the table is repeated below with all known corrections incorporated.

#### TABLE OF MS-DOS EXTERNAL COMMANDS

NOTE: The DOS versions containing the commands listed are shown by an 'X' in the appropriate column; a lower-case 'x' indicates that not all copies of DOS 3.1 contain this command, or that the command is undocumented, or only partially documented, in some DOS 3.1 manuals. DOS Ver. 1 = DOS 1.25; DOS Ver. 2 = DOS 2.11; DOS Ver. 3 = DOS 3.1.

DOS Ver.	1	2	3	DOS Ver.	1	2	3	DOS Ver.	1	2	3
132C	X	X	x	EXESIZE	X	x		PRINT	X	X	
132OFF	X	X	x	FF			x	RDCPM	X	x	
132ON	X	X	x	FGREP	X	X		RECOVER	X	X	
ANSI			x	FILCOM	X	X		RENDIR		X	
ASSIGN		X		FIND	X	X		RESTORE		X	
ATTRIB		X		FORMAT	X	X	X	SCRNM84			x
AUTOSET		x		GETSCRN			x	SCRNMX80			x
BACKUP		X		HDPARK			x	SDCOPY	X		
CHKDSK	X	X	X	JOIN			x	SEARCH	X	X	
COMMAND	X	X		KILLSCRN			x	SHARE		X	
CONCAT	X	X		LABEL			x	SORT	X	X	
CPM	X	x		LINK			X	SUBST		X	
DCOPY	X			LS		X	X	SYS	X	X	
DEBUG		X		MDDCON	X	x		TAIL	X	X	
DISKCOPY	X	X		MORE	X	X		UNIQ		X	
EDLIN	X	X	X	MV	X	X		VDIR		X	
EXE2BIN		X		PORTSET	X	x		WC	X	X	

This ends the external commands. Next we will take up batch files.

#### BATCH FILES

Batch files are ASCII files which group a number of DOS commands for sequential execution when invoked. In this sense, they are very similar to macros, which you may have dealt with in your favorite word processing, spreadsheet or other application program. They are as simple or complex as you care to make them. Many functions may be made simpler to enter, may insure accurate entry or may perform a related group of commands automatically. A rule of thumb is that if you find yourself performing the same key sequence five or more times in a session, it may pay dividends to put the sequence into a batch file and save doing the repetitive steps manually.

If DOS is to recognize a BATCH file as such, it is important to follow the several rules that have been set down to define them. The rules cover:

- \* How to name a BATCH file
- \* What type of file must a BATCH file be
- \* BATCH file limits
- \* How to run and how to stop BATCH files

The naming of BATCH file is simple, but must be done accurately. Each BATCH file must carry the extension .BAT, to distinguish it from .COM, .EXE, .DBF and other file naming conventions. The .BAT extension will insure that DOS recognizes the class of file, and will treat it properly to perform its function when invoked. The first eight characters must follow the same rules as the naming of any other file. Reserved words and symbols cannot be used in the file name unless the spelling is made different than that of the reserved words. Examples of acceptable file names for a BATCH file are GO.BAT and WS.BAT. How each one functions is up to you.

BATCH file names should not be the same as DOS command names, such as DIR.BAT or FORMAT.BAT. If they are the same, DOS does not know whether to pick the DIR command or the DIR.BAT file selection. Actually, DOS has a precedence built-in which selects the DOS command first, then a .BAT command. If they have the same name, the DOS command is selected, and your beautiful .BAT command never even gets a chance!

A BATCH file must be an ASCII file (i.e., without any formatting commands, such as are placed in almost any word processing document). However, this does not mean that you cannot use a word processor to create BATCH files. Quite the contrary. Word processing programs are usually the easiest way to create them, with the proviso that they

be created as an ASCII file or that they can be converted to an ASCII file at the time they are stored on disk. For example, WordPerfect calls it the TEXT IN/OUT mode; WordStar calls it the non-document mode; WordStar 2000 calls it the UNIFORM mode; and Q & A calls it the EXPORT/ASCII mode. Sometimes merely PRINTING the file to disk will give you an ASCII file.

To test if the file is in ASCII form, try: 'TYPE (filename.ext) | more'. The '| more' part of the command line will make the pesky text behave and not scroll off the screen before you have a chance to read it. Make sure that more.com is on the default drive, or that you have specified a path to the drive it is on. If the result on the screen is completely readable, with no esoteric symbols of any sort, it is an ASCII file regardless of the name used to dress it up.

There are other ways to create an ASCII file. By using the DOS command COPY CON, followed by a file name and optional extension, anything typed to the screen and followed by a Z (that's Alt-Z) will produce an ASCII file stored on disk. Edlin, DOS's line editor, will also produce an ASCII file. With edlin or a word processor you can go back and repair mistakes found on previous lines (the edit function), but with copy con mistakes may only be corrected before the Return key is struck at the end of a line.

There are some limitations to batch files. First, only DOS commands that work in the immediate mode at the DOS prompt can be included. There are limited exceptions to this by creating a few small response programs in assembly language, then calling them at the appropriate times from within the batch file. In addition there are some subcommands (which we will take up later) usable in batch commands. Finally one may use variable input parameters. Bottom line is that you will be using mainly immediate DOS commands, and they will look the same in the batch files as they do when they are typed at the DOS prompt.

To run a batch file, simply type in its name (but not its extension - that's not needed) at the DOS prompt. The batch file takes over and does its sequence of pre-programmed commands until it has performed all the commands in the file - if we write them correctly.

To stop a batch file while it is running, type Alt-C. DOS asks if you want to terminate the batch job. Press 'Y' to terminate, or 'N' to continue with the batch file. Now let's try our first batch file.

The degree to which batch files can serve as tools depends

on which version of DOS is used, and the ingenuity which one brings to the task. There are, however, some specific principles which will help the process.

## DOS 1.25 BATCH COMMANDS

In DOS 1.25, the extent of the commands specific to batch files are limited to REM and PAUSE. Even with this limited offering, batch files can do quite a bit.

### REM

The REMark command allows one to insert comments to the batch program, which will not show on the screen. REM is used to give the program a name for identification, perhaps the date of the latest revision, and the name of the author. One could even add an address and phone number - especially if it is to be given to friends, who might need such information if they get stuck in using the program or have a question about it. You may also want to insert comments to yourself on the operation of a particular line of commands. In short, you can add any information (not for display) with the REM command. REM is used in the following format:

#### EXAMPLE 1:

```
rem This is TEMP.BAT, rev. 13, dtd. 4/8/88
a:
copy b:%1 /v
```

Where: The line starting with the REM command gives the name of the .BAT file, the revision number and the date. You may use as many REM lines as you like, but be sure to insert a new REM command on each new REM line. 'A:' sets the default drive to drive A:. 'Copy b:%1 /v' copies the file of your choice on drive B: to drive A:, and verifies it.

Can you modify this batch file to loop through the copy command, using a different file name each time? The goal is to copy a list of files from A: to B:.

### PAUSE

The PAUSE command stops the operation of the batch file to allow time for performing a task before continuing. The format for its use is:

#### EXAMPLE 1:

```
REM prep.bat
type prep.txt
pause
```



Where: 'Prep.txt' asks you to: "Insert a formatted disk in drive A: ", for example. 'Pause' provides a stopping point in the .BAT file program to insert the disk. As part of the PAUSE command, a message is displayed on the screen, saying: "Strike any key to continue . . . ". Note that 'prep.txt' is simply a small ASCII file containing the quoted line, above. It is called by the word 'type' in the batch file.

## DOS 2.XX & 3.1 BATCH COMMANDS

In DOS 2.XX five new commands for batch file use were added. These are carried over into DOS 3.1 without addition of new commands. They are:

ECHO	IF-EXIST
FOR-IN-DO	SHIFT
GOTO	

### ECHO

The ECHO command is an ON-OFF toggle command. When ECHO is ON, it displays the command line or message following it on the line only. If you plan to ECHO a message which is longer than one line, another ECHO command is necessary on each new line needed for the message. ECHO ON is the default condition, meaning if you do not want commands or messages displayed, you must say so, by inserting an ECHO OFF command on the command line preceding the 'not-to-be-displayed' material. It looks like this in use:

#### EXAMPLE 1:

```
echo off
b:
echo HELLO! HOW ARE YOU?
echo ARE WE READY TO WORK, NOW?
echo Put your work disk in drive B:
pause
word
```

Where: The 'ECHO OFF' command says that all lines following will not be displayed, unless they are preceeded with the ECHO command. The next three line are preceeded with the ECHO command, and are therefore displayed. 'Pause' gives the operator time to insert the disk in drive B:, and 'WORD' invokes the program with the file name 'WORD'.

### FOR-IN-DO

This command states that FOR (the variable substituted

here) IN (a set of things to operate on) DO (the command specified here, to the same variable following FOR). The variable following FOR must be in the form of '%variable', and may be named anything EXCEPT the numbers 0 through 9. Those are reserved for the batch dummy variables which are named %0 through %9.

In operation this command reads in sequentially, the members of the set following IN. These are the files to be worked on. The command action is given following DO, and the variable following the command must match the name of the variable following FOR, at the beginning of the command. They must be preceeded by a double percent sign (%%). The set members following IN may be named either by the replaceable batch variables %0-%9, or any other alphanumeric characters, and must be enclosed in parentheses.

The proper format is:

```
FOR %%variable IN (set) DO command %%variable
```

EXAMPLE 1: To create directories for four different groups, Quarters 1 through 4, use:

```
FOR %%A IN (Q1 Q2 Q3 Q4) DO MD %%A
```

Where: %%A is the variable to be operated on. Q1 Q2 Q3 Q4 are the set members to be sequentially substituted into %%A. MD is the command to be performed (make directory), and %%A is the variable on which to perform the command.

### GOTO

The GOTO command is used to place the action of a batch file program at different points in the program, depending on conditions immediately preceding the GOTO command. When used with the IF command, GOTO is placed on the same line as IF; otherwise it is placed on a separate line. The GOTO command shifts program action to the line immediately following a LABEL. LABELs are composed of any eight (or less) alphanumeric characters preceeded by a colon (:). For example, a LABEL might be ':SECURE', and it occurs on its own separate line in the batch file.

EXAMPLE 1: To display one of two messages, based on the content of a keyboard input requires that the program make a 'decision'. Here is an example:

```
echo off
if %1 = private goto hello
echo Not Authorized
```

```

goto end
:HELLO
  echo I've been waiting for you!
:END

```

Where: 'Private' is substituted for '%1', causing the program to 'goto hello', where ':HELLO' is programmed to display the message: "I've been waiting for you! ".

If 'private' had not been typed in and substituted for %1 by the program, the equality requirement (the double equals sign (=) - DOS cannot use a single equals sign, since that is reserved for assignment statements, such as 'device=port.exe') would not have been satisfied, and the program would instead have gone on to the next line in sequence, which would cause the display to read: "Not Authorized ".

So, whether the batch file was invoked with the word 'private' or not would determine the response displayed. Notice that either route through the program ends up at the :END label, and the program action is complete.

#### IF-EXIST

We used the IF command in the example above, but there are further expansions on the command which may also be used. IF may be used either in the positive form or the negative. To use the negative form, add 'NOT' after IF. There are three modes of use for the IF command. They are: 1) an equality, as demonstrated in the preceding example; 2) an 'existence check' (does the specified thing exist or not?); 3) an error level of number 'n' occurs.

In an equality, two alphanumeric strings are compared, both for upper/lower case, and for match. If a match is found, then the command specified is performed. For example:

EXAMPLE 1: IF %1=George goto (label)

The %1 would be typed in at the keyboard, while the 'George' would be stored in the program. For a match to occur, the spelling of George would have to be correct, and the initial 'G' would have to be capitalized.

An 'existence check' states that IF EXISTS (a path and/or a file name), then a command is performed. The 'path/file name' is typed in at the keyboard, and if the program finds it, it will execute the command specified. It might look like this:

EXAMPLE 2: IF EXIST A:TEST.BAT DEL

The program would delete A:TEST.BAT if that file were found.

The ERRORLEVEL version of the IF command states that IF ERRORLEVEL 'n' has occurred, then perform the specified command. An error level occurs if there has been a failure of the program to act as expected, and this error level is reported to DOS, but normally is not displayed. However, it may be recovered and used as criteria for command action through the use of ERRORLEVEL in conjunction with IF. Because the IF command checks for an ERRORLEVEL equal to or greater than the ERRORLEVEL generated by the program exit action, any checks must be done starting with the highest level and progressing downward. Usually this is not a problem, since the most common applications check for a '1' or a '0' only.

EXAMPLE 3: ERRORLEVEL is '0' when a program exits normally. It is a '1' if some abnormal event occurred. We may want the batch file to send a completed job to another program if all was well in the first program, and it exited normally (an ERRORLEVEL of '0'). We might have two batch files which perform label generation with the first file, and label sorting with the second. Here's how we might set up the master batch file to determine if the label generation program terminated normally:

```

echo off
genlabel
if not errorlevel 1 goto sort
goto end
:SORT
  srlabel
:END

```

Where: Genlabel is the program to generate the labels, the errorlevel is not 1 (i.e., it is 0) and the program jumps to the label :SORT, where the action is provided by SRLABEL.

#### SHIFT

The SHIFT command works with batch file dummy parameters, %0-%9. It shifts all dummy parameters one place to the left. When more than ten dummy parameters are used, the program will process only the first ten, unless SHIFT is used. As each dummy parameter is replaced with inputs from the operator, SHIFT then drops that one from the queue, and renumbers all remaining dummy parameters one digit lower - in effect moving all the queued parameters one position to the left, and allowing more than ten parameters to be processed.

EXAMPLE 1: To display several directories, one after another, a batch file might look like this:

```
dir/p %1
:NEXT
shift
if %1==stop goto end
dir/p %1
goto next
:END
```

Where: A .BAT file named DIRS would be invoked as follows:

```
DIRS WP DB CAD STOP
```

The first line of the .BAT file would display the WP directory, moving into what amounts to a looping action in the :NEXT label. The SHIFT command places DB in the %1 position, a check is made on the content of %1, and the dir/p %1 command line displays the DB directory. The next line moves to SHIFT again, and CAD is now placed in %1. The check is made again on the content of %1 and the dir/p %1 command line displays the CAD directory. The next time around the loop, %1 DOES equal STOP, and the execution jumps to :END. The number of directories displayed is not limited to the 10 replaceable parameters when SHIFT is used.

One last point to make. Starting with DOS 2.XX, the colon (: ) at the beginning of a line in a batch file will serve the same purpose as REM; i.e., that line will not be displayed.

That wraps up the batch commands. Now for a little fun.

#### EXAMPLE BATCH FILES

Here are some batch files for you to play with, to use and to enhance. Each one is meant to get you started toward a useful function, but not to make it as elegant or flexible as possible. By all means add further function to these files; and, if you like them, use them to simplify or to make the operation of your computer safer.

#### CK.BAT

This batch file demonstrates a method of saving typing. In this case, it is applied to shortening the character string for the command CHKDSK to CK by creating a file named CK.BAT. It assumes the CHKDSK command is on disk A:. Substitute another drive letter, if yours is different.

The '%1' after CHKDSK stands for the drive letter you want checked. By using the '%1', you may specify any valid drive you choose, and may then change it to another drive letter in order to check another drive. Short as it is, CK.BAT is general purpose, and it will shorten your typing as you invoke it.

```
echo off
cls
a:
chkdsk %1
```

The proper format is:

```
CK B:
```

#### DLETE.BAT

This .BAT file is designed to move a file you want to delete into a separate directory named DLETE, which must have been created before DLETE.BAT is invoked. In this way, you can clear a directory of files you want to get rid of, and still have one last chance to review it before a later, final deletion takes place. The program assumes the directory DLETE is on drive A:, and the file to be deleted (moved to DLETE) is on drive B:. Here is the .BAT file:

```
echo off
a:
cd \dlete
copy %1
del b:%1
```

The program could as easily have had DLETE created within it, but repeated use of the .BAT file would show an error message saying that it could not create the directory, except for the first time it was invoked. This could be confusing to the user. The %1 dummy parameter is replaced by the name of the file to be deleted. That file is first copied to the DLETE directory, and then is really deleted from its original directory, leaving the copy in the DLETE directory until a decision is made on final disposition.

The proper format is:

```
DLETE
```

#### SORTFOR.BAT

This program will sort a list of ASCII data, and print the results to a (default) printer. If you do this sort of thing often, this one can really save time. You can sort for any attribute in the database records:



1. Q and A
2. dBASE III
3. DOS
4. UTILITIES
5. AutoCAD
6. BASICA

Again, the vertical lines are only to count the blank lines in the file. They should not be entered into the batch file. This text file is also called from the SETUP.BAT file above, and needs no format here to invoke it.

#### HELP\_F.BAT

This batch file is a generic one, and all sorts of variations of it may be generated. The .BAT file here, serves only to call the text file following. Here is the .BAT file:

```
echo off
cls
type help_f.txt
```

#### HELP\_F.TXT

This text file is a FORMAT help file, displayed when called by the HELP\_F.BAT file, above. You could type instructions to someone not familiar with your machine, or with a particular task, and make it available to them just by typing 'HELP\_F' or whatever you named the .BAT file. The file might look like this, when displayed on the screen:

#### FORMAT HELP

To format a disk, follow the steps shown below:

1. First get a blank diskette and insert it into drive B:.
2. Type '4mat'.
3. Screen will tell you that all is ready to format drive B:, single sided. You will be told to "Press any key to start".
4. Press any key and the formatting will proceed. When complete, you will be asked, 'Format another? (Y/N): '. Answer with either a 'Y' or an 'N', depending on whether or not you have more disks to format.

END OF HELP FORMAT

Don't take the instructions too literally - they may be different for your machine! You may, however, put in any instructions on any subject, then arrange the .BAT file names to relate to the subject for which the .BAT file calls the .TXT file of the same name.

#### 4MAT.BAT

This file will format on the same disk drive every time. It will help avoid a mistaken format command applied to a default drive you never intended to format. As written, it expects the FORMAT.EXE command on the A: drive, and the disk to be formatted on drive B:. It will return to A: as the default drive when you exit the FORMAT command.

```
echo off
a:
format b:
a:
cls
```

The proper format is:

```
4MAT
```

#### E.BAT

When I was working on these .BAT files, I had them stored on drive B:, but when I wanted to modify them using EDLIN, which was stored on drive A:, to access EDLIN with less typing, I made a .BAT file to call it.

```
echo off
a:edlin
```

The proper format to invoke it is:

```
E
```

#### LOCATE.BAT

This batch file locates a file if it exists on the drive or in the directory you specify. It then tells you it was found, or that it was not found. You might try adding some more commands to the batch file to get it to tell you something more specific than this version does.

```
echo off
cls
if not exist %1 goto NOFIND
echo The file %1 has been located!
goto END
```

```
:NOFIND
```



To CREATE a message for a given date:

1. Type 'CREATE mm-dd'
2. Month & Day must be separated by '-'. Use two digits each for Month & Day.

To VIEW your messages for a given date:

1. Type 'MSG mm-dd'
2. Month & Day must be separated by '-'. Use two digits each for Month & Day.

#### CREATE.BAT

```
echo off
cls
echo
echo
echo
echo
echo
echo Type 'i' to prepare for typing in your message
echo Type in your message
echo Type Alt-C on the SECOND blank line when
done
echo To Exit, type 'e' at the left asterisk
a:edlin %1.msg
```

#### AUTOEXEC.BAT (modified)

```
echo off
prompt $p$g
cls
type remind.txt
```

The message reminder system works like this:

1. Modify AUTOEXEC.BAT to append the items shown to the existing AUTOEXEC file. Do not duplicate lines already in the current file with lines in the new file. The line 'type remind.txt' must be the last one in the AUTOEXEC.BAT file. To experiment, it is SAFEST to use a disk just for practice, and place the AUTOEXEC.BAT file, above, on that disk. Then start by calling this special AUTOEXEC.BAT file from that disk, and not trying to modify a perfectly good file already installed.
2. The new AUTOEXEC.BAT file calls the REMIND.TXT file, which tells you how to

CREATE or to VIEW a file. Follow the instructions to CREATE or VIEW a memo.

3. When you are through VIEWING or CREATING messages, the program automatically takes you to the main program menu.

If you have this under your belt, then you recognize the difference between .BAT ASCII files and .TXT ASCII files. The .BAT files run the commands, and the .TXT files contain information for display. They work together with each other, but more important is the fact that they also work with DOS pipes and other commands so that you can process data, store it in files and manipulate it in many ways.

Hope you have had fun with these. But don't forget that the .BAT file can be a real tool for you, and it is not beyond your reach to create them yourself. A little patience will usually clear up any questions you may have.

So -- your homework is to use .BAT and .TXT files to create a mailing list of ten names, addresses and phone numbers; show how it can be updated, sorted, duplicates removed - in short, do most of the functions a commercial program for maintaining mailing lists would do, though the methods might not be quite so automatic. You might want to start by reviewing the MSG.BAT series, above.

Entries will be put in the public domain, for all to enjoy and use.

\* \* \*

#### VICTOR 9000 COMPUTER FOR SALE

VICTOR 9000 with 128 K RAM,  
dual single-sided floppy drives.  
EPSON RX-100 wide carriage printer with DotsPerfect  
upgrade chip (offers near letter quality),  
plus 14 other print styles.  
FITTED vinyl dust covers  
for computer system and printer.  
Software includes: MS-DOS 1.25, CP/M 86,  
CPA-Client Programmed Accounting 3.62.  
Asking \$500; other offers considered.

FOUCHE & ASSOCIATES  
Sue Ann Fouche, Owner  
1150 Kieffer St.  
Wooster, Ohio 44691  
(216) 264-1289

\* \* \*

# FROM THE BOARDS

[FROM THE BOARDS is assembled from messages appearing on various (generally Victor-specific) bulletin boards, mainly The PUB and BAVUG, by John and Mary Jo Knobelsdorf. Only minor editing is done to correct obvious typographical errors and to eliminate some superfluous material. An effort has been made toward a degree of standardized formatting and the elimination of excess space to the extent practical without degrading legibility. Garbles that cannot be readily rectified are allowed to remain. Believing that considerable worthwhile material is contained in these excerpts, our intention is to include this feature in each issue of VictorTalk, as space will allow. - Ed.]

1] #174 (of 179) JOHN DANKOWYCH, on 18-MAR-88  
15:53 (Size: 8386 Bytes)  
Subject: RSL Ramdrive for WAL V2

R.S.L. Ramdrive 200

I have just uploaded RSLDRIVE.ARC to file section 6. This package allows V9000 users of the ESCORT V2 ramboard, to install a ramdrive while retaining the option of assigning 896 kilobytes of memory to DOS. This option allows the user to run programs, such as Ralf Kurschat's IBMBIOS Ver. 3.12, which require the presence of 896 kilobytes of DOS memory, while retaining access to the ramdrive. Programs running under IBMBIOS Ver. 3.12 will also be able to access the ramdrive. Upon exiting from IBMBIOS, any files saved in the ramdrive will continue to be accessible via V9000 mode programs.

## LET'S CLEAR UP THE CONFUSION # ONE

There seems to be much confusion as to whether or not IBMBIOS will run with CGA emulation when using a WAL V2 ramboard. Some messages on the subject seem to drop the distinction between having a WAL V2 board installed and have WAL ramdrive installed. There is a difference. The first is the hardware ramboard. The second is a piece of software. You don't necessarily have to run the ramdrive software if you have the V2 board installed. On the other hand, the software definitely requires the V2 hardware to be installed.

## LET'S CLEAR UP THE CONFUSION # TWO

Brad Chase explained quite eloquently why IBMBIOS might fail CGA emulation if DOS has less than 896 kb assigned to it. He forgot to point out that IBMBIOS 3.12 won't even install if you have less than 896 DOS memory. This is what

mystifies me. Those who have been having problems with CGA emulation must be running CGinit with an earlier version of the IBM emulator (which installed even if DOS memory was less than 896 kb). The moral of the story is get more memory (up to 896 Kb) and if you already have the WAL V2 board and you want to run IBMBIOS 3.12 with a ramdrive then download RSLDRIVE.ARC.

## RESTRICTIONS

Western Automation places TWO RESTRICTIONS on the users of the ESCORT V2 board. The first pertains to the HARDWARE. The user must not install more than 832 kilobytes of physical memory in the V9000 before installing the ESCORT V2 board. (This restriction may not be necessary with the special PROMS for the Valid board. I haven't investigated this yet. I may look at it when I get my Valid board. If anyone knows, explicitly, what the special proms do, please post an explanation here.)

The second restriction is a restriction on the WAL RAMDRIVE SOFTWARE. Their ramdrive requires the user to not assign more than 832 kilobytes of memory to DOS. It is this second restriction that prevents IBMBIOS 3.12 from running. If you do not install the WAL RAMDRIVE, but you have at least 896 kb of total memory present in your computer, then DOS will see 896 kb at boot time and IBMBIOS can be loaded successfully.

The R.S.L. Ramdrive 200 contained in RSLDRIVE.ARC removes the above second restriction and, thus, allows you to have a ramdrive and an 896 kb DOS (for your IBMBIOS) too.

## PRINCIPLES OF OPERATION

TERMINOLOGY : 64k segment = bank = page = window

## WHY THE WAL WINDOW WAS UNUSABLE BY DOS WITH THE WAL RAMDRIVE INSTALLED.

Owners of the WAL V2 board may have noticed that, although they can't assign the D000H (832k - 896k) segment to MSDOS, they do not lose the use of a segment because of the window. All banks on the board (24 fully populated) are accounted for. This indicates that there must be logic on the board that physically switches in a particular 64k bank of memory to be accessed via segment address D000H. The particular bank that is switched in, is selected by writing a bank select byte to a memory mapped I/O port (EA90:[5]) on the WAL board. This, in



combination with the dip switch settings, causes the WAL board to correctly set the appropriate high-order bits of the memory address on the WAL board and, thus, select the desired bank. The low 16 bits are then decoded when the 8088 executes a write to the D000H bank. A similar process occurs on the read cycle.

The 64k WAL ramdrive window is fixed at segment address D000H. It cannot be moved to another location. The bank switching is only enabled if the hardware logic decodes a 0DH in the high nibble of the address. This makes sense because, without this precondition, any write to any memory location would overwrite the ramdrive data at whatever bank happened to be last selected.

We can understand why WAL imposed the 832k restriction (on DOS memory) when we realize that DOS sits at the top of memory. Any write to a ramdrive bank may involve code which is sitting in the D000H segment. (The driver may be there, as well as the DOS DTA (data transfer area)). The result may be an attempt to execute code which has been switched out of the D000H bank and is inaccessible. Hence, the system hangs.

#### THE R.S.L. SOLUTION

The solution to this problem is to ensure that the critical I/O code and sector buffers reside in a secure area of DOS memory. (Somewhere other than D000H). One way to ensure this integrity is to split the ramdrive into two modules. One a device driver with no IO code to the ramdrive and the other a memory resident module which can be accessed via user interrupt vector 0B0H. The memory resident module contains all of the critical buffers and ramdrive IO code. The device driver is loaded high in memory (partially in the D000H segment when 896k of DOS memory is selected), while the resident module is loaded low in memory, well away from the bank switched WAL window.

The steps involved in writing a block of sectors of data to a particular block of sectors in the ramdrive are as follows:

- 1) DOS issues an output command to the device driver. It passes pointers to its internal buffer (DTA) which contains the sectors of data to be written.
- 2) The device driver is entered and a jump to command 8 (output) results in the execution of interrupt 0B0H.
- 3) The interrupt 0B0H call transfers control to the resident module, which then causes a jump to its command 8. The DTA pointers, the start sector and number of sectors are passed to local (resident module) variables.

4) The WRITE\_BLK routine is called. Before we do any bank switching we first copy a sector of data from the DTA buffer to a local (resident) sector buffer.

5) Now WRITE\_BLK calculates the bank select byte, based on the sector number passed from DOS and the config.sys parameters. This bank select byte is now written to the WAL memory mapped control port (EA90:[5]). If we were to return to the DOS code at the top of memory at this point, the system would hang since DOS is now in a bank which has been switched out.

6) Next, the start sector is used to calculate a byte offset within the bank we will be writing to. Now we simply move 512 bytes from the local (resident) buffer to the selected bank by writing to segment D000H, at the above calculated offset.

7) Before exiting, we restore DOS to the D000H segment by writing the D000H bank select byte to the control port. This byte is simply 0DH - pre\_wal\_mem where pre\_wal\_mem is the total physical memory installed before the WAL V2 board. At this point, we also update source buffer pointers and counters.

8) The WRITE\_BLK routine loops to write the next sector, if any. Up to 65535 sectors may be written per call.

9) When no more sectors remain, the resident module returns control to the device driver and, in turn, to DOS.

A similar locally buffered procedure is applied on ramdrive reads.

#### SPEED

I haven't accurately timed the driver, but it doesn't look noticeably slower than the WAL Ver. 1.14 ramdrive.

#### FUTURE DEVELOPMENTS

Outside of bug corrections, one update is envisioned. This will permit the user to reconfigure the DOS memory and ramdrive size parameters without rebooting. A second associated program will be a Lotus/Intel/Microsoft Expanded Memory Manager. This program will run on the V9000 with a WAL ESCORT V2 ramboard installed. It will provide the LIM EMS Int 067H functions to either V9000 mode programs or to IBM mode programs running under the IBMBIOS. MS WINDOWS uses these interrupts to access extended memory on IBM PCs.

WARNING: I haven't used the ramdrive a lot yet, so there may still be a gremlin in the code. Please report any problems to me asap.

\* \* \*

# V9000 VENDORS

Two years ago VictorTalk published an extensive listing of Victor 9000 support resources. ("V9000 Vendors and Sources / Marti Moleski", VictorTalk, Vol. 4, No. 6, 1986.) The listing was substantially that carried in the then-current Victor Circle NewsDisk. Since that time a number of changes have occurred in the Victor support area. The most important of these has been that Victor Technologies, Inc. has essentially ceased their support of the Victor 9000. ("Victor MS-DOS Release 3.1 is the only Victor 9000 software currently sold and supported by Victor Technologies, Inc.") As these changes in Victor 9000 support have become known to us, they have been reported in VictorTalk. Also, some changes have been reflected in the updated listing included in most releases of Victor Circle NewsDisks.

The listing that follows is based upon the most recently received NewsDisk (#34, dated September 27, 1987) and contains only sources of hardware, software or services that are unique to the Victor 9000. The principal changes made to that listing are in format. The available information is

listed in the following order: company name; name of person(s) to be contacted; company address and telephone/telex number; parenthesized information on products sold, etc.; parenthesized quotes or notes from users; date of last updating of information and source. Listing is alphabetical, by company name (or by last name when no company name is given).

In a few instances, further updating has been done by the editor, based on the latest available information; these changes are identified by: "VictorTalk 12/88". It should be noted that most of the information contained herein has not been updated within the last two years and may well be out of date. Victor\*Group plans to improve this situation by obtaining verification and update information from each resource. Volunteers are sought to assist in this effort; please contact the new editor, George Anderson (see page 24). Anyone having verified update information is also requested to contact the editor. As updates become available, revisions to the list will be published in VictorTalk and coordinated with the Victor Circle NewsDisk.

## A Prompt

Debora A. Levy  
1016 B Street  
San Rafael, CA 94901  
1-415-485-1558  
1-800-227-1617, ext 521  
(Gone out of Victor business?)  
3.3.86

Advent Systems, Inc.  
Dr. Reinhold P. Sell  
88 Nashua Road  
Londonderry, NH 03053  
603-432-2800  
(Advent - ca. 1983-4: \$12,000 system  
with 68000 chip in V9, 16 fonts, full  
display of 8.5x11" page, ultra-hi-res)  
3.3.86

Affiliated Computer Groups  
Joel Schardt  
4605 Cedar Lake Road  
Minneapolis, MN 55416  
800-826-3711 (orders only)  
612-377-7880  
(Sells to members of users'  
groups and to resellers)  
4.11.86

American Design Components  
62 Joseph St.  
Moonachie, NJ 07074  
800-524-0809 / 201-939-2710  
(TM100-4's for \$99.00 each;  
TM55-4's for \$79.50)  
3.4.86 Chase

Ashton-Tate  
10150 West Jefferson Blvd.  
Culver City, CA 90230  
213-204-5570  
(dBase III needs White Crane Fix)  
3.3.86

AutoCAD  
2320 Marinship Way  
Sausalito, CA 94965  
415 332-2344  
Telex 275946  
(AutoCAD ver. 2.17c)  
VC ND 13 Grabowski

AutoCAD support is handled by:  
Sun-Flex, Inc.  
David Render  
20 Pimentel Court  
Novato, CA 94947

415-883-1221  
11.24.86 Parpa TVC

Benner Data Services, Inc.  
Rt. 2, Box 180  
Centreville, MD 21617  
301-758-0381  
10.8.87 Vaughn The PUB

Borland Gameworks  
Martin X. Moleski, SJ  
1225 Otis Street, N.E.  
Washington, D.C. 20017  
(Chess, Bridge, Go-Moku)  
9.24.86

Bullock & Associates  
39522 Benavente Place  
Fremont, CA 94539  
415-651-6183 (7AM - 7 PM)  
(Parts, repairs, hardware, soft-  
ware...)  
11.23.87

Daniel Cerutti  
2, rue Voltaire  
1201-Geneve, Switzerland  
(IBM emulator)  
3.3.86

Computer Products International

740 South Hillview  
Milpitas, CA 95035  
408 945-0100

(Excelsior Hard Disks)  
VC ND 12

Condor Computer Corporation

1490 Eisenhower Place  
Ann Arbor, Michigan 48104

(Relational DBMS v2.11.11)  
12.4.87 Happel

Cosmos Engineering Company

2358 South Robertson Blvd  
Los Angeles, CA 90034  
213-202-0624  
213-558-3905

(Publishes FREE catalogue)

Ed Costa

200 Westland Ave.  
Manchester, NH 03103  
(603)-783-9401

(Plus PC and Network S/W)  
12/88 VictorTalk

DAC Software

4801 Spring Valley Road  
Building 1108  
Dallas, TX 75224

(Affiliated issues Victor format)  
5.21.86 ACG

Datatron Data

Matts Gabrielsson  
Orfa Skoltrand 5  
12612 Stockholm, Sweden  
(0046)8-7445920

(Hqtrs of Victor International)  
3.3.86

Datamension Corporation

615 Academy Drive  
Northbrook, IL 60062  
(312) 564-5060

(Report Manager ver. 2.20)  
4.8.86 Miller

Dayton Software

Charles Dayton  
4530 Vandever Avenue, Suite 1  
San Diego, CA 92120

(Speedy 3, SMP 1.10, etc.)  
6.29.86

Disk Drive Repair

8220 S.W. Nimbus  
Beaverton, OR 97005  
503-626-7104

(HD swaps, replacements; SS/DS  
fixes, etc. Call for pricing and  
brochure)

6.18.87 The Inputer

DOX Electronics, Inc.

81 Scholfield Road  
Rochester, NY 14617  
(716) 266-3930

(Ken Michael of DOX Electronics,  
who used to be Victor's Senior  
Computer Field Engineer in the New  
York region, is now selling the  
Tandon TM 100-4's for \$85.  
Although the price is a bit higher  
than the California junk shops, Ken  
assures me that all drives are tested  
on a V9000 by him and that they  
have passed the Victor diagnostic  
tests. The leads are also long  
enough so that you don't have to play  
amateur technician to install the  
things. The drives come with a 90-  
day, money-back warranty.)

8.26.86 Fuzzy Bear

Zoltan Egeresi

5500 Coast Road  
Santa Cruz, CA 95060  
408-425-4512

(Spare Parts, used Victors)

5.10.86 MicroVision 3/4.86

Fido Software

Tom Jennings  
164 Shipley St.  
San Francisco, CA 94107

(BBS Software, FidoDISK, Viamini)  
12.4.87

Global Technologies

16572 East Louisiana Drive  
Aurora, CO 80017  
303-337-7758

3.3.86

Gray Research Group

Winnipeg, Canada  
204 943-9000  
Telex: 07-57147

(Concurrent DOS)

GSC Enterprises

Mr. Gregory Bird, proprietor  
R.R. #4 (Shep's Subdivision)  
Cambridge, Ontario  
Canada N1R 5S5  
519-623-6232

(Board-level repairs, component  
sales (exchanges))

(Says Jan: "Mr. Bird spent 14 years  
with Victor Canada and handled both  
diagnostics training and board repairs  
on the Victor 9000. He suggests  
that any would-be U.S. walk-in  
customers declare their machines or  
components at U.S. customs prior to  
entering Canada, and check with  
customs re prevailing duty on  
electronic stuff.")

9.24.86 Jan Wells

GTE Telenet Communications Corp.

12490 Sunrise Valley Drive  
Reston, VA 22096  
703 689-6000

(PC Pursuit - Low-cost  
noncommercial BBS access service)  
VC ND 12

Handic Software

400 Patterson Plank Road  
Carlstadt, N.J. 07072  
1 800 524-0484

(Voice data, Word Processing,  
Data)

Heritage Software Research

Technologies  
213 384-4120  
(Smartkey II+?)

High-Tech Electronics

John Ede  
3512 Teton Road  
Philadelphia, PA 19154  
215-632-8840

(Spare Parts)

(Be sure to nail him down on whether  
the part is used/new! 15-20,000 line  
items of spare parts for Victor things.  
Service Manager for Victor for 15  
years or so in Philly.)  
3.3.86

**Hopewell Tech**

Barbara Hopewell & Dan Ruff  
215 West 92nd Street, Apt. 8A  
New York, New York 10025

3.3.86

**Issue**

Postbus 3707 1001 AM  
Amsterdam, Nederland  
(Also PO Box 222, Brighton

BN13BR UK)

(512K RAM board, clock card,  
boot ROMs & RAMDISC providing

non-destructive reset, etc.)

12/88 VictorTalk

**J&M Enterprises (Jim Barlog)**

907 East Liberty Drive  
Wheaton, IL 60187  
312-668-4622

(Repairs)

11.23.87 Affiliated Computer  
Group

**JB Technologies, Inc.**

21011 Itasca St, Unit "F"  
Chatsworth, CA 91311  
(818) 709-6400 or 884-8500

(New HD drives for V9??; DS floppy  
drives; \$59 for DS floppy. Call to  
verify price/address/shipping.)

"Almost certainly the leads on these  
will need to be lengthened, and the  
heads may or may not meet Victor  
spec's (I'm gambling that they will).  
But the order-taker was sufficiently  
familiar with the V9 to specify door  
switches when I failed to do so."

10.25.87 Jan Wells The PUB

**Marvin Krull**

9985 Empire Grade  
Santa Cruz, CA 95060  
(Motherboard RAM upgrades,  
repairs, etc)

11.24.86 PUB

**Lexisoft, Inc.**

Venita Kent-Basham,  
Asst. Mgr - Tech. Support  
Michael Huskey,  
Tech. Support Specialist

Box 1950

Davis, CA 95617

Telex 466 007

(Spellbinder word processor)

12/88 VictorTalk

**Manco Industries**

Tony or Mike Mancusco

P.O. Box 303

Sun Valley, CA 91352

818-767-0495

(Tandon TM 100-4 Disk Drives)

("People wishing to upgrade single  
side floppy systems to double side  
need to obtain TM 100-4s to replace  
the 100-3s in their system. I saw an  
ad this week for TM 100-4s for \$49  
each, about the best price I've seen.  
[Curiously enough, TM 100-3s were  
\$75.] I don't know anything about  
the source." -- Brad Chase)

5.10.86 Chase

**Mark Elfield & Associates, Inc.**

4206 Terrace St.

Oakland, CA 94611

(415) 652-2231

(FastTrax - disk optimizer  
program)

12/88 VictorTalk

**MBA Software Limited**

Roy J. Cannon

Millbank House

Catherine Wheel Road

Brentford TW8 8BW, England

01-847-1686

(CAD Power Plus (Doodle))

3.3.86

**MCI Mail**

Customer Support

Box 1001

1900 M St., N.W.

Washington, D.C. 20036

1 800 MCI-2255

1 800 424-6677

(Electronic Mail)

VC ND 12

**MicroPlus Software**

POB 7325

Santa Rosa, CA 95407

(707) 526-3763

(Accounting Plus)

12/88 VictorTalk

**MicroPro International Corp.**

1299 4th Street, Suite 400

San Rafael, CA 94901

415-499-1200

(WordStar 3.3, MailMerge)

**Microsoft Corporation**

16011 N.E. 36th Way

Box 97017

Redmond, WA 98073-9717

206-882-8088

1.3.87 Lloyd LeBlanc TVC

**MicroStuf**

Ron Roth

Robert Strong

1000 Holcomb Woods Parkway

Suite 440

Roswell, GA 30076

(Crosstalk 16 v. 3.50 6/20/85,

maybe 3.61? 10/25/87)

3.3.86

**MicroVisual Research, Inc.**

58 Beachcroft Road

Wall Heath, Kingswinford

DY6 6HX ENGLAND

(RAM cards)

VC ND 12

**Mid Atlantic Tech**

Dick Morgan, President

Stephen Pawloski, Computer

Dealer Manager

Baltimore, MD

301-243-6593, 6597

(Distributor)

3.3.86

**Minitab**

Vicki Jodon

215 Pond Laboratory

University Park, PA 16802

814-865-1595

(Statistics package)

3.3.86

Orion MicroSystems

5107 SE Henderson St.  
Portland, OR 97206  
503-771-5922

5.10.86

Penny Software

Dale Copps  
741 West End Avenue  
New York, NY 10025  
212 222-4853

(The DC Puzzler (double crostics)  
1.0)  
VC ND 12

Research Technologies, Inc.

3757 Wilshire Boulevard,  
Suite 211  
Los Angeles, CA 90010  
1-800-824-5537  
(in CA 213-384-5430)

(Smartkey)  
12.17.86 Sam Martin The Pub

Scroll Systems

5530 N. Camino Escuela  
Tucson, AZ 85718  
602-299-8087

(PS, SST+)  
1.3.86

Share House

800-843-2823  
(In Illinois, (312)-564-5625

(Report Manager v2.5av \$49.75;  
manual \$49.75 - an upgraded  
version of Victor-Calc by  
Datamation)  
12/88 VictorTalk

SoftCraft, Inc.

222 State Street  
Madison, Wisconsin 53703  
(608) 257-3300

(Fancy Font)  
("I use FF regularly with my Epson  
FX80+. It's a great set of programs  
and the company involved is very  
responsive to questions and such.  
They DO support the Victor 9000  
and should have the latest updates  
available. The price is around \$225  
bucks, but well worth it if you want  
to do any fancy printing.")  
1.???.87 Jan Ewing The Pub

Softquest, Inc.

PO Box 3456  
McLean, VA 22103  
703-281-1621

(The Smart Checkbook \$149)  
3.3.86

Star Software Systems

367 Van Ness Way  
Torrance, CA 90501-1494  
213-533-1190

(Accounting Partner (Full  
Accounting System))  
5.10.86

Sub Rosa, Inc.

1-966 Waverly St.  
Winnipeg, MB  
CANADA R3T 4M5  
204-284-4380

(Max1.1 - compiled database  
manager \$249)  
5.10.86 MVision 3/4.86

Stratos Systems

David Koshinz & Yorgos Hatzis  
6-C Hanger Way  
Watsonville, CA 95076  
(408)-761-2162

(Repairs, services & parts  
supply; both formerly with  
Victor Tech., Inc.)  
12/88 VictorTalk

Technically Yours Enterprises,  
Inc.

Mike Penna  
PO Box 1414  
Aptos, CA 95001  
(408)-688-5201/(408)-761-0765

(V9 hardware, software, &  
computer accessories; formerly  
with Victor Tech., Inc.)  
12/88 VictorTalk

Techno-Logic Concepts, Inc.

One Parklane Boulevard  
Suite 1628 East  
Dearborn, Michigan 48126  
313-271-5700  
1-800-TLC-SERV

(Distributor?; fixed MicroSoft  
Chart)  
10.7.87

Techno-Logic Concepts

(Chicago Office)  
Woodfield Park Office Plaza  
999 Plaza Drive  
Suite 400  
Schaumburg, Illinois 60195

3.3.86

TP Group

PO Box 509  
London N1 1YL  
ENGLAND  
01-833-3501

(Deals for Readers)  
12.1.86 Victor Sirius File,  
October '86

Tyx Corporation

Marketing Office  
1861 Wiehle Avenue  
Suite 260  
Reston, VA 22090  
703-471-0233

(Used to sell UNIX, TEX systems)  
4.8.86

Valid Technologies

Wendy and Mike Cykana  
1401 Pioneer Parkway West  
Suite 107  
Arlington, TX 76013-6251  
817-261-0998

(Great RAM card: 768K with clock  
in one slot)  
6.06.87

Victor Technologies, Inc.

Al Krause, President  
395 Phoenixville Pike  
Malvern, PA 19355

(Victor MS-DOS 3.1 - sole  
software supported)  
12/88 VictorTalk

\* \* \*

COMPUTERS FOR SALE

2 Victor 9000 Computers,  
with 10 Meg HD - \$500 each  
Compatible Centronics Printer - \$300  
Compatible Okidata Printer - \$250

Don Piche  
Box 100  
Rossland, BC, Canada V06140  
(604) 362-7371

\* \* \*

## CONTRIBUTORS

To facilitate readers contacting authors, we continue the policy of providing author's addresses and phone numbers (unless requested otherwise). This issue's contributor is:

George Anderson  
782 Jordan Ave.  
Los Altos, CA 94022  
(415) 941-3029

As has been the past practice, articles not identified by a byline have been authored and/or assembled by the editor.

As was described in the last issue (VictorTalk, Vol. 6, No. 5, Sep-Oct 1988, page 1), the duties of the former editor will be subdivided and shared by an Editorial Board. This transfer of duties will be effective with the publication of this issue, (before the end of 1988). Henceforth, until other participants are identified, the positions of Managing Editor and Executive Editor will be handled by George Anderson, the position of "Paste-up Person" and maintenance of subscriber records, by Andrew Rudiak. As for the past two years, VictorTalk mailings will be handled by Boardman Moore, the Victor\*Group chairman (assisted by others, as necessary). Membership/subscriptions, public domain software mail orders, general correspondence and finances will continue to be handled by Carol Hirner. (See adjoining masthead.)

Assembling new public domain software disks and maintenance of the software catalog will be transferred, as quickly as possible, from John Knobelsdorf (who has handled it so ably for the past several years) to a new Public Domain Librarian. The external handling of public domain software orders will remain unchanged. (See Mail Order Form, page 25.)

\* \* \*

## VictorTalk

**Address inquiries to:**  
BAY AREA VICTOR USERS GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

**Subscriptions:** Carol Hirner

**Editor:** (thru 1988) Jack Prichard  
Box 267  
Pt. Reyes Station, CA 94956  
(415) 663-8062

**Editor:** (1989 on) George Anderson  
782 Jordan Ave.  
Los Altos, CA 94022  
(415) 941-3029

**VictorTalk** is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in **VictorTalk**. Copyright 1988 by Victor\*Group. Contents may be reproduced by permission only. **Permission is only a post card away.**

**VictorTalk** is an independent publication not affiliated in any way with Victor Technologies, Inc. **Victor** is a registered trademark of Victor Technologies, Inc.

**Submit Articles** on MSDOS disks in ASCII or Spellbinder format (either single-sided or double-sided, in Victor or IBM format), with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Doublespace between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

**BAVUG BBS** is on line 24 hours a day: (415) 574-7914, Franz Hirner, Sysop.

**VICTOR\*GROUP meetings** are normally held at 7:30 PM on the first Wednesday of the month at the Holiday Inn, Powell Street, Emeryville, CA (adjacent to Highway I-80).

Yes, I want to join.

Enclosed is a check/money order for \$35.00  
(foreign membership, \$40.00 is U.S. dollars)  
for my 1989 Victor\*Group membership.

Make checks payable to:

## VictorTalk

Name \_\_\_\_\_

City \_\_\_\_\_

Title \_\_\_\_\_

State \_\_\_\_\_ ZIP \_\_\_\_\_

Company \_\_\_\_\_

Country \_\_\_\_\_

Address \_\_\_\_\_

Phone \_\_\_\_\_

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
City _____ State _____ & Zip _____	
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Un-archived versions are available where "A", "AB", or "ABC" appears below number. If originally released (archived) version is desired, cross out corresponding "A", "AB", or "ABC". (Original releases of all other disks contain no archived files.) Note that pricing below is per "disk number" rather than per "disk" (re-released unarchived "disk numbers" may be comprised of one disk (A), two disks (AB), or three disks (ABC). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No. Disks	Vol. No. Disks	Vol. No. Disks	Vol. No. Disks
4 2 61-64	5 1 78-81	5 5 87-90	6 3 96-97
4 3 65-71	5 2 82-84	5 6 91-92	6 4 98
4 6 72-77	5 4 85-86	6 2 93-95	6 5 99

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56*	57*	--	59	60
													A						AB
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
												A					AB	AB	AB
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	—
AB	AB	AB	AB	AB	AB	AB	ABC	ABC	AB	AB	ABC	ABC	AB	ABC	AB	AB			

\* 34 & 35 are a two-disk set; 56 & 57 are also a two-disk set

1988 Victor\*Group member: No  Yes  Date joined/renewed \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, 1988, sets of 6, \$12.00 /set.. \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1989 membership fee \$35 (receive all 1989 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR GROUP

1493 Beach Park Blvd., No. 180

Foster City, CA 94404

BULK RATE  
U.S. POSTAGE  
PAID  
LAFAYETTE, CA 94549  
PERMIT NO. 39

PRINTED MATTER

WHY YERRE LATE (and other lies)

VICTOR TALK

DOWN  
INTRO





---

# NOTICE!

We are sending this issue of VictorTalk to everyone who was on our 1988 mailing list. However, if your mailing label says "FINAL NOTICE," our records show that you have not yet renewed your membership for 1989. This is the last freebie we will send. If you have not renewed yet, please cut or copy the coupon from the back page and mail it in **N O W** with your \$35 (\$40 in U. S. currency for foreign subscriptions). You will get no more notices. Please act immediately.

# VICTOR<sup>®</sup> TALK

Bi-Monthly Magazine for Victor 9000 users

VictorTalk, Vol. 7, No. 1, Jan-Feb 1989

## WHY WE'RE LATE (and other lies)

**G**reetings, salutations, and welcome to VictorTalk's sixth season. Yes, I know: This is Volume 7, No. 1. So why is it only the sixth season? It's like this: VictorTalk began in September of 1983. On January 1, 1984, after only 4 issues, our then-editor decided that his next issue began Volume 2. Thus, the beginning of our second year became Volume 3 instead of Volume 2, and we have been one volume ahead of reality ever since.

However, while we may be one volume ahead, at the moment we are roughly one issue behind. You should have gotten this issue around end of February. Instead you are getting it either in late March or early April, depending. Perhaps you want to know why. (Actually, you are about to find out whether you want to or not.)

First, this issue is the shakeout cruise for a new editorial arrangement: the editor edits, in Los Altos, CA. He mails the partially-digested files to the layout man, in Lafayette, CA. The layout man (Yours truly) mangles the files with his Procrustean word processor until they fit into the usual format. The paste-ups go to our printer liaison (also in Lafayette), who takes the boards to the printer, retrieves the completed magazines, and then entrusts them to the tender mercies of the U. S. Postal Service. These uniformed public servants allegedly deliver the magazines to you, our Gentle Readers, expeditiously. Experience demonstrates otherwise.

Observant readers may have noticed that our boys in blue-gray mediate this process twice. Your Gallant Editor entrusted the floppy disk containing this issue to the posts on February 3.

When your Ruthless Layout Man hadn't received said disk by February 13, he called Ye Olde Ed. Disk #2 was in the mail on February 14. Neither disk had reached Lafayette by February 17, so your Exasperated Formatter got on the horn once again and arranged to meet your Bemused Redactor at a convenient midway point on February 20, at which time a 3rd floppy would

be physically handed from the Party of the First Part to the Party of the Second part without the intervening malign neglect of government functionaries.

This time fate intervened in the form of a dead battery. The 2/20 rendezvous misfired. After further enrichment of Ma Bell, the Fateful Floppy was finally conveyed to your Curmudgeonly Correspondent on 2/21, a mere 18 days late.

It is March 9 as this is written. This issue is just hours away from being handed to the printer. Neither disk from Los Altos has made it to Lafayette yet. We will be entrusting finished VictorTalks to the U. S. Snail very soon now. Here is their track record for our last issue (Vol. 6 No. 6): Promised: Delivery everywhere in the U.S. in five working

days. Actual: 19 days from one end of Lafayette, CA to the other. Some readers report that their issue took as much as 24 days to arrive. We're working overtime to ensure that your next issue is on time. In the meanwhile, anyone starting a movement to impeach the Postmaster General can count on our help.

### IN THIS ISSUE:

#### PAGE

- 2 BUILD YOUR OWN AUDIO BOARD  
Hugh Connelly's schematics and full instructions for making your CODEC talk.
- 9 NEW USER'S CORNER  
Notes about Al Thompson's early experiences
- 10 CHK\_FLOP  
A disk alignment utility program, described by Franz Hirner
- 15 BLANK.BAS  
C. Robert Elford's program for printing blank forms
- 15 THE VICTOR PRIMER  
Tips for the absolute beginner, by Dan Mardesich
- 20 V9FOSSIL  
A communications driver for the 1990s, by Franz Hirner
- 23 BINKLEY  
A distinkley improved terminal program, described by Franz Hirner in the third leg of his hat trick.

## BUILD YOUR OWN AUDIO BOARD/Hugh Connelly

One of the unique features of the Victor 9000 is its capability to record, store, and play back human voice sounds when equipped with the audio input option. This is made possible by an Encoder/Decoder (CODEC) chip whose official name, for all you high-tech types is a 'continuously-variable slope delta modulator' (CVSD). The chip was manufactured by Harris Semiconductor, a division of Harris Corp., as the HC-55516. The audio subsystem, which is made up of this chip, the loudspeaker, and the audio input option described below, provides the following capabilities:

- Generation of voice, tones, bells and other sounds
- Volume can be controlled by the software
- Sounds can be edited and manipulated in digital form by software
- The encoding/decoding (sampling) rate can be controlled by software
- Can be used for digital encryption and transmission of voice data

To take advantage of this built-in capability, all that is needed (and was supplied as an option by Victor Technologies) is an audio preamplifier, microphone, and appropriate software. (In fact, I have it on good authority, that a high output ceramic microphone, producing an AC audio signal of 2 Volts Peak-to-Peak, can substitute for both the standard microphone and the preamplifier board. I have used a dynamic microphone that has less than 0.5 VPP AC output and got a usable albeit low-quality voice file without using an audio preamplifier circuit.)

Victor's Audio Toolkit is the full-blown version of the software that allows editing of voice files and the high-level language interfaces needed to integrate sounds into application programs. The Audio Toolkit has recently been released by Victor Technologies for sharing, copying, etc. by Victor user groups and Victor-interest BBS's.

Accompanying this introduction to the audio input option are schematics (Figure 2.) and instructions on how to build your own audio input option with parts obtained from Radio Shack or any local electronics supply store. There is one exception, and that part is obtainable by mail from the author or from a firm listed in the instructions. The audio preamplifier board is very easy to construct - it contains about 25 parts in addition to the microphone and its jack.

An alternative audio preamp/filter circuit is contained in the data sheet for the Harris CVSD chip, and is shown in Figure 3.

It is even simpler than the circuit Victor used, and has approximately half the components of the Victor version. The Harris circuit is simpler because it does not include the voltage isolation protection circuit contained in the Victor version. However, the values of the components are uncommon and are generally 1% tolerance levels. I have not tried this version and thus cannot vouch for its performance; but for those interested, the Harris data sheet is available by mail. Just write me and I will send it.

Those of you who would like to take advantage of this unique feature of the Victor and WOW your friends and/or kids, should have little difficulty in building the audio input circuit (The first use I made of my audio input option, was to have my daughter load up the voice file I had recorded and play it back on her fifth birthday - you would have loved her reaction as the computer sang happy birthday to her!). Anyone who is interested in following up on this project can give me a call for advice and/or verbal assistance. I would also like feedback on the instructions and on how easy it was for you to build.

This article was originally published in Victor Circle Newsdisk #36 in December 1988. A graphics rendition of the Victor schematic was produced for the NewsDisk using the Victor 9000's EFONT character generation software. The audio features for NewsDisk #36 were produced using the audio toolkit and two different audio input options - the original Victor version on Bert Happel's V9000, and my homemade version on my V9000. It was impossible to tell the difference between the sound produced by the original board and that produced by the homemade board. Those of you who purchased Talking Star-Trek from the Victor\*Group Public Domain libraries or Brad Chase's Victor Sound Laboratory Public Domain disk already know of the voice playback capabilities of the audio input option. Those of you who have some of the earlier NewsDisks with the artful compositions by Andrew Rudiak know of the systems capabilities to produce music. Now that the software to take advantage of the Victor's audio capabilities is freely available why not put on your hobby hat and build the hardware?

I would like to thank the following folks (in alphabetical order) for their assistance in helping me with this project:

Bert Happel  
Franz Hirner  
John Navas  
Gratz Roberts  
Charlie Vaughn

## STEP-BY-STEP INSTRUCTIONS ON BUILDING YOUR OWN AUDIO INPUT OPTION

**WARNING:** I have been as careful as possible in detailing the steps I took in building my board, but since no one has tried to build a board based on the following instructions, I can't be sure that they are error free. If someone is successful in building a board based on these instructions (or, heaven forbid, finds a mistake in the instructions), please let me know so that we can pass the word along in a future article.

All of the materials needed to build the board can be purchased from a combination of Radio Shack, local electronics store, and, in the case of the CLM6200, from Hugh Connelly (while my supply of 4 lasts) or D.F. Associates of Mt. Laurel, NJ (see parts list for full address - minimum is 5 parts at \$4.25 each). Total cost for the board should run under \$25 (assuming only one CLM6200 is purchased). The microphone and batteries run another \$25. (A cheaper mike that works fine and, in fact, eliminates background noise better and doesn't need a battery, is available for \$17.95 from Radio Shack (#33-986A). If you don't have soldering equipment, then add another \$10 to \$20.

Since I knew I would be writing an article on building the board, I used a standard Radio Shack PC board (#276-168), which is both readily available to most readers and is indexed so that locations for components and soldering can be exactly specified. Unfortunately, since I was limited to using this kind of board, about a dozen jumpers are needed and the layout of the components is not very attractive. Functional yes, pretty, no!

If you have no experience soldering electronic parts onto a PC board, get a book or pamphlet from Radio Shack or an electronics store on exactly how to do it. Several components (the transistor, diodes, and opto-isolator) can be destroyed if a heat sink is not used during soldering or if too powerful a soldering pencil (more than 25 watts) is used.

To start out, let's get an understanding of the makeup of the circuit board. First, the side of the board with the copper metal circles (or pads) and lines will be referred to either as the "bottom" side of the board or the "solder" side of the board. Conversely, the side without the metal will be called the "top" of the board or the "component" side of the board. With one exception, all of the components will be inserted on the component side of the board. Similarly, with one exception, all wires will be inserted on the component side of the board.

Now comes the indexing of the hole positions on the board. I used the Radio Shack PC Board #276-168. I recommend highly that you stick to this board unless you have electronics experience and can translate the instructions to fit another board. Either way, you can, with these instructions, build a working audio option board and enjoy its features on your V9000.

Across the top of the board are two rows of holes with numbers assigned from 1 to 15, but not printed on the board. The diagram shows numbers at positions 1, 5, 10, and 15 to help you in orienting the board properly. **CAUTION:** These numbers run from RIGHT to LEFT - the reverse of the normal way we might expect, so be careful in the assembly steps.

Below the top two rows is the main part of the board which has an array of holes 25 across by 30 down. At the top of each of several of the 25 columns of holes are letters (again from right to left) A, B, C, D, E, and F. I will refer to the column directly under a letter as 1, so for example, the holes making up the column under letter A will have A1 for the column coordinate. The column coordinates can take on the following values:

A1..A4, B1..B5, C1..C5, D1..D5, E1..E5, and F1.

From top to bottom at each edge of the board are the row coordinates. They range from 1 to 30 from the top of the board down, with numbers at 1, 5, 10, 15, 20, 25, and 30. To pinpoint a particular hole on the board, all that is required is the row and column coordinates in that order. So, for example 15C4 refers to the hole in the horizontal row next to the number 15 on the board and in the column three columns to the LEFT of the column with a C at the top (remember, the column with the C is C1; and the numbers increase from RIGHT to LEFT).

A rough diagram of the board's coordinate structure is shown in **Figure 1.** (next page).

Next comes locating the rest of the component parts needed to build the audio preamp. board. The list of parts, together with their values is given below:

### COMPONENTS AND VALUES LIST FOR CONSTRUCTION OF VICTOR AUDIO PREAMP. BOARD

#### NOTES:

1. All Resistor values in ohms, 1/4 watt, 5% tolerance
2. All Capacitor values in microfarads, 5% tolerance (except as noted)
3. Capacitors may be ceramic disk or Mylar

Ref. Des., Value	Ref. Des., Value
R1 4.7K	C1 0.033
R2 33K	C2 470Picofarads
R3 1.0Meg	C3 0.1
R4 240K	C4 0.1
R5 39K	C5 1500Picofarads
R6 330K	C6 330Picofarads
R7 68K	C7 0.1 / 20%

Ref. Des., Value      Ref. Des., Value

R8	22K	C8	0.01
R9	68K	NOTE: I used caps with up to 20% tolerance.	
R10	1.5Meg	The board as constructed by Victor used	
R11	1.5Meg	5% tolerance except C7, which was 20%.	
R12	Not Used		
R13	Not Used		
R14	10K		
R15	2K (PCB Potentiometer)		
R16	4.7K		
R17	330K		

**Transistor**  
Q1 2N904

**IC Quad Op Amp**  
U1 LM324

**Opto-Isolator**  
A1 CLM6200

**Diodes**

- CR1 1N4148
- CR2 1N4148

**Miscellaneous:**

- Radio Shack PC Board (276-168)
- Radio Shack Microphone Jack (274-252A)
- 6-pin male connector (J1 on preamp board)
- 6-pin female connector (J1 on preamp board)
- 6-pin female connector (to J17 on CPU board)
- Radio Shack 14-pin DIP Socket (276-199)
- Shielded Audio Cable (2 feet)
- Radio Shack Microphone 600-ohm (33-2011) or
- Radio Shack Microphone 600-ohm (33-986A)
- (Both come with a 6-10 foot cable and jack)

NOTE - Also available from Radio Shack:

- LM324 Quad Op-Amp (276-1711)
- 1N4148 Diodes (276-1122)
- 2N904 Transistor (276-1603)

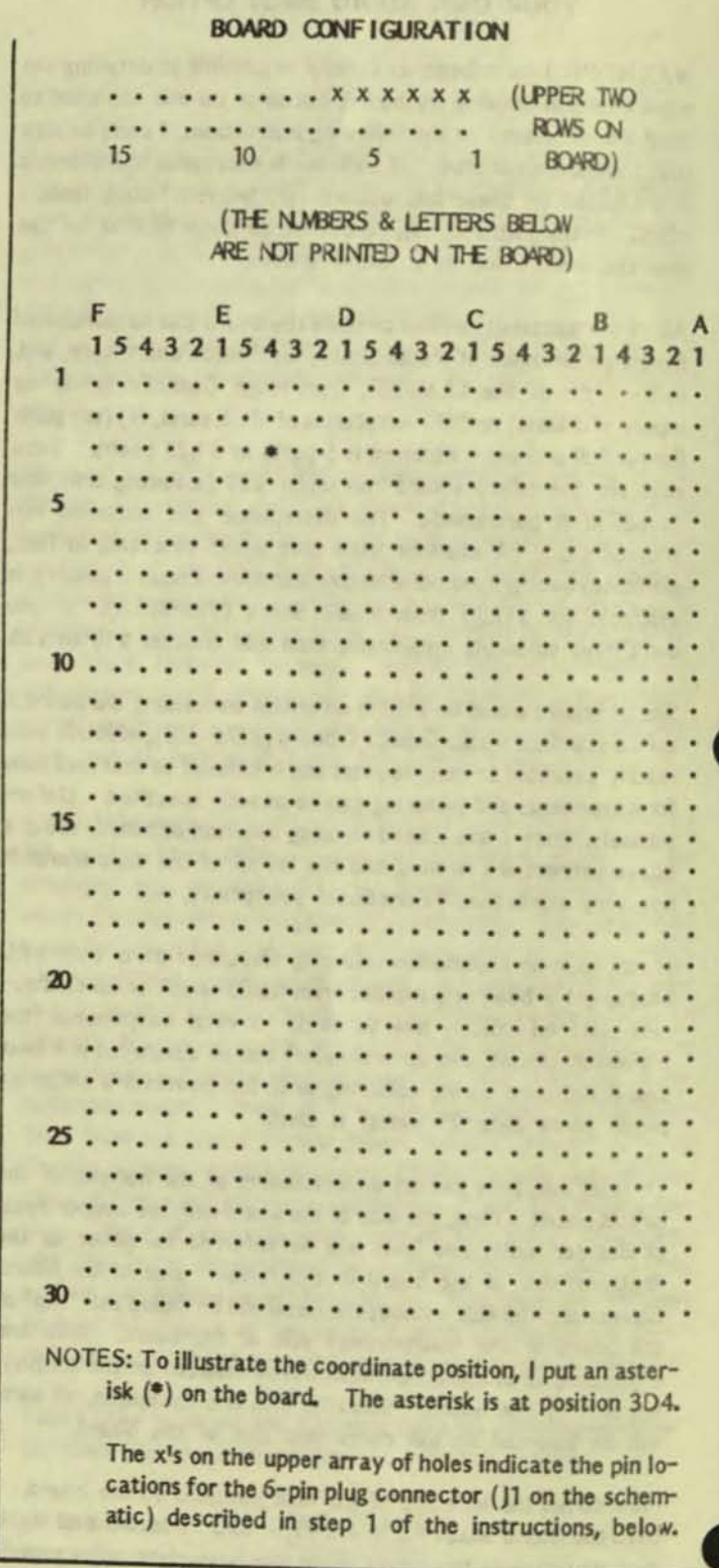
CLM6200 (I have 4 left). Also can be mail-ordered -- min. of 5 parts -- from:  
D. F. Associates, Inc.  
RT. 73, Suite 3  
Mt. Laurel, NJ 08054

Now that you have found the components, a road map, or schematic, of the circuit is needed. See next page, **Figure 2**.

**STEP-BY-STEP INSTALLATION INSTRUCTIONS:**

- 1) Install the 6-pin plug into holes 1 through 6 (shown as x's on the board diagram) of the upper row of the two 15-hole rows at the upper part of the board. The long pins go through the board from the top and should be soldered on the bottom. Keep the solder flat (you only need a minimum of solder to make a good electrical joint - don't add

Figure 1.



more solder than is necessary to wet the copper circle on the board and to form a small fillet to the pin) so that the



6-pin socket case will fit snugly against the bottom of the board over the pins. Connect pins 2 and 3 together by running a strip of solder between them on the bottom of the board or by using a small piece of wire soldered across the top pins. Make sure none of the other pins are connected. Gaps should separate the solder between each of the other pins. (This applies to all of the solder points for all components.) The metal sleeves that will connect the pins to the socket case need to be crimped to fit snugly around the pins. The socket case can then be inserted over the sleeves onto the bottom of the board.

This 6-pin jack hardware connects the preamp board to the J17 jack on the CPU motherboard. The 6-pin connector setup I used was from GC Electronics (Rockford, IL) Cat. Nos. 41-046 and 41-126. They are described as 6 position .100 center. One is a locking polarizing header; the other is called crimp type polarizing locking connectors. I got these at a local electronics supply shop.

- 2) Solder wires from the pins of the 6-pin plug (from the component side of the board) as follows:

To the bottom side of the board:

From Pin 1 to 19C5  
From 19C5 to 22D4

To the component side of the board:

From Pin 3 to 1B1  
From Pin 5 to 12D2  
From Pin 6 to 16A2

Also solder points 1A1 and 1A2 together.

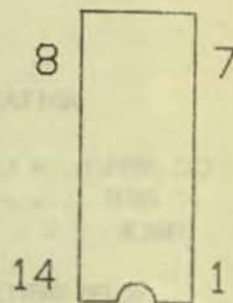
NOTE: Pin 4 has no connection and, as stated above, Pins 2 and 3 are connected to each other.

- 3) Solder wires between the following points along the component side of the board:

From 12C4 to 13C5  
From 7D4 to 16C5  
From 15C4 to 13F1  
From 15B4 to 19C3  
From 18B3 to 21B3  
From 16B4 to 14B3  
From 12A3 to 12B4

- 4) Mount the 14-pin DIP IC socket onto the component side of the board, and solder each pin to the copper circles on the solder side of the board. Do NOT install the IC into the socket until step 37 below. Pin 1 of the IC will be facing away from the 6-pin jack. (It will face toward row 30 rather than toward row 1). The 14 pins of the socket are to be soldered as follows:

Point	Pin#	Pin#	Point
12C3	. 8	7 .	12B5
13C3	. 9	6 .	13B5
14C3	. 10	5 .	14B5
15C3	. 11	4 .	15B5
16C3	. 12	3 .	16B5
17C3	. 13	2 .	17B5
18C3	. 14	1 .	18B5



- 5) Install and solder capacitor C1 between 29B3 and 25B3.
- 6) Install and solder resistor R1 between 25B4 and 17A2.
- 7) Install and solder capacitor C2 between 18A1 and 18A3.
- 8) Install and solder resistor R2 between 17B3 and 17A3; also solder point 17A3 to point 18A3.
- 9) Solder (using heat sink on leads from component side of board) the two short leads of the CLM6200 opto-isolator (A1) as follows:  
  
The lead with the silver mark next to it on the case of the opto-isolator to 20D5. The other lead to 22D5. Solder point 22D5 to point 22D4.
- 10) Temporarily insert resistor R3 at 17B4 and 21B4 (don't solder yet). Take the long leads from the opto-isolator (A1) and extend across the component side of board to either side of R3. Bend each lead to mark the distance. Remove R3 and wrap each long lead from A1 around each lead of R3 twice (wrap close to the resistor body. It doesn't matter which A1 lead is matched to each side of R3). Solder the wrapped leads onto R3 leads using heat sink on A1 leads. Trim excess A1 leads after soldering.
- 11) Install and solder resistor R3 between 17B4 and 21B4.
- 12) Install and solder resistor R4 between 16A3 and 16B3; also solder point 16A3 to point 16A2.
- 13) Install and solder capacitor C3 between 14B4 and 14A1.
- 14) Resistor R5 is the only component to be installed on the solder side of the PC board. Solder resistor R5 on the solder side of the board between 18B4 and 13B4. Trim leads from component side of board so they are as flush as possible and don't contact any other leads on the component side of the board.
- 15) Install and solder resistor R6 from 13B3 to 13A3.
- 16) Install and solder resistor R8 from 7B3 to 11B3; also solder points 12A3 to 13A3 and 11B3 to 12B3.

- 17) Install and solder resistor R7 between 11B4 and 11C4.
- 18) Install and solder resistor R9 between 11C3 and 14C5.
- 19) Install and solder capacitor C5 between 11C5 and 13C4.
- 20) Install and solder capacitor C6 between 14C4 and 14D1.
- 21) Install and solder capacitor C4 between 7B4 and 7C3.
- 22) Install and solder diode CR2 between 7C4 and 1C4 using heat sink. (The cathode is indicated by a black stripe on the diode body, and corresponds to the straight crossbar on the diode symbol on the schematic; the cathode should be toward point 7C4).
- 23) Install and solder diode CR1 between 7C5 and 7D5 using heat sink (the cathode should be toward point 7D5).
- 24) Wrap the leads of capacitor C7 around the leads of resistor R10 twice and solder C7 to R10. Trim the excess C7 leads. Install and solder the combined C7/R10 component between 16C4 and 16F1.
- 25) Wrap the leads of capacitor C8 around the leads of resistor R11 twice and solder C8 to R11. Trim the excess C8 leads. Install and solder the combined C8/R11 component between 17C4 and 17D3.
- 26) Install and solder resistor R17 between 17D5 and 17E5.
- 27) Install and solder resistor R14 between 18C4 and 18D4.
- 28) Install and solder (3 leads) variable resistor R15 as follows:
  - Lead 1 at 20E5
  - Lead 2 at 19E5
  - Lead 3 at 18E5

Connect points 20E5 and 20F1.
- 29) Connect points 18E5 and 17E5.
- 30) Install and solder resistor R16 between 19D5 and 23D5.
- 31) Connect points 19D5 and 20D5 (use heat sink on A1 lead).
- 32) Connect points 23D5 and 22D5 (use heat sink on A1 lead).
- 33) Install and solder transistor Q1 (using heat sink) as follows:
  - Emitter at 17D4
  - Base at 18D5
  - Collector at 19D4
- 34) Connect points 19C3, 19C4, and 19C5.
- 35) Install and solder shielded audio cable leads as follows:
 

Strip cover from outer cable, unwrap braided copper ground and twist into a short lead; carefully remove cover from inner cable exposing about 1/2 inch of bare wire. Be sure that the inner lead remains covered where it starts contact with the braided copper ground lead. Use a drill bit or an awl to slightly enlarge the hole from the component side at point 30C2. Install and solder the braided copper ground lead at 30C2. Install and solder the inner signal lead at 29B5. Run the shielded cable along row 30 on the component side of the board and tie it down to the board with some covered hookup wire at the standoff hole on the bottom left corner of the component side of the board.
- 36) Doublecheck all solder points to make sure they do not unintentionally come in contact with adjacent points. It's good practice to wire brush the solder points to make sure that both sharp solder points are smoothed and that the points are separated as required from adjacent points. Check that all ground columns (A1, B1 and B2, C1 and C2, D1 and D2, E1 and E2, and F1) are separated from adjacent non-ground columns.
- 37) Install the IC chip U1 (LM324) into the socket. Pin 1 end is identified on the chip by a U-shaped cutout on the top of the chip at the center of the end. This end should face row 30 on the component side of the board. (In other words the chip should look like it's installed upside down). Be careful not to bend the chip leads when fitting it into its socket.
- 38) The amount of shielded audio cable you need depends on the type of CPU Motherboard and whether you have an internal hard disk (see preamp installation instructions below). After a test installation of the board, mark where to cut the audio cable to eliminate any excess. On the back of your computer, remove the small black metal plate (accessory plate) and route the cable carefully to the back of the machine and through the hole that the accessory plate covers. Depending on the type of power supply and whether you have an internal hard disk, you may also have to route the cable through a rectangular cutout in the power supply assembly housing. I did on my 10Meg Internal HD/256k machine.
 

The accessory plate will need a hole drilled through it to install the mike jack hardware. The jack hardware is inserted through the hole from the side of the plate that faces the inside of the computer. The washer and nut are installed on the outside of the plate to secure the jack.

Repeat the process for stripping the audio cable described



in step 35. Insert and wrap the signal lead around the terminal on the mike jack that connects to the long tension prong that extends out the back of the jack. Insert and wrap the braided copper ground lead to the other terminal, which connect to the case of the mike jack. Solder the wires to the terminals. These will take a little longer to solder as the large metal terminals will have to be heated.

- 39) If you have access to an external DC power supply that can generate both the +5 to +6 Volt and the +10 to +12 Volt supplies, you can adjust the variable resistor R5 prior to installation. Insert the +12V supply lead into the pin 6 socket hole on the 6-pin socket, and the +6V supply lead into the pin 1 hole. Insert the ground lead from the external power supply into either the pin 2 or pin 3 socket hole. Insert one lead of a 0.1uf capacitor into the pin 5 socket hole. Attach either an oscilloscope or a multimeter to the other end and set it to read AC voltage on a scale low enough to show 2 Volts Peak-to-Peak.

Insert the microphone plug into the jack and while talking into the mike, adjust resistor R15 so that the output signal at the capacitor connected temporarily to pin 5 is 2 Volts AC.

If you don't have access to an external power supply, you'll have to make adjustments by ear once the preamp is installed. If the signal sounds either weak or overdriven (the sound 'breaks up' and becomes distorted), then R15 needs adjustment. Use a small screwdriver to adjust R15 throughout its range by trial and error.

#### HOW TO INSTALL YOUR PREAMP BOARD ON THE CPU MOTHERBOARD

- 1) If you have a 256K motherboard, follow the instructions on removing it that are available on the PUB or elsewhere. On the 128K board, it appears that the board does not need removal. My instructions apply to a 256K Motherboard (without gate arrays) in a 10Meg internal HD V9000. The Victor installation instructions for other setups are summarized below, but I have no first-hand knowledge of their accuracy.
- 2) Locate the 6-pin header jack (J17) on the motherboard. On my board J17 is located at the very front of the motherboard a few inches to the left (when viewed from the front of the machine) of the 2-pin speaker jack. Carefully fit the J1 connector socket of the preamp board onto the pins of J17. The J1 socket end of the preamp board faces the front of the machine (In other words, most of the preamp board will extend out from J17 towards the rear of the machine).
- 3) Route the audio cable along the motherboard to the rear

of the machine. I ran mine between the two innermost expansion card slots. Don't allow the cable to make contact with resistors or capacitors on the motherboard. Reinstall the motherboard and reconnect cables, speaker, and power supply. As noted above in step 38, route the audio cable through the rectangular opening at the bottom left (when viewed from the rear) of the computer chassis backplate. Remove the 1 by 2-inch accessory plate on the rear panel by removing its two screws. Drill a hole through it the size of the threaded portion of the microphone jack and install the jack onto the accessory plate.

- 4) Reinstall the accessory plate and the top and rear covers to the machine. Plug the microphone into the mike jack and you're done!

#### INSTALLATION INSTRUCTIONS PROVIDED BY VICTOR TECHNOLOGIES FOR OTHER CPUS

In installing the preamp board, the principal difference you need to worry about, among the various CPU motherboards made for the V9000, is the location and pin orientation of the audio input header J17. Victor Technologies installation instructions, provided in the Audio Input Option section of the Audio Toolkit II manual, describes three different motherboard configurations.

These are listed below along with the location of J17 (the location description is as viewed from the front of the machine).

CPU Motherboard Description	Location of Audio Input Header J17
128K CPU Board -- no internal hard disk	Left rear next to header J10.
256K CPU Board -- with or w/o gate arrays	Right front (a few inches to the left of 2-pin speaker header)
128K CPU Board -- with internal hard disk	Left rear next to header J10.

**NOTE:** Victor's instructions call for the use of an extension cable: a 6-pin male connector on one end, and a 6-pin female connector on the other. A two-wire shielded audio cable and two standard wires make up of cable. The cable runs between the preamp board connector and J17 on the CPU board. The preamp board is installed on the bottom of the hard drive unit by means of plastic standoff inserts mounted in the holes on the bottom of the hard drive and on the preamp PC board.

It is not clear that this extension cable is necessary. It depends on whether there is enough clearance at the J17 connector for the preamp board to be plugged directly into the connector.

## New User's Corner/Al Thompson

*[Editor's Note: The New User's Corner is a new feature designed to answer the questions YOU want answered. No question is too small, and, although some might be too large, we will go down fighting in trying to answer them. You can contact the New User's Corner by sending a letter to the Editor (address at the back of this issue), or post your question on the BAVUG BBS for an answer in this column. It's meant to be fun as well as to pass on 'Hints and Kinks' for others. So go to it!]*

A year and a half ago I came to California with my V9 hidden in a pile of personal possessions which filled my Cougar to overflowing. The "thing" was already five years old, and had been turned on at least ten or twelve times! Considering the fact that the V9 was my first encounter with computers, I considered myself an expert of sorts with all that experience under my belt. After all, I knew where the switch was to turn it on; and could I ever impress people who had never seen a computer before. I told them it could really do 'neat things' if we only had all the software (read that word somewhere), and they seemed to believe me.

So, with all that expertise going for me, I really had to back it up when I got to California. In one of the manuals I remembered seeing an address for Victor which was just down the road apiece in Scotts Valley. "Simple," I said, "I call them and find out just what I can do with it".

Luckily they saw through my disguise almost immediately, and I talked to someone who put me in touch with Boardman Moore of the Bay Area Victor User's Group. He told me when the next meeting would be held, and I showed up. I didn't know it then, but it was my good fortune to meet Franz Hirner and others who were destined to make my boast of 'expertise' come true. Of course I'm not there yet, but you know the feeling.

My first technical question to Franz was, "If I filled this thing with concrete, would it make a good boat anchor"? "Wait a minute," he said, "what do you want to do with it"? To which I responded "How do I know if it will do anything"?

Thus I admitted the extent of my expertise. With that admission, I began to think seriously that maybe something could actually be accomplished with a computer, and that I would have to stop planting petunias in the open Drive B door. Besides, every time Drive B activated, it sprayed water, dirt and petunia blossoms in all directions.

Today, thanks to the help and encouragement I have gotten from the User's group, I can use the V9 for a number of projects. I am writing this article on the V9, using a pure IBM-compatible word processing program called Q & A, which I run under the IBM emulator written by Ralf Kurschat in Germany and refined for the American keyboard by Franz Hirner. Works like a charm! Thanks guys!

For several months, now, I have even tackled writing a program for commercial tire maintenance, using dBASE III. I hope to use this program in my business as well as sell it to commercial accounts for their own use. That doesn't mean I haven't had trouble working my way through the programming. When I started I knew zip about dBase programming, but you've got to start somewhere. When I look back on what I've learned in a few short months, I am amazed. To think that I would be using a computer and actually doing some of the wonderful things I had dreamed were possible back in the midwest!

THE POINT IS -- there are a lot of new users out there, and YOU have the support of the network of the user's groups and the people in them willing to share with you anything they can contribute to your success in using the V9. After all there is satisfaction in helping others as well as in getting help from others. Join a group. Go to the meetings, and don't be afraid to ask questions. The only dumb question is the one that wasn't asked. In fact, if this article rings a bell with you, write in and ask for the information you need to help you over a rough spot - large or small. We will do our best to answer any question and to provide resources to you to go further. Make this your corner. We'll be here!

# CHK\_FLOP/Franz Hirner

## PURPOSE:

CHK\_FLOP was initially written, with much appreciated help from John Navas, as an aid to recovering data that had been written to a badly out-of-alignment Victor 9000 floppy drive. After several attempts at manually adjusting the drive heads resulted in the infamous DOS "Abort, Retry, Ignore" message or in system lockups, I decided to write a program that would allow the Victor disks to run continuously regardless of disk errors. This would allow the disk read/write head(s) to be manually adjusted more easily until the data to be recovered could be read. Once a drive is removed from its cage, CHK\_FLOP permits proper adjustment as required within a few minutes.

Although originally written to recover data from badly aligned disks, CHK\_FLOP will just as easily permit aligning a disk so that it will read a perfectly aligned or bench mark disk.

While the principles of CHK\_FLOP are not difficult and could easily be applied to any disk drive, CHK\_FLOP has been hard coded for the Victor 9000. The code could easily be modified to run on other computers. Also, the program could be enhanced to write the data being read to a new file automatically.

CHK\_FLOP as written, reads the physical sectors on the disk rather than the logical sectors associated with DOS files. This permits reading sectors 0 through 1223 on side '0' (single-sided drives) or 0 through 2390 on double-sided drives. CHK\_FLOP allows the user to select the beginning sector to read, and the number of sectors to read continuously, from either single- or double-sided floppies. Selecting side '0' (computers almost always begin counting with zero) from the menu will limit access to legal sectors on a single-sided drive, while selecting side '1' allows access to the full range of sectors on a double-sided drive. The beginning sector and/or number of sectors to read are adjusted if the user enters data that would result in trying to read a sector below '0' or above the maximum number of sectors available on a single or double-sided drive, as selected from the menu.

## FLOPPY DISK ALIGNMENT:

CHK\_FLOP has been used successfully not only to adjust a disk to an unknown out-of-alignment floppy,

but also to adjust several disk drives to read data properly written by a disk drive known to be in alignment.

The term alignment is a misnomer. True disk alignment requires an alignment disk that has been very carefully created by the O.E.M. so as to provide both high and low signals that can be read by a scope to insure read/write head alignment to original specifications. CHK\_FLOP merely allows the user to adjust the read/write heads to match a given diskette. True alignment may or may not result depending on the alignment of the drive that originally produced the diskette. The important point is that the drive can be adjusted to read the data. If the data happened to be written from an aligned drive, then the disk being adjusted will be very close to a truly 'aligned' drive. With little care you can match the 'alignment' of any two floppy disks with a fairly high degree of accuracy.

Normally, commercially produced software is distributed on disks written with fairly accurately aligned drives and may be used as a substitute for an 'alignment disk'. Personally, I have set aside 20 disks which have been formatted on a 'very-nearly-new' Victor 9000 for future use as reference disks.

## PROCEDURE FOR ALIGNING FLOPPY DRIVES:

If you have access to a true alignment setup and A/D board then stop here. CHK\_FLOP is not meant to be a substitute for true factory alignment - only a stop gap. The use of an alignment disk and O.E.M. software is the best way to align your drive. Unfortunately, not many of us have access to the real thing and as time goes on it will be more difficult to find a repair facility that can and will align Victor Drives for you.

If you must align your drives, then proceed as follows:

Determine the machine to be used to create a disk that will be used as your master disk or a diskette that you know to be 'in alignment'. If you have selected a machine to make your reference disk, format a NEW, UNUSED diskette on the subject machine. You do want a brand new disk (or one that has been erased with a bulk eraser) to insure that there are no latent magnetic impulses on the disk - neither formatting information nor program data.

## THE CHK\_FLOP PROGRAM

Place the newly formatted floppy in the drive to be 'aligned'. Select either '0' or '1' from the menu depending on the type of drive - single or double-sided. (Note: computers begin counting from '0'. A single-sided drive writes data on side '0' only, and a double-sided drive writes it on both sides '0' and '1'). Select sectors at the outer, middle, and inside positions of the drive. Use the eccentric screw located on the read of the carrier to 'zero' the recording head over the track. When aligning the bottom read/write head (side '0') limit the sectors read to values between 0 and 1223 for sector. Use values between 1224 and 2390 when adjusting the top read/write head (side '1'). Usually it is not necessary to adjust both heads separately unless the disk has been disassembled for repair or the heads have been replaced. Adjust the eccentric screw to move the head position across the complete track. Note that error messages will be displayed when CHK\_FLOP is not properly reading the selected sector but the disk will not stop spinning. DOS keeps track of the sector being read and reports an error if you go too far and position the head over an adjacent track. No error messages are displayed when CHK\_FLOP is successfully reading the sector selected. Note carefully the amount of movement that you can make on the eccentric adjustment screw without moving the head to the point that errors are produced. Split the difference and you should be very close to 'on-the-money'. Repeat on at least outer, middle, and inner sectors until you are satisfied. Differences in the stepper motor from one drive to another may require slight fine tuning of the adjustment between inner, middle, and outer sectors, in order to achieve the best overall adjustment.

Usually, it is not necessary to adjust the top or '1' read/write head separately unless the head has been replaced or removed, but you should check this head if you have double-sided drives. If adjustment is necessary, you must use a trial-and-error position since the top head is not equipped with an eccentric adjustment.

After side '0' has been adjusted to your satisfaction lock the three carrier retainer screws firmly. It's a good idea to format a new disk in the drive you have just aligned and confirm that you can read all sectors on the machine you used to create the master disk before putting the machine back together.

CHK\_FLOP, while perhaps crude by professional standards, does make it easy to match the alignment of one drive to another. Don't use it unless you have to, but when you have to recover data from an 'out-of-alignment' disk, CHK\_FLOP may save the day.

```
/* CHK_FLOP  flop3
   Program to read user defined floppy disk sectors
   for alignment purposes.
```

```
   Sector Size = 512 bytes
   Cluster Size = 4 Sectors
   Clusters per DSDD Disk 594 (Disk Tools)
   Sectors Per SSDD Disk = 1124 (Victor Tech Ref)
   Sectors per DSDD Disk = 2391 (Victor Tech Ref)
```

```
*/
```

```
/*
```

```
A better way to comment out routines is:
```

```
#if 0
```

```
...
```

```
#endif
```

```
or more elegantly:
```

```
#define DEBUG
```

```
.
```

```
#ifndef DEBUG
```

```
...
```

```
#endif
```

```
*/
```

```
#include <assert.h>
```

```
#include <conio.h>
```

```
#include <dos.h>
```

```
#include <malloc.h>
```

```
#include <process.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define DOS_READ (0x25) /* DOS absolute sector(s)
   read */
```

```
/*
```

```
   define clear_scrn() printf("\x1bE")
```

```
*/
```

```
unsigned sect_single_side = 1223;
```

```
unsigned sect_double_side = 2390;
```

```
unsigned available_sectors = 1223;
```

```
unsigned disk_A = 1;
```

```
unsigned disk_B = 2;
```

```
/*
```

```
You should get these values from DOS:
```

1. Use `_dos_getdrive` to get the number of the current drive.
2. Use `_dos_setdrive` to set the drive to the current drive so that you can get the total number of

drives available.

3. Use DOS Function 1C - no special MS C library function so use `intdos()` - to get the information you need about any of the legal drives.

```
*/  
  
unsigned boot_drive; /* program run drive */  
unsigned count = 1; /* the real number of  
sectors to read */  
unsigned disk_side = 0; /* valid choices 0 or 1 */  
unsigned drive; /* current drive - default */  
unsigned number_of_drives;  
unsigned number_of_sectors_to_read = 1; /* user  
specified number to read */  
unsigned sectlen = 512; /* Victor sector size */  
unsigned sector = 0; /* sector number to read */
```

```
void calculate_count(void);  
void clear_scrn(void);  
void  
exercise_disk(unsigned,unsigned,unsigned,unsigned);  
void flush_key(void);  
void get_number_of_sectors_to_read(void);  
void get_sector_to_read(void);  
void locate(char,char);  
void menu(void);  
void set_victor_drive(void);  
void say_status(void);
```

```
void main(int ac, char **av)  
{  
    int choice = 0;  
    set_victor_drive();  
    clear_scrn();  
    menu();  
  
    while(choice != 'Q')  
    {  
        /* Display Menu & Action */  
        locate(20,1);  
        printf(" ");  
        printf("\rEnter Choice... ");  
        choice = toupper( getche() );  
  
        if ( choice == 'A' )  
        {  
            _dos_setdrive(disk A, &number_of_drives);  
            _dos_getdrive (&drive);  
            say_status();  
        }  
  
        if(choice == 'B')  
        {
```

```
            _dos_setdrive(disk B, &number_of_drives);  
            _dos_getdrive (&drive);  
            say_status();  
        }  
  
        if(choice == '0')  
        {  
            disk_side = 0 ;  
            available_sectors = sect_single_side;  
            if (sector > sect_single_side)  
                sector = sector - (sect_double_side -  
sect_single_side);  
            calculate_count();  
            say_status();  
        }  
  
        if(choice == '1')  
        {  
            disk_side = 1;  
            available_sectors = sect_double_side;  
            calculate_count();  
            say_status();  
        }  
  
        if(choice == 'S')  
        {  
            get_sector_to_read();  
            calculate_count();  
            say_status();  
        }  
  
        if(choice == 'N')  
        {  
            get_number_of_sectors_to_read();  
            calculate_count();  
            say_status();  
        }  
  
        if(choice == 'R')  
        {  
            exercise_disk( drive-1, sector, count, sectlen  
);  
        }  
    }  
  
    /* Reset to boot drive before exiting */  
    _dos_setdrive(boot_drive, &number_of_drives);  
    clear_scrn();  
    return;  
}  
  
void clear_scrn(void)
```

```

{
    putchar(0x1b); /* could use define manifest constants
*/
    putchar('E');
    return;
}

```

```

void menu()
{
    locate(1,1);
    printf("\t\t VICTOR 9000 FLOPPY DISK EXERCISER
v1.2
\n\n");
    printf("\t\t Written by Franz Hirner\n");
    printf("\t\t(C) November 6, 1988, All Rights
Reserved\n\n\n");

```

```

    printf("\n\n\t\t (0) - Select Side 0 (Single-Sided
Floppy) \n");
    printf("\t\t (1) - Select Side 1 (Double-Sided
Floppy) \n");
    printf("\t\t (A) - Select Drive A: \n");
    printf("\t\t (B) - Select Drive B: \n");
    printf("\t\t (S) - Start Physical Sector Number
\n");
    printf("\t\t (N) - Enter Number Of Sectors To
Read\n");
    printf("\t\t (R) - Read Sector(s) Continously
\n\n");
    printf("\t\t (Q) - Quit \n\n");
    say_status();
    return;
}

```

```

/* driver for locate.c */
/* locate.c function is called with a char value
between*/
/* 1 and 25 for line, and 1 and 80 for col. */

```

```

void locate(line,col)
char line,col;
{
    if ((line < 1) || (line > 24) || (col<1) || (col >
80))
        printf ("Value out of range\n");
    else
        /*
printf ("%c%c%c%c",ESC,parameter,line+31,col+31);
*/

printf ("\x1bY%c%c",line+31,col+31);
return;
}

```

```

}

void say_status()
{
    locate(6,1);
    printf("\r Test Drive = %c Sector = %d Side
= %d Number Sectors To Read = %d ",
'A' + drive - 1, sector,
disk_side,number_of_sectors_to_read);
    locate(20,1);
    return;
}

```

```

void set_victor_drive()
{
    /* current default drive*/
    _dos_getdrive (&drive);

    /* Now reset drive and get the number of drives */
    _dos_setdrive(drive, &number_of_drives);

    boot_drive = drive; /* save boot drive for exit */

    drive = disk_B; /* force default to B */
return;
}

```

```

/*
** Get text for DOS error code
*/
char * /* pointer to error message text */
errmsgtxt(
    unsigned char errcod /* DOS error code */
)
{
    static char *errmsg[] = {
        "Write-protect error",
        "Unknown unit",
        "Drive not ready",
        "Unknown command",
        "Data error",
        "Bad request structure length",
        "Seek error",
        "Unknown media type",
        "Sector not found",
        "Printer out of paper",
        "Write fault",
        "Read fault",
        "General failure"
    };
    return (errcod < sizeof(errmsg) / sizeof(*errmsg))
?
    errmsg[errcod] : "Unknown error" ;
}

```

```

/*
** Flush keyboard
*/
void
flush_key(void)
{
    while(kbhit())
        getch();
}

/*
** Exercise a disk drive with DOS absolute sector(s)
reads
*/
void
exercise_disk(
    unsigned drive, /* drive number (0 : A, 1 : B,
    ...) */
    unsigned sector, /* relative sector number */
    unsigned count, /* number of sectors to read */
    unsigned seclen /* length in bytes of a sector */
)
{
    union REGS ir, or;
    struct SREGS sreg;
    void far *buffer = malloc(count * seclen); /*
    allocate buffer */

    assert(buffer);

    ir.h.al = (char)drive; /* setup registers */
    sreg.ds = FP_SEG(buffer);
    ir.x.bx = FP_OFF(buffer);
    ir.x.cx = count;
    ir.x.dx = sector;
    locate(20,1);
    printf("
");
    printf("\rPress any key to terminate...\r\n");

    while(!kbhit()) { /* until keypress */
        int36x(DOS_READ, &ir, &or, &sreg); /* read */
        if (or.x.cflag) /* if error, display */
            /* message */
            cprintf("\r%-30s\r", or.x.cflag ?
errmsgtxt(or.h.al) : ".....");
        else
            cprintf("\r
\r");
    }

    flush_key(); /* toss keypress(es) */
}

```

```

        free((void*)buffer); /* free read buffer */
    }

void get_sector_to_read()
{
    /* get a sector to read from the user */
    locate(20,1);
    printf("
");
    printf("\rEnter Beginning Sector.... ");
    scanf("%u",&sector);

    /* make sure the sector is within the bounds of a
the floppy */

    if ( (disk_side == 0) && ( sector >=
sect_single_side ) )
        sector = sect_single_side;

    if ( ( disk_side == 1 ) && ( sector >=
sect_double_side ) )
        sector = sect_double_side;

    return;
}

void get_number_of_sectors_to_read()
{
    locate(20,1);
    printf("
");
    printf("\rEnter Number Of Sectors To Read.... ");
    scanf("%u",&number_of_sectors_to_read);

    if (number_of_sectors_to_read > 100)
        number_of_sectors_to_read = 100; /* 128*512 =
64K DOS LIMIT */
    return;
}

void calculate_count() /* real beginning sector to
read */
{
    count = number_of_sectors_to_read;

    /* don't exceed the capacity of the drive */

    if (sector + count - 1 > available_sectors)
        count = available_sectors - sector + 1;

    number_of_sectors_to_read = count;
    return;
}

```

# BLANK.BAS/C. Robert Elford

At times there's a need for a blank form upon which to record data or to make tabulations. When the proper ruling is not readily available, drawing up a suitable form with ruler and pen is, to say the least, time consuming. In such a situation it is convenient to call up BLANK2.BAS, tell the computer what column width and what line spacing is required, and in less time than it takes to tell it, to have the form printed out and ready to go.

That is the function BLANK2.BAS. A few brief notes to explain its operation:

a. The maximum width of the form is limited to 80 spaces. The program will print a form of any width up to that 80-space limit.

b. The form can consist of up to 40 columns, and the width of the columns may each be any value, as long as the 80-space maximum is not exceeded. The vertical ruled line counts as one of the spaces of a column, e.g., a 19-space column would require 20 spaces. It is out of this requirement that the limit of 40 columns grows; 40 1-space columns plus the 1-space vertical line for each would fill the 80 space maximum.

c. Line spacing is flexible, but note that a single-spaced form would require an entry of "0", since that entry really indicates how many 'extra' lines of vertical ruling are inserted

between the ruled horizontal lines.

d. The number of lines desired is up to you; if you want to, you can exceed the normal 11-inch paper length. At the end of the print-out there is a form feed that should take you down to the end of the 11-inch page on which you are printing.

e. You are permitted to input the number of copies of the form required. The program will repeat itself until that number has been reached and will then stop.

Try it. I think you will find it a convenient program to keep on your utility disk, and it requires very little storage space.

```
10 'BLANK2.BAS prints a blank form with column widths and
```

```
20 'and line spacing adjusted to your specifications
```

```
30 '---VBASICA---1-4-89 / ---C. Robert Elford---
```

```
40 CLEAR : DIM J(50)
```

```
50 DEFINT I-L:DEFSTR S:ICOUNT=0
```

```
60 INPUT "Number of copies of the form";INBR
```

```
70 INPUT "Width of form, in spaces (Maximum 80)";J
```

```
80 PRINT "Enter width, in spaces, of each column desired, limited to 40 columns"
```

```
90 PRINT "and the width of the form. (Column separator line counts as one space.)"
```

```
100 PRINT "That means a 20-space column should be entered as <19>. Follow each"
```

```
110 PRINT "entry with a <CR>. The last entry should be a <-9> to terminate this loop."
```

```
120 FOR I=1 TO 40
```

```
130 INPUT "Column width in spaces:";J(I)
```

```
140 IF J(I)=-9 THEN 170
```

```
150 NEXT I
```

```
160 INPUT "Line spacing (0=single;1=double spaced; etc.)";K
```

```
170 INPUT "Number of form lines required on form";L
```

```
180 TITLE$ = STRING$(J,95)
```

```
190 LPRINT TITLE$:LPRINT:LPRINT TITLE$:LPRINT '**Title Lines
```

```
200 S2 = CHR$(124):I2=0
```

```
210 FOR I1=1 TO L
```

```
220 IF K=0 THEN 310
```

```
230 FOR I=1 TO 40 '**Print cols.
```

```
240 IF J(I)=-9 THEN 280
```

```
250 S1=STRING$(J(I),32)
```

```
260 LPRINT S1;S2;
```

```
270 NEXT I
```

```
280 LPRINT
```

```
290 I2=I2+1
```

```
300 IF I2=K THEN 420 ELSE 230
```

```
310 FOR I=1 TO 40 '**Print lines
```

```
320 IF J(I)=-9 THEN 360
```

```
330 S1=STRING$(J(I),95)
```

```
340 LPRINT S1;S2;
```

```
350 NEXT I
```

```
360 LPRINT
```

```
370 NEXT I1
```

```
380 LPRINT CHR$(12)
```

```
390 ICOUNT=ICOUNT+1
```

```
400 IF ICOUNT < INBR THEN 190
```

```
410 END
```

```
420 I2=0
```

```
430 GOTO 310
```



# The Victor Primer/Dan Mardesich

*Note: The Victor Primer is a new, ongoing series intended for people who are new to computing and have obtained a Victor 9000 with little or no documentation. We will try to define new terms as they are presented and offer resources for further study or problem solving. Your comments and suggestions are welcome.*

## INTRODUCTION

So, you have been presented (or confronted) with a computer that no one has ever heard of. "A Victor? You mean Vector?" "No, Victor Nine Thousand." "Oh. Is it IBM compatible?" "Well, yes and no..."

At this point you may wonder what you got yourself into.

Don't despair. Help is available and you have a really fine machine. The automotive analogy that comes to mind is the Packard: not state-of-the-art today, but substantial, well-built, comfortable, and elegant. And parts to repair your Victor are probably easier to come by than those for a Packard.

This series will try to guide you through your first encounters with your Victor, to help you to feel more comfortable with it, to begin to learn to use it, and to discover what a wonderful tool it is. It will also point you in the direction of resources for support and further study.

Let me say first-off that I am not a programmer, DOS expert, or even a hacker. I think I have some basic understanding about some of the technical aspects of computing, but I won't make many bets based on that knowledge. However, I have solved some problems with and about my Victor and have been using it in my business since 1983. I have learned by reading manuals, books and magazines, by availing myself of that wonderful phenomenon of the PC revolution called the Users' Group, and by thinking, "I wonder if it could do this" and then proceeding to try it. Most important of these has been Victor\*Group in the San Francisco Bay Area, a terrific bunch of people who have put together demonstrations and classes and who have shared their experience and expertise in helping me and others to understand what was going on, how to do something in a more efficient way, or have suggested trying a new piece of public domain software. It is in that spirit I offer to share what I have learned to those coming to the computer from a similar non-technical background.

The first thing to realize about your computer is that it is very tough and you don't have to worry about doing something that will destroy it. You might wreck something if you pushed it off the table, but it will take quite a bit of physical abuse before it dies. You certainly cannot wreck it by pushing the wrong button. You can lose data (and a lot of it in some cases) but for the most part you have to go out of your way to do that. So relax and let's take a look at your machine.

## THE PHYSICAL COMPONENTS

The Victor 9000 consists of three components: the keyboard, the monitor, and a box with a black face plate.

### KEYBOARD

The KEYBOARD comes in three basic versions. The original one which has eight or ten large keys across the top (depends on the particular model), and a later, slimmer one with ten large keys, and a place on the keyboard to rest your hands. The large keys are called FUNCTION KEYS and are numbered from 1 through 7 plus "HELP" on the old keyboard and 1 through 10 on the alternate older keyboard and on the newer one. Function keys are keys which are utilized by a software author (even you and I, sometimes) to issue a command or series of commands to accomplish certain tasks peculiar to computers. The function each one performs is more or less unique to each software package.

The main section of the keyboard under the first seven function keys is your basic QWERTY typewriter keyboard. To the right (below the blank plate on the old keyboard and below function keys 9 and 10 on the new keyboard) is the numerical keypad with the various arithmetical operators, the excellent and useful double zero and a key marked "ENTER" which acts (in most programs) as the "RETURN" key does. Between these two sets and in a row at the far left are some small function keys. More about these later.

Notice that some keys have symbols on the inclined front surfaces as well as on the top. There are more of these on the new keyboards than on the old one, but the characters produced are the same on both, for the same location on the keyboard. For example, the 2 key has "2" and "@" on the top on both keyboards. On the new keyboard, there is "<" on the front face. This is be-

cause computers have an extra key labeled "ALT" which selects an ALternate symbol going to the computer. It works in a manner similar to the "SHIFT" key, but produces the symbols shown on the front of certain keys. This special key is referred to in software manuals as the "ALT" or "CONTROL" key. Software done specifically for the V9000 will usually use "ALT", but programs which have been designed for another computer (Big Blue, for instance) will often call it "CONTROL", since some other machines have included two keys ("ALT" and "CONTROL") instead of just the one, as on the Victor. In any case, you should use the "ALT" key on the V9000 (in combination with others specified) whenever either "ALT" or "CONTROL" is called for.

Just as you can produce a 2 or an @ on a typewriter depending upon whether or not you first hold down the shift key, you may produce a 2 or < depending upon whether or not you hold down the "ALT" key while pressing the 2 key. The symbol for the "CONTROL" key is " ^ ". Thus, ^ d in the literature means "hold down the ALT key (on the V9000) and press d". Be sure to respect the upper or lower case specified in the instructions, or the instruction to the computer may give you something you didn't expect. If you have the old keyboard, you may find it useful to print out the top row of characters along with their "SHIFT" value and "ALT" value and tape it to the top of the keyboard or to the computer. [Victor\*Group still has a few ALT-faced keycaps and F9 and F10 keys plus switches available. Sets are \$5 for members, \$10 for non-members. Specify the exact keys you need when you send your order to Carol Hirner. - Ed.]

The Victor keyboard was designed primarily as a word-processing keyboard and most of the functions performed by the small function keys relate to that environment depending upon how the manufacturer has enabled them. Looking at the keys below Function Key 8 (HELP on the old keyboard) you see "INS" and "DEL". "INS" usually toggles (switches back and forth) between "INSERT" (add letters to a line) and "TYPEOVER" (type over the letters on the screen). "DEL" usually delete the character that the cursor is on. Their action is dependent on the software and how it uses these keys. The various programs will tell you what the keys will do in each particular instance. (To this day I do not know what a "VTAB" is.) ["VTAB" stands for Vertical Tab. It is sometimes used to jump vertically from one form field to another several lines down. -Ed.]

The "ALT" value for the four cursor arrow keys (next to the 1 and 0 of the keypad) are for adjusting the brightness and contrast of the screen. Try them. The two

above them turn the volume of the speaker up or down.

### MONITOR

The MONITOR is the box that looks more or less like a TV set. It is designed to sit on top of the computer on a clever little base that allows it to tilt back and forth and swivel left and right. With other computers you not only have to buy your monitor separately, you must also buy a tilting stand if you want one and an anti-glare screen. Victor generously supplied us with both. The anti-glare screen is a fine nylon mesh impregnated with carbon as the filament is made. It is stretched tightly over the cathode ray tube (CRT), and held in place by the frame around it. Do not jab the screen with a sharp instrument or you could tear the mesh. This will not hurt the computer particularly, but would probably make the screen difficult or annoying to read. It will readily absorb finger oils if you touch it, and this degrades the sharpness of the character images on the screen. It is almost impossible to remove completely finger oils that get on the mesh, so try to avoid handling that part of the monitor. Clean it occasionally with a lint-free cloth moistened with a little Windex or similar glass cleaner, but don't let it drip and run down behind the frame, or let it saturate the mesh. Use cleaners sparingly. If it is really dirty, it can be removed and washed with a mild soap in soft water. Just let it soak like you would a pair of nylon stockings (men - ask your wife if your don't feel practiced in the care of nylons), and rinse in clean water until there is no change in the color of the water. Let dry, and reinstall.

### COMPUTER (or CPU)

This brings us to the COMPUTER. This is the beige box with the black front, and cables coming out of the back. The monitor sits on top. This is also referred to as the "CPU" or Central Processing Unit. This is the box that has all the circuitry (chips and things) that make the computer compute. On the front you will either have two horizontal slots or you will have one horizontal slot to the right of center and some grille work to the left. On both models there is a small grille area to the far right for the speaker and ventilation.

The horizontal slots are FLOPPY DISK DRIVES. They are designed to receive the 5.25" magnetic disks used for storing data and programs. They come in two varieties: single-sided and double-sided. There is no way to tell single- from double- from the outside, but we'll see later how to find out which you have. If you have only one horizontal slot you have a double-sided drive and a HARD DRIVE. The hard drive is behind the grille to the left. The hard drive or HARD DISK is just like the floppy except there are actually form one to several hard

## GLOSSARY

disks which are stacked and sealed in a box, and which spin much faster than the floppies do. If the hard disk drive was installed at the factory, it is probably a ten megabyte drive, but it might be larger - we'll see later how to find that out. The drives are usually "A" on the left and "B" on the right whether they are hard or floppy. However, be warned that some machines are set up with the hard drive called "C". Most software documentation refers to the hard drive as "C" so don't let this confuse you when you start working with DOS or software manuals. Use "A" if that is what your hard disk responds to, or modify the drive letter to match your computer configuration.

Let's look at the back of the computer. As you face the back of the computer on the far right is a rocker switch at the top which turns the machine on and off. Below that is a recess which receives the female end of the power cord. Between these is a round plug with a slot. This covers a fuse.

Across the bottom are several connector receptacles. The first two on the bottom right are RS-232 SERIAL PORTS. Port A is the leftmost of the two, and Port B is to the right (again looking at it from the back). These connectors have 25 pins. Other manufacturers refer to these as COM1 and COM2, and you may see them so defined in your software programs. Whether they are called Ports A and B, or COM1 and COM2, their function is exactly the same. Only the names have been changed to protect... The receptacle to the left of the serial ports has nine pins in it, and is for the monitor cable.

To the left, with a little wire clamping bracket on each side, is the PARALLEL PORT. It has 36 pins. The serial and parallel ports are used to connect your computer to the outside world of printers, plotters, modems, other computers, etc. To the left of the parallel port is a square RESET BUTTON. CAUTION! PRESSING THIS BUTTON WILL NOT DAMAGE THE MACHINE BUT WILL DESTROY THE WORK YOU HAVE BEEN DOING SINCE YOU LAST SAVED YOUR WORK. To the far left is a receptacle similar to a telephone jack, but with eight wires. This receives the cable from the keyboard.

Plug in your keyboard, monitor cable and power cord, flip your rocker switch and let's see what happens.

(To be continued)

**ALT KEY:** In the lower left corner of the keyboard; a key analogous to the typewriter "shift" key; allows additional characters to be programmed to virtually any key. The IBM keyboard has both an ALT key and a CONTROL key. Some software re-programs the Victor keyboard to permit both.

**CONTROL KEY:** See ALT KEY.

**CURSOR:** A small bright rectangle on the screen showing you where your place is. If it is "on" a character, that character is shown in "reverse video," i.e. a black character surrounded by a block of green. On the IBM, the cursor is an annoying blinking underline.

**CPU:** Central Processing Unit; the computing part of computers.

**CRT:** Cathode Ray Tube; the TV-like screen that displays what is happening in the computer. Same as VDT.

**DISK DRIVE:** Part of the hardware that receives the magnetic disk media upon which are "written" data and programs. The FLOPPY DISK DRIVE receives 5.25" floppy disks for the input of new information or the storage of computed information. The HARD DRIVE or HARD DISK or FIXED DISK is a set of non-floppy disks that are permanently sealed in a box. The box may be installed inside the computer box or it may be external and connected to the computer via a special cable.

**DOS:** Disk Operating System: the program that runs the programs allowing you to do things with the computer. There are two for the Victor 9000: MS-DOS (the most prevalent one and very similar to PC-DOS which is specific to the IBM) and CP/M-86 (standing for Control Program/Micro with the 86 signifying that it runs the 8086, 16-bit processor chip. An early operating system, it is based on a popular and very successful forerunner called CPM, for the early 8080 and Z-80 8-bit computers.) This is a whole chapter in itself.

**FUNCTION KEY:** Keys that do not normally appear on typewriters or calculators, which are programmed by software to do certain tasks or routines when pressed. The TAB key on a typewriter is analogous: it moves the carriage a preset number of spaces when pressed.

**HARD DISK:** See DISK DRIVE

**HARDWARE:** The computer and its connected electronic and electromechanical machinery e.g.: computer, keyboard, hard drive, printer, etc. See also SOFTWARE.

**PARALLEL PORT:** A receptacle on the back of the computer to which an external device (usually a printer) is connected.

See also **Serial Port**.

**QWERTY:** The first six keys in the upper left corner of a typewriter. This describes the standard American keyboard. There are others.

**SERIAL PORT or RS-232 PORT:** A receptacle on the back of the computer to which an external device is connected. These devices can be printers, modems, mice, another computer, etc. Data is delivered in a continuous stream of individual "bits" rather than in blocks of eight bits via the Parallel Port. It is therefore slower than a parallel port, but it can send and receive data over longer distances without error.

**SOFTWARE:** The programs or instructions encoded on a disk which tell the computer how to handle data. Software is to hardware as cassette tapes are to your stereo system.

**VDT:** Video Display Terminal: The same as a CRT; the display screen.

#### RESOURCES

##### **Currently active Victor User Groups:**

Please let us know if there is one in your area and when and where you meet.

The **Bay Area Victor Group** meets the first Wednesday of each month at the Emeryville Holiday Inn at 7:30 pm.

Holiday Inn (Bay Bridge)  
1800 Powell St.  
Emeryville, CA 94608

Telephone: (415) 658-9300

Everyone is welcome and people are always anxious to help new users.

##### **Victor Sales & Service:**

If you provide Victor 9000 service, hardware or software, or know who does, send names, addresses and phone numbers to me for inclusion in a comprehensive list to be published in a future installment of this column.

There is not much written specifically about the Victor 9000 hardware. We can refer you to the Victor "Operator's Reference Guide" in the original documentation. There is an excellent "Operator's Reference Guide II" as a separate booklet packaged with Victor's MS-DOS 3.1 which is the only software currently available from Victor. If you upgrade to DOS 3.1 you will receive that and a lot more information about MS-DOS and the VBASICA language. Victor may be contacted at the address below:

##### **VICTOR**

396 Phoenixville Pike  
Malvern, PA 19356 USA

Telephone: (215) 251-5000  
FAX: (215) 251-5049

On computing in general, browse in your local public library and bookstores. Even the big chains have realized there is a market for computer books. Also try your local electronics shop, they may have an excellent book rack.

We can also recommend, with proper modesty, back issues of "Victor\*Talk" available from the address in the back of this issue.

\* \* \* \* \*

*Dan Mardesich is an interior designer doing business in San Rafael, California using a Victor 9000 and drafting with AutoCAD on a 386 clone that is even more obscure than a Victor. He can be reached at 415-454-6661 or by mail at 1569 Fourth Street, San Rafael, CA 94901.*

\* \* \* \* \*

#### **From the Boards**

#99 03 Dec 88 04:35:24 [7]

**From:** Judith Moore **To:** All  
**Subj:** dolt-tested recipe

##### **Pecan Triangles**

- 1 cup (2 sticks) unsalted butter, room temperature (means fluid in August; brick-hard in January)
- 1 cup sugar
- 1 egg, separated (from whom?)
- 2 cups sifted all-purpose flour
- 1.5 Tblsp. grated lemon peel
- 1 cup finely chopped pecans

Preheat oven 275 deg. Grease 15"x10" jelly roll pan; line with parchment. Cream butter and sugar together until light and fluffy. Beat in egg yolk; mix in flour & lemon peel. Spread dough evenly in prepared pan (this is the fun part). Beat egg white and brush over the top of dough; sprinkle nuts over and press lightly into dough. Bake until "golden brown", about 1.25 hrs. Cut hot pastry into squares and halve diagonally (that's why they're called Triangles!). Cool on rack.

\* \* \* \* \*

## V9FOSSIL / Franz Hirner

As mentioned in past articles, one of the pleasures of being the Sysop of BAVUG is participating in the development of new software for the Victor 9000. By that measure, the months of November and December, while long, were very pleasurable indeed.

The Victor 9000 had fallen behind the times in the communication software area. Yes, we have VICMINI and XMODEM from the Public Domain Library and the commercial program CrossTalk XVI, but newer file transfer protocols have been developed since these programs were written. YMODEM and ZMODEM in particular are among the new protocols that allow the transfer of files much more efficiently than the older Telink and XMODEM protocols, especially when modem speeds exceed 1200 bps. Also, many users today use PC Pursuit (a fixed \$25.00 monthly fee buys unlimited computer access across the country during evening and weekend hours). The XMODEM file transfer protocol has problems with the PC Pursuit packet switched network and file transfers. If the transfer can be completed at all, they are very slow. ZMODEM, on the other hand, is a very efficient protocol to use for PC Pursuit file transfers.

If you haven't looked lately, 2400 bps (bits per second) modems are now available for under \$200.00 and are rapidly replacing the older, slower 1200 bps modems. 9600 bps modems, while still fairly expensive, are becoming common. The program overhead in the XMODEM and Telink transfer protocols is so heavy that little or no additional effective speed is apparent when using the higher speed modems. Without a communications program to take advantage of these newer protocols, the Victor 9000 was falling behind the rest of the world. We couldn't have that!

I had done some communications programming on my Victor in preparation for revising Wall Street Weekly. I found that communications programs are among the most difficult to write. Direct program access to the hardware, finely-tuned accurate timing routines, and provisions for the numerous modems on the market are just a few of the problems that make creating communications programs tedious and slow. We might write one communications program, but undoubtedly there will be still newer, better protocols developed in the future. Who among us will take the time to write the next

generation communication program? Was there a way to keep from reinventing the wheel and provide a communications platform to the future without writing a complete program from scratch? As it turned out, yes!

A little history. Other non-IBM compatible users have also found that the communications world is not very friendly. Tom Jennings, the author of the FIDO BBS™ software, created a generic communications driver which made it possible for machines other than IBM and Victor to run the Fido™ BBS software so long as DOS 2.x was available for their machines. Thom Henderson (also the creator of ARC) chose to build on Tom Jennings work and extend the functions of Tom's first universal communications driver when he developed an automated computer mail system called SEAdog. Vince Perriello then coded a driver for his DEC Rainbow so that he could run SEAdog. About this time Wynn Wagner had encountered some serious difficulties in developing his Opus RBBS system. Specifically, he couldn't force the Greenleaf™ Communications Libraries to behave in exactly the way he felt Opus required. Enter Bob Hartman, who had worked with Thom Henderson. Having already enjoyed success in the effort with Thom Henderson, he suggested to Wynn that with very few extensions, any driver that was already SEAdog™ 4.0 compatible could drive Opus as well. About that time, Vince called Wynn to discuss porting Opus to the DEC Rainbow. Wynn called Bob, Bob called Vince, and the FOSSIL driver came into existence. FOSSIL is an acronym for "Fido/Opus/SEAdog Standard Interface Layer".

Several communications programs, BBS programs and utilities make use of the FOSSIL driver in the BBS community. Why not implement a FOSSIL driver for the Victor so that we could take advantage of not only existing software, but new programs that will be developed in the future?

FOSSIL is to communications as ANSI.EXE is to screen control. It's a "driver" that makes it possible for an application program to access the hardware of a particular machine. Any software written to use the features of the driver will run on any computer so long as a FOSSIL driver is available for that computer. There are FOSSIL drivers for the IBM, Tandy, Sanyo and DEC Rainbow computers. Why not one for the Victor?

Thanks to the efforts of John Navas, one of the regular callers to the BAVUG BBS, V9FOSSIL was born.

V9FOSSIL is a shareware product. Shareware is a software distribution system whereby the user gets to try the software before buying. If he likes the software and uses it, he is expected to register and pay for the software. It's a moral obligation. John's terms are very liberal. To quote him:

*My standard driver may be freely copied and distributed.*

*I'm not going to pretend that people have to register/contribute if they don't want to; the whole thing will strictly be voluntary.*

*A \$25 contribution requested. For that you get (if you include a stamped self-addressed diskette mailer and a blank diskette):*

- 1. Latest standard version of the driver plus a special version optimized for the NEC V-20 chip (i.e. faster).*
- 2. My MS C FOSSIL interface library source and object (for programming).*
- 3. I'll take a shot at answering questions or fixing problems (no warranty and results not guaranteed); non-contributors are on their own. Perhaps access to a private FOSSIL message area on BAVUG?*
- 4. Diskette will include some useful FOSSIL-specific software (like BT) set up for the Victor.*
- 5. Notice of any new version or bug fix of the driver released within 12 months of contribution.*

*Source code to the driver available for personal non-commercial use with a total contribution of \$50 (additional \$25).*

V9FOSSIL not only provides the means to run existing communication software (see *Binkley Article*), but also provides easy access to all necessary communications routines to enable you to write your own communications programs without the need to resort to assembler code. Your programs will run on the Victor 9000 and any other machine that has a FOSSIL Driver available.

If you would like to write your own software, the accompanying table lists supported FOSSIL functions.

You don't have to be a programmer to take advantage of all that V9FOSSIL has to offer. If you want to join the modern world of communications and run existing software that uses the FOSSIL interface pick up a copy of V9FOSSIL from BAVUG or the Public Domain Library and join the fun. Better yet, register with John Navas to receive the latest copy of V9FOSSIL, source code, utilities, etc. The more people who register, the more likely John is to continue enhancing the product for all our benefit.

Special thanks are due to John Navas for creating V9FOSSIL and to Tom Jennings for making his Victor specific code from Fido available for the project.

Both Bert Happel and I have done extensive testing of V9FOSSIL and have found the driver to be solid and dependable. Using Binkley's ZMODEM transfer protocol, even at 1200 or 2400 bps speed, we usually achieved an efficiency of 90% or better. XMODEM or Telink transfers at lower speeds are only 60% to 70% efficient on direct connections. PCP connections drop to only 20% to 30% efficiency. Using a HST 9600 bps modem, 310K of files were transferred over ordinary telephone lines in under 7 minutes. With two Victors direct wired, file transfers were accomplished at 38,400 bps. Other successful tests include the SEAdog electronic mail system. Other communications programs that support FOSSIL include the Opus bulletin board system.

If you're into modem communications and wished you could proceed beyond the limitations of XMODEM and Telink protocols, you might appreciate V9FOSSIL and the communications software that it makes available for the Victor 9000.

#### FOSSIL Driver Functions:

AH = 00h	Set baud rate
AH = 01h	Transmit character with wait
AH = 02h	Receive character with wait
AH = 03h	Request status
AH = 04h	Initialize driver
AH = 05h	Deinitialize driver

# V9FOSSIL/Track Driver

- AH = 06h Raise/lower DTR
- AH = 07h Return timer tick parameters
- AH = 08h Flush output buffer
- AH = 09h Purge output buffer
- AH = 0Ah Purge input buffer
- AH = 0Bh Transmit no wait
- AH = 0Ch Non-destructive read-ahead
- AH = 0Dh Keyboard read without wait
- AH = 0Eh Keyboard read with wait
- AH = 0Fh Enable or disable flow control
- AH = 10h Extended Control-C / Control-K checking and transmit on/off
- AH = 11h Set current cursor location
- AH = 12h Read current cursor location
- AH = 13h Single character ANSI write to screen
- AH = 14h Enable or disable watchdog processing
- AH = 15h Write character to screen using BIOS support routines
- AH = 16h Insert or delete a function from the timer tick chain
- AH = 17h Reboot system
- AH = 18h Read block (transfer from FOSSIL to user buffer)
- AH = 19h Write block (transfer from user buffer to FOSSIL)
- AH = 1Ah Break begin or end
- AH = 1Bh Return information about the driver

Additional functions are provided to support external applications. In fact, V9FOSSIL loads other programs as external applications and these programs may in turn load still other programs.

\*\*\*\*\*

### For Sale:

- 1 - V9000, 386K, 1.2 meg DS floppies, palm rest keyboard monitor, New in original box. Tested, Original software and/or updated.
- 1 - V9000 386K, 620K disks, keyboard, monitor. Never Sold. Used only in Victor dealership. Like-new condition. Original software and/or updated.
- Victor 6025 printers (Epson FX-100) New in original cartons. Includes cables.

Cables: 10 ft. shielded 36-pin Centronics. Both ends for direct hookup from V9000 parallel port to Epson FX or MX printers. Also 6 ft. ribbon cables.

### Software: ALL NEW

- Victor Graphics Tool Kit II
- Victor Audio Tool Kit (hardware, microphone, software)
- Victor/Microsoft BASIC compiler
- Victor/Microsoft FORTRAN 3.3
- WordPerfect 4.2 for V9000
- ABSTAT statistics package
- Kaleidoscope Database System with Check System Application & Instructions for V9000 or IBM
- Sensible Solutions Database System for V9000 or IBM
- Microsoft 2.0 or 3.0 with IBM disks, V9000 conversion disks and autoloader.
- ARMOR eleven-module accounting system for both v9000 and IBM, or order individual accounting modules.

### Contact:

Realtech, Inc.  
311 N. Jackson, P.O. BOX 458  
Tullahoma, TN 37388

1-615-455-6938

## BINKLEY/Franz Hirner

**B**INKLEY is a versatile communications software package with many of the latest, more efficient file transfer protocols. It is designed to interface with any computer for which a FOSSIL interface is available. To quote from the Binkley manual, Binkley is "A Free-for-the-Asking FidoNet Compatible Electronic Mail Interface and Dumb Terminal Package" written by Bob Hartman and Vince Perriello. There is no charge for Binkley. You are encouraged to express your appreciation for Binkley either by writing code or application notes, or just by sending along a "thank you" message. If you feel a monetary contribution is in order, the author suggests that "you make a contribution to your favorite local charity in the name of Binkley".

Binkley is more than just a dumb terminal package, though. Binkley supports virtually all of the important, modern file transfer protocols that have been developed since the last release of Crosstalk XVI (v3.5) for the Victor 9000 was published. Binkley supports both the popular and efficient YMODEM and ZMODEM file transfer protocols. Best of all, Binkley is one of the generic programs written to use a FOSSIL interface, such as V9FOSSIL, to communicate with the computer's hardware. Binkley not only offers the latest and fastest file transfer protocols, but it will also run on your Victor 9000 and on other MS-DOS machines as well.

Binkley is a comprehensive terminal communications and electronic mail program but most users will be concerned only with its terminal communication capabilities. Binkley uses the very efficient ZMODEM file transfer protocol (plus others) which is much more efficient than Kermit, XMODEM, Telink or other ancient protocols. Binkley was used as one of the test programs for the development of V9FOSSIL written by John Navas. I have used Binkley for four weeks and am very impressed with its capabilities. While Binkley can be used as a stand-alone communications program, I prefer CrossTalk's capability for dialing and logging on. But I can have the best of both worlds: I can start in Crosstalk, then use Crosstalk's "RUN" command to call up Binkley and take

advantage of its efficient ZMODEM and YMODEM file transfer capability when transferring files.

Both an extensive User Guide and a Reference Manual are included with Binkley (probably worth as much as the program itself -Ed.). While communications are sometimes difficult for the new user to figure out, Binkley's documentation is better than that supplied with many commercial programs. In addition to the manuals, sample configuration files are also provided to get the user up and running in a minimum of time.

The user's guide explains how to install BinkleyTerm. It also describes basic operational procedures. A glossary is provided which explains concepts and terminology that may be unfamiliar to new users. Concepts and terminology that may be of interest to new or experienced users alike are covered in the Reference Guide, an alphabetically arranged manual separate from the User's Guide.

A minimum of equipment is needed to run Binkley. A Victor 9000 with DOS 2.1 or higher operating system, floppy disks, V9FOSSIL, and an auto-dial modem compatible with the Hayes AT command set is all you need. If you want to use Binkley for a "mailer" program a hard disk should be considered, but most users will not use the mailer capabilities.

BinkleyTerm requires approximately 140k of RAM in any operational mode. Allow a little memory for V9FOSSIL and the package should run in 256K machines if you don't have many TSR's loaded.

BinkleyTerm's Terminal Mode offers functionality similar to that provided by Telix, Qmodem and ProComm. You can use your computer and modem to call on-line services and electronic bulletin board systems (BBS). BinkleyTerm offers a full selection of file transfer protocols for use in exchanging files with remote systems. It also offers optional VT-100 terminal emulation, open architecture for adding additional file transfer protocols, support for high speed modems up to 19,200 baud, session logging to a flat file or printer, and more.



Binkley is easy to set up. To install and operate BinkleyTerm for use exclusively in Terminal Mode, follow these steps:

Prepare disk media. Create a sub-directory on your hard disk, or format a single diskette.

Move the software to the newly prepared media.

With a standard text editor, or with EDLIN, edit the sample configuration file, BINKLEY.CFG, included in the distribution archive or diskette. The parameters needed are outlined in the file itself on comment lines, or you can refer to the Reference Guide for more information.

Obtain and properly install a FOSSIL driver in accordance with the directions included with your driver. See the accompanying V9FOSSIL article.

Invoke BinkleyTerm (enter 'BT' on the command line).

Press Alt-F10 for a brief help screen.

The various keystrokes available in Terminal Mode are covered in detail in the Reference Guide.

BinkleyTerm does require that ANSI screen control codes be used. You will need to run Victor's ANSI.EXE before running Binkley. Also, since Binkley makes use of both ALT-Keys and Control-Keys, a special FOSSIL Keyboard should be mod-con'd into your system before Binkley is loaded. This can all be done from a batch file, even when Binkley is run as an external program under Crosstalk.

```
:BINKLEY.BAT
ANSI
MODCON FOSSIL.KB SAVE.KB
V9FOSSIL BT
MODCON SAVE.KB
DEL SAVE.KB
```

BinkleyTerm provides a rich selection of file transfer protocols, hard coded for ease of use and efficient operation. On occasion however, other protocols may also be desired. BinkleyTerm supports external file transfer programs that conform to the standard used by Opus-CBCS. External file transfer protocols MUST identify themselves as being compatible with Opus-CBCS in order to work with BinkleyTerm. Such protocols are often available from Opus-CBCS systems. Reportedly ASCII, XMODEM, Kermit and other protocols are

available, but I haven't tried protocols other than those provided in Binkley.

The Binkley documentation includes an extensive troubleshooting section to assist in solving problems, but use as a terminal/communications program is really straightforward and should present few problems. The advanced features, such as automatic mailer systems are for the experienced modem user. Binkley has the power to grow with your capabilities including connecting into the USENET systems if you so desire.

### SCRIPTS

A script is a series of instructions used when dialing a particular system. They allow the system to "look" for particular information coming across the line, and act according when the desired information is received within a set time limit.

Scripts are essentially a mini-programming language, and as such, take study and practice to use effectively. Scripts are stored in flat ASCII text files, and edited using any standard text editor (such as DOS' own EDLIN). Once written, scripts are associated with a particular nodelist entry for use each time the given node is dialed.

Script use is triggered by the appearance of a file name, inside double quotes, in the phone number field of a nodelist entry. For example, instead of seeing 1-303-555-6789 in the nodelist data file, you might see "MYSCRIPT.SCR"1-303-555-6789.

Notice that the name of the script file is **immediately** (without spaces) followed by the area code and phone number. The field must appear in the format shown, including hyphens. The area code portion of the number can be up to 10 characters long (for use with certain long distance carriers).

Binkley supports script files to automate calling your favorite BBS or communications service. Following is a sample script to be used for dialing a system via PC Pursuit.

```
; set the baud rate
baud 1200
; call telenet
xmit |telenet number|
; wait for the 1200 from CONNECT 1200
pattern 0 1200
wait 30
; tell telenet we are here
```



init string is sent to the modem after DTR is toggled.

#### **ALT-J**

Jump to DOS. Invokes COMMAND.COM to allow "quick and dirty" DOS commands on the fly. Enter "EXIT" at the DOS prompt to return to BinkleyTerm.

#### **ALT-L**

Toggle session logging on and off. The session logging function allows you to designate a file or printer, and have the entire terminal mode session echoed to the selected file or device.

When first invoked, this command will prompt you for a file name. Type in the desired name of the log file, including drive and path designation if desired. You may also enter a printer device name, e.g.: LPT1, LPT2, etc. for printer echoing.

Invoke the command again to end logging to the file or device.

#### **ALT-M**

Poll a node in the nodelist, by node address. This command requires that mail handling be implemented as described in the User's Guide section, "Unattended Mode".

You are prompted for a FidoNet node address. Once entered, BinkleyTerm will dial the system (if it exists in the nodelist) and attempt to exchange mail with the system. If there is outgoing mail to the system, or if mail is waiting for you on the remote system, it will be sent during the mail session.

#### **ALT-P**

This command allows you to change the communications port in use. Invoking the command shifts the port number to the next higher port. The number of ports supported is determined by your hardware, and by the 'Maxport' statement in the configuration file.

#### **ALT-R**

This allows you to dial out with a "scan

list". Empty elements are designated by 'Null.' Using the function keys (or the equivalently numbered numeric keys) you can select the scan list element you wish to enter. Enter a telephone number just as you would with the ALT-D command. Press 'Enter' to "store" your entry for that element. When you have stored all the desired elements, press 'Enter' to begin the dialing process.

Each number will be dialed in sequence, until a connection is made. After the session is completed, a 'Null' will be stored for that element. Invoke ALT-R again, and press 'Enter' to dial the remaining elements. Press 'Escape' at any time to abort the dialing process.

Note that the dialing list is volatile. Once you exit the Terminal Mode, your dialing list will most likely be erased.

#### **ALT-S**

Invoking this command sends a "break" signal via the modem to the remote host. This command requires that a FOSSIL of revision 5 or later be installed. (V9-FOSSIL conforms to v5 specifications).

#### **ALT-U**

This command is used in Point and BBS installations to shift to Unattended Mode. BinkleyTerm is put into the mailer mode, ready to accept calls from remote systems. In order to operate, the mailer functions must be implemented as described in the User's Guide section, "Unattended Mode".

#### **ALT-X**

Allows you to exit the program and return to DOS. If BinkleyTerm was invoked with a batch file, this command will cause BinkleyTerm to exit to the batch file with an errorlevel of 1. The batch file must trap this errorlevel, and exit accordingly.

Note that this function DOES leave the DTR line (data terminal ready) on the modem 'high.'

### ALT-Y

Used primarily for Point operations, this function initiates a mail session with the host. This is designed ONLY to work with the system listed as your Boss node in the configuration file.

Issuing this keystroke invokes a temporary mailer mode, and BinkleyTerm attempts to dial the Boss system to exchange mail. Mail handling must be set-up as described in the User's Guide section, "Unattended Mode". If mail is waiting to be sent to the Boss, or if mail is waiting on the Boss system for pickup, it will be sent during the mail session.

### PgDn

This allows you to download a file from the host with which you are connected. You are prompted for the desired protocol to use, and for the required file information, if needed (protocol dependent). If you have the 'Download' statement properly implemented in the configuration file, the downloaded files will be placed in the directory specified. Otherwise, the file will be saved to the current directory.

### PgUp

This allows you to send a file to the remote host. You are prompted for the desired protocol to use for the transfer, as well as the file information. You may specify a complete drive, path and filename using standard DOS conventions, e.g., C:\PATH\FILENAME.ARC.

When a batch protocol is used (SEALink, Telink, or ZMODEM), wildcards are allowed in the filename.

Binkley can be used as a stand-alone communications package or called as a process under CrossTalk. Either way, Binkley may provide all the communications capability you need. You will find that Binkley is a full-featured communications package that will increase your Victor's capability. Give it a try.

\* \* \* \* \*

V9000 for sale  
384K, 2 double-sided drives  
very good condition  
asking \$300.00

John Marcella  
759 Temescal Way  
Redwood City, CA 94062  
1-415-368-3064

\* \* \* \* \*

An assortment of Victor V9000 computers for sale - some with enlarged memory, hard disk, network cards and much, much more. Some machines have 70 hard disks, 896K and networking. All or part at very reasonable prices.

Contact: Glenn E. Copas  
1680 Civic Center Drive Suite 109  
Santa Clara, CA 95050

1-408-247-4475 or 1-408-251-5255

\* \* \* \* \*

For Sale: 4 ea. V9000, each with 512K 10 meg. hard disk, good working condition. Includes following software: MS DOS 2.11, WordPerfect 4.1, Lotus 123 1A, Multiplan 1.06 Also includes manuals, shipping crates, monitor. Make best offer.

Contact: Madison Valuation Associates  
Attn.: Cathy Patterson  
Madison, WI

1-608-257-2757

\* \* \* \* \*

For Sale: V9000 complete with keyboard and monitor. 10 meg. hard disk 512K, PC PLUS installed, XTALK software, manual for V9000. Asking \$500.00. Questions or reasonable offers, call below.

Contact: Sherry Everett  
1-215-293-7302 days, or  
1-215-855-0140 eves., weekends

\* \* \* \* \*

## CONTRIBUTORS

To enable our readers to contact our authors, we provide the authors' addresses and telephone numbers (unless requested otherwise). Here are this issue's contributors:

Hugh Connelly  
13441 Pt. Pleasant Drive  
Chantilly, VA 22021  
(703) 378-1912

C. Robert Elford  
4097 39th Avenue  
Oakland, CA 94619  
(415) 531-5755

Franz Hirner (x3)  
631 Matsonia  
Foster City, CA 94404  
(415) 349-3602

Dan Mardesich  
1569 Fourth Street  
San Rafael, CA 94901  
(415) 454-6651

Al Thompson  
1126 Breezewood Court  
Sunnyvale, CA 94084  
(408) 737-2259

\* \* \* \* \*

## VictorTalk

### Address inquiries to:

BAY AREA VICTOR USERS GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

**Subscriptions:** Carol Hirner

**Editor:** George Anderson  
782 Jordan Ave.  
Los Altos, CA 94022  
(415) 941-3029

**Layout:** Andrew Rudiak

**VictorTalk** is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in **VictorTalk**. Copyright 1988 by Victor\*Group. Contents may be reproduced by permission only. **Permission is only a post card away.**

**VictorTalk** is an independent publication not affiliated in any way with Victor Technologies, Inc. **Victor** is a registered trademark of Victor Technologies, Inc.

**Submit Articles** on MSDOS disks in ASCII or Spellbinder format (either single-sided or double-sided, in Victor or IBM format), with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

**BAVUG BBS** is on line 24 hours a day: (415) 574-7914, Franz Hirner, Sysop.

**VICTOR\*GROUP meetings** are normally held at 7:30 PM on the first Wednesday of the month at the Holiday Inn, Powell Street, Emeryville, CA (adjacent to Highway I-80).

## Yes, I want to join.

Enclosed is a check/money order for \$35.00 (foreign membership, \$40.00 is U.S. dollars) for my 1989 Victor\*Group membership.

Name \_\_\_\_\_

Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

vol.3 no.4 4/85

Make checks payable to:

## VictorTalk

1493 Beach Park Boulevard, #180  
Foster City, CA 94404

City \_\_\_\_\_

State \_\_\_\_\_ ZIP \_\_\_\_\_

Country \_\_\_\_\_

Phone \_\_\_\_\_

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
City _____ State _____ & Zip _____	
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Un-archived versions are available where "A", "AB", or "ABC" appears below number. If originally released (archived) version is desired, cross out corresponding "A", "AB", or "ABC". (Original releases of all other disks contain no archived files.) Note that pricing below is per "disk number" rather than per "disk" (re-released unarchived "disk numbers" may be comprised of one disk (A), two disks (AB), or three disks (ABC). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No. Disks	Vol. No. Disks	Vol. No. Disks	Vol. No. Disks
4 2 61-64	5 1 78-81	5 5 87-90	6 3 96-97
4 3 65-71	5 2 82-84	5 6 91-92	6 4 98
4 6 72-77	5 4 85-86	6 2 93-95	6 5 99

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54 A	55	56*	57*	--	59	60 AB
61	62	63	64	65	66	67	68	69	70	71	72	73 A	74	75	76	77	78 AB	79 AB	80 AB
81 AB	82 AB	83 AB	84 AB	85 AB	86 AB	87 AB	88 ABC	89 ABC	90 AB	91 AB	92 ABC	93 ABC	94 AB	95 ABC	96 AB	97 AB	98	99	101

\* 34 & 35 are a two-disk set; 56 & 57 are also a two-disk set

1988 Victor\*Group member: No  Yes  Date joined/renewed \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, 1988, sets of 6, \$12.00 /set.. \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1989 membership fee \$35 (receive all 1989 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR\*GROUP

1493 Beach Park Blvd., No. 180

Foster City, CA 94404

BULK RATE  
U.S. POSTAGE  
PAID  
LAFAYETTE, CA 94549  
PERMIT NO. 39



PRINTED MATTER

## IN THIS ISSUE:

**Page**

- 1 I/O REVISITED**  
Everything you ever wanted to know about input, output and then some. Either from the files of Police Squad or **Channel 9000** (my notes are murky on this point)
- 11 PUBLIC DOMAIN DISK #101 DISKID**  
Last issue we told you about BinkleyTerm, V9FOS-SIL, and CHK\_FLOP. Disk #101 is where you find'em.
- 12 THE VICTOR PRIMER**  
**Dan Mardesich** continues this informative series for the **absolute** beginner
- 16 AUDIO ERRATICA**  
**Hugh Connelly** corrects a few booboos and appends some programs to last issue's article on how to build your own CODEC board.
- 21 NEW USER'S CORNER**  
New user **Paul Schmielewski** checks in with a few thoughts
- 22 POP QUIZ**  
Score as follow:  
1-5 points: **Gong Show** watcher  
6-10 points: **USA Today** reader  
11-15 points: Video game player  
16-20 points: Average **VictorTalk** reader  
20-24 points: Next **VictorTalk** editor
- 23 ISSUE NEWS**  
More Victor news, commentary and products from the Netherlands



# VICTOR<sup>®</sup> TALK

Bi-Monthly Magazine for Victor 9000 users

VictorTalk, Vol. 7, No. 2, Mar-Apr 1989

## I/O Revisited

Many of us need to, or have wanted to, perform a control or data gathering function which should be something a computer can do well. But it never gets done because when it comes time to start hooking things up, we find that we just do not have enough information to make the external hardware work with our computer. In this article we will attempt to provide you with information which will go a long way to furnishing you with the data you need. We are going to cover the V9000 specifically, but the principles are similar for other computers, and you just might get enough insight into the process to try your hand at VIs or even (heaven forbid!) an IBM or clone. With that said, let us look at the methods by which the V9000 sends and accepts signals via its 'I/O' (Input/Output) ports.

The Victor 9000 is provided with input/output ports able to process data either in a serial (Serial Port) or a parallel (Parallel Port) fashion. By proper matching-up of the encoding or decoding capabilities of the port we plan to use, and those same protocols on the hardware we plan to hook up to the V9, we can accomplish a wide variety of tasks under computer control. In essence we adjust the signal requirements through software, firmware or hardware to match the computer's interface coding, as seen at the port. We will concentrate on some of the common interfacing tasks between peripherals and hardware associated with the port and located in the CPU. To do this, we will also discuss associated software considerations and their role in marrying a CPU port with peripheral hardware.

### V9000 SERIAL PORTS

The V9000 has two serial ports located on the lower portion of the back panel of the CPU, under the power supply. These two ports, called Com Port 1 (COM1) and 2, or SERIAL A and B, are the two rightmost connectors on the CPU back panel. You might be using one of them for connecting a serial printer, or a modem now, but there are many other peripherals you can connect to these ports. Serial ports come in two standard flavors generally, and one should know the difference if for no reason other than to get yourself out of a jam, when your new hookup doesn't work.

### THE EIA RS232C SERIAL STANDARD

The two types of standard serial port are the Data Communication Equipment (DCE) port configuration, and the Data Terminal Equipment (DTE) configuration. DCE equipment was originally conceived of as equipment which communicated or sent signals (regardless of the use of those signals), while DTE equipment was conceived of as being terminal equipment communicated-with such as a terminal connected to a mainframe computer, or a

printer (though not all printers are wired as DTE equipment). An example of DCE equipment is a modem. There are many other examples in industry.

These standards are defined in an Electronic Industries Association (EIA) Standard, RS232. RS232 specifies that a "1" shall be represented by a voltage level of -3 to -20 volts, while a "0" is represented by a voltage level of +3 to +20 volts. This provides enough voltage to drive the signal over fairly long distances without becoming weak enough to be overwhelmed by electronic noise generated in the equipment or picked up by the cable. The standard is explicit on which signals and signal levels go on which pins, but they do not define in detail what equipment may be called 'DCE' or 'DTE' equipment. Based on the description above, one might expect to connect DTE equipment to DCE equipment. Would that it were so easy. Some manufacturers wire their equipment to the configuration opposite what you might be led to expect. Therefore, confusion sometimes occurs in that some manufacturers call a common piece of equipment 'DCE' while others call essentially the same equipment 'DTE'. The connectors on the V9 back panel are configured as 'DTE', meeting the original intent of the EIA standard. So, to work.

### CONNECTIONS

Connections made between equipment conforming to the DCE - DTE Standard would be done using a cable with wires terminating on the same numbered pin at each end of the cable. The connectors used on serial ports are specified in the Standard, and the V9 uses the standard 25 pin "D Subminiature" connector. It is widely available in the industry. Another well-known brand of computer, capable of establishing a de facto standard by itself, uses a 9 pin version of the same connector type, since by no means are all the pins which are defined used in real life circumstances.

The table below lists the V9's serial ports A and B DTE connector pins and the signals on those pins. It shows the connections to any DCE device which meets the RS232C standard. The arrow heads show the direction of signal flow. If there are heads at both ends, it means that sometimes the flow is one way, and at other times the flow is reversed:

DTE CONN	DCE CONN	SIGNAL FUNCTION
1	1	Protective Ground
2	2	Transmitted Data



A register is at address E800:0041. In addition to being the source of the auxiliary signals, two bits of this register are used to control the selection of the internal or external baud rate clocks. The bit assignments for the VIA port are:

Bit	Signal
0 (LSB)	Serial A Clock Select (0=internal)
1	Serial B Clock Select (0=internal)
2	Port A Ring Indicator
3	Port A Data Set Ready
4	Port B Ring Indicator
5	Port B Data Set Ready
6	Keyboard Data Input
7 (MSB)	Vertical Signal Input

#### BAUD RATES

The MPSC requires a clock to control the baud rate (the baud rate is approximately the same as bits-per-second), which can be derived either from an internal or an external source. The internal baud rate clock is an 8253 IC chip called a Programmable Interval Timer. If an external clock source is used, it is supplied through pins 15 & 17 of the serial port connectors. Use of these pins to supply an external baud clock is V9 specific, and does not reflect pin assignments standardized in the RS232 Standard. Therefore, when working with serial equipment to be connected to the V9, be sure use of these pins for an external baud clock does not conflict with pin assignments in the equipment in question, and that a suitable clock is available, or can be made available, through these pins.

For the internal baud rate clocks, counter 0 of IC 8253 is used as the Port A baud rate clock. Counter 1 of the same IC is used for the Port B baud rate clock. A 1.25 MHz signal is fed to counters 0 and 1, and the counters divide the 1.25 MHz signal to give the appropriate clock rate. IC 8253 is programmed to provide a square wave output, known as Mode 3 of the 8253. The 8253 registers are memory mapped at these locations:

Register	Address
Counter 0 . . . . .	.E000:0020
Counter 1 . . . . .	.E000:0021
Counter 2 . . . . .	.E000:0022
Mode Word . . . . .	.E000:0023

Divisors used for producing commonly used baud rates are given in the following table:

Baud Rate	Low Byte	High Byte
300	04h	01h
600	82h	00h

Baud Rate	Low Byte	High Byte
1200	41h	00h
2400	21h	00h
4800	10h	00h
9600	08h	00h
19200	04h	00h

For example, to set the baud rate for port A to 9600 baud, execute the following instruction sequence:

```
MOV AX,0E000H           ;set I/O base address
MOV ES,AX
MOV BX,0020H           ;set 8253 offset
MOV BYTE PTR ES:3[BX],36H ;load counter 0 mode 3
MOV BYTE PTR ES:0[BX],08H ;least significant byte
MOV BYTE PTR ES:0[BX],00H ;most significant byte
```

External baud rate clocks may be fed to the V9 through pins 15 & 17 of the serial port connectors. Pin 15 is the pin on which the external 'transmit baud rate' clock is delivered to the computer serial port, and pin 17 is the pin on which the external 'receive baud rate' clock is delivered. These clock inputs are generally used only with high-speed synchronous modems; but since they are specific to the V9 they should be used with caution - make sure you can supply the requisite clocks on those pins from the external serial equipment.

Switching between internal and external baud rate clocks is done by setting two output signals on VIA2 (see Auxiliary Signals, above). The current state of these bit settings can be determined by reading the VIA register. For example, if you wanted to change only port A to external clocks, you would execute the following instruction sequence:

```
MOV AX,0E800H           ;load I/O base address
MOV ES,AX
MOV BX,0041H           ;load register offset
                        ;(port B, here)
MOV AL,ES:[BX]         ;get current state
OR AL,01H              ;force bit 0 to 1
MOV ES:[BX],AL         ;output new state
```

The above sequence maintains the previous state of register bit 1 (the bit controlling port B).

#### EXTERNAL PARALLEL PORT

The parallel port on the back of the V9 was designed to be used either as a Centronics parallel printer port or as an IEEE-488 parallel data port. It may be used as a Centronics parallel port without any change, since all pins are wired to the Centronics de facto standard. Due to the differences in the IEEE-488 and the Centronics standards, connections must be adapted to

the IEEE-488 standard through use of a special adaptor cable.

The parallel port is controlled by VIA1, (base address E800:0020), with a few signals controlled by VIA2. The port signals are buffered using bus transceivers designed for IEEE-488 compatibility (75160 & 75161). The transceivers are wired in such a way that, in the IEEE-488 mode, the V9000 must be the controller (and the only controller) on the buses.

The direction of most of the signals on the connector are controlled by the state of the VIA1-PB2 output. The table below gives, for each pin of the connector, the source and direction with regard to the output of VIA1-PB2. It should be noted that for a VIA2-PB2 low output, the direction of pin 15 is controlled by the output of VIA1-PB3 (pin 35). If the output of VIA1-PB3 is low, pin 15 is an output, otherwise it is an input.

Pin	VIA2-PB2		Signal Source
	L	H	
1	I	O	VIA1-PB0, VIA1-CA2
2	I	O	VIA1-PA0
3	I	O	VIA1-PA1
4	I	O	VIA1-PA2
5	I	O	VIA1-PA3
6	I	O	VIA1-PA4
7	I	O	VIA1-PA5
8	I	O	VIA1-PA6
9	I	O	VIA1-PA7
10	O	I	VIA1-PB6, VIA1-CA1
11	I	O	VIA1-PB5, VIA2-CA2
12	-	-	N/C
13	O	I	VIA1-PB7, VIA1-CA2
14	-	-	N/C or GND
15	I/O	O	VIA1-PB1
16	-	-	GND
17	-	-	CHASSIS GND
18	-	-	N/C
19-29	-	-	GND
30-31	-	-	N/C
32	-	-	TIED TO PIN 13
33	-	-	GND
34	O	O	VIA1-PB2
35	O	O	VIA1-PB3
36	O	O	VIA1-PB4

Care should be taken when configuring the port to make sure that there are no driver conflicts between the VIA's and the transceivers. In other words, if signals are to be changed from outputs to inputs, change the VIA's direction before changing the transceiver's direction. If signals are to be changed from

inputs to outputs, then change the VIA's direction.

The parallel port connector on the back of the V9 uses a ribbon connector. A ribbon connector is different than a ribbon cable. Ribbon, when used to describe a connector refers to the contacts used in the connector. These are made of a small, flat ribbon of plated metal - hence, the name. The connector looks different in that the usual pins and sockets are not used; just the ribbons in place of the pins. Ribbon connectors may be used with either a ribbon cable or a standard round cable.

### CENTRONICS INTERFACE PROTOCOL

The Centronics Interface protocol, as implemented on the V9, is a very simple, 8-bit, parallel data transfer protocol. It performs three functions. They are:

1. ensure the printer is on-line and not busy
2. place the data on the interface data lines
3. strobe the printer

The BUSY and ON-LINE signals are high true. The DATA STROBE is a low going pulse.

The current V9 BIOS programs configure the VIA-PB0 output high and use only the following pins for data transfer:

Pin	Signal	<----- IN : OUT ----->
1	Data Strobe	: ----->
2	Data 1 (LSB)	A
3	Data 2	L
4	Data 3	L
5	Data 4	D-- : ----->
6	Data 5	A
7	Data 6	T
8	Data 7	A
9	Data 8 (MSB)	-*-
11	Printer Busy	<----- :
13	Printer On-Line	<----- :

### IEEE-488 INTERFACE

#### or GENERAL PURPOSE INTERFACE BUS (GPIB)

A full description of the operation of the GPIB bus would take a book, and is beyond this article. Instead, we will discuss some of the factors involved in connecting GPIB equipment to the V9.

The interface connector on the back of the V9 is not configured for standard GPIB connection. The standard GPIB connector is a 24-pin, ribbon connector. The table below shows how to wire an adaptor cable to allow connection of IEEE-488/GPIB devices to the V9. In addition, pins 18 - 24 on the GPIB connector should be wired to the GND pins on the V9 connector.

V9	GPIB	Signal	Pin	Signal	Pin	Signal
1	6	DAV	9	N/C	10	LIGHT PEN IN
2	1	DIO 1	11	GN	12	CA1
3	2	DIO 2	13	GN	14	CA2
4	3	DIO 3	15	GN	16	PA0
5	4	DIO 4	17	GN	18	PA1
6	13	DIO 5	19	GN	20	PA2
7	14	DIO 6	21	GN	22	PA3
8	15	DIO 7	23	GN	24	PA4
9	16	DIO 8	25	GN	26	PA5
10	7	NRFD	27	GN	28	PA6
11	10	SRQ	29	GN	30	PA7
13	8	NDAC	31	GN	32	PB0
15	5	EOI	33	GN	34	PB1
17	12	SHIELD	35	GN	36	PB2
34	17	REN	37	GN	38	PB3
35	11	ATN	39	GN	40	PB4
36	9	IFC	41	GN	42	PB5
			43	GN	44	PB6
			45	GN	46	(Codec) PB7
			47	GN	48	CB1
			49	GN	50	CB2

#### NOTES:

1. DIO lines are bidirectional, and may be used for commands or data
2. DIO LOW = '1'; DIO HIGH = '0' - watch the polarity of the signal!

The IEEE-488 driver will be incorporated into the BIOS at system generation time instead of the Centronics driver. The driver functions will be controlled by sending the appropriate ESCAPE sequences to the driver. As mentioned above, the V9 must be the only controller on the buses, and will not pass control to any other device. It should be pointed out that one of the sixteen GPIB signals is missing from the V9 implementation: the DATA Valid (DAV) signal. This signal can be dispensed with if the signal Not Ready For Data (NRFD) is held high (e.g.: with a resistor biasing the line to +5 volts).

#### USER PORT

The V9 provides an internal parallel port accessible through a 50-pin ribbon cable and connector. The connector is wired to VIA3, and is memory mapped with a base address E800:0080. All of the functions of VIA3 are available with the exception of Timer 1 and output PB7, which are used for the CODEC clock (audio output). The configuration of the port connector is given below:

Pin	Signal	Pin	Signal
1	-12 VOLTS	2	-12 VOLTS
3	N/C	4	N/C
5	+12 VOLTS	6	+12 VOLTS
7	+5 VOLTS	8	+5 VOLTS

While the usual method of attaching to the user port is by using a ribbon cable and connector, I have found another way that allows the plugging of a small circuit board directly into the connector. AP Products Inc. of Mentor, Ohio, makes a connector (part number 922576-50) intended to be used to attach two ribbon cables to one header. However, this connector can be soldered to a circuit board and used to connect the circuit board to the user port. Care must be taken with the size of the circuit board to insure that it does not hit the disk unit support bracket directly over the user port connector. This limits board height to about 1.9 inches.

I used this method for the design of a board to connect a daisy-wheel printer with a parallel interface to the V9. On a board 1.9 inches high and 2.7 inches wide, I was able to get four 20-pin IC's, a resistor pack, a 50-pin ribbon cable connector, and the AP Products connector. Designing the interface this was considerably less expensive than using an expansion bus board.

#### DEVICE DRIVERS

Now that we have discussed the hardware aspects of I/O, it seems appropriate to give attention to the software aspects, i.e. device drivers. But first we should define the concept of logical and physical devices.

A logical device is a generalization of a physical mechanism that performs a particular function. Printers, which perform the function of transferring electronic data to paper, are commonly

referred to as listing devices. By our definition, a listing device is a logical device, since it doesn't refer to a particular type of printer. An HP DeskJet, on the other hand, is a physical device. You might think of a logical device as a 'blank' to be filled in, while a physical device is what you fill the blank with.

MS-DOS defines four logical (general, not specific) devices that it directly supports. The first is the CONSOLE device, the principal interactive device for communication with the operator. Second is the LIST device, the device to which hard copy output is usually sent. The last two are the AUXILIARY INPUT and AUXILIARY OUTPUT devices which are nondescript input and output devices that are usually user defined. (An example of an AUXILIARY OUTPUT device would be a second printer.)

Hardware independent software running under MS-DOS (meaning software that does not depend on the specific features of a given computer) performs all I/O via these logical devices. The program does not know, or care, about the actual source or destination of its communications. The computer operator connects the desired physical devices to the computer and then tells the operating system which physical device to associate with each logical device.

However, manufacturers do not know for certain what make, or even what type, of physical device will be connected to their computers. Since they do know what provisions they have made to connect external devices to their computers, meaning I/O ports, they have taken to referring to these I/O ports as the physical devices, rather than the mechanisms to which they are attached.

In addition, the software that drives the I/O port, the device driver, is considered to be a part of the physical device. Some manufacturers provide more than one physical device that all use the same I/O port, but incorporate different driver software. This is useful in the situation, for example, where sometimes a printer requiring hardware handshaking is used on a port, and at other times a printer requiring software handshaking is attached to the same port. By changing the logical-to-physical device assignment, the software driving the port is changed. (This concept is not currently implemented with MS-DOS or CP/M for the V9).

### IOBYTE

Each of the four logical devices mentioned above, can be associated with one of four physical devices. Not all physical devices can be assigned to a given logical device. The table below gives the names of the physical devices that can be associated with each logical device. Where the CP/M names differ from the MS-DOS names, the CP/M names are given in parentheses.

Logical Device	Physical Devices
CON	TTY CRT BAT UCT

Logical Device	Physical Devices
AUXIN (AXI or RDR)	TTY PTR UR1 UR2
AUXOUT (AXO or PUN)	TTY PTP UP1 UP2
LST	TTY CRT LPT UL1

The following table describes the physical devices that are currently implemented on the V9:

Physical Device	Description
CRT	Keyboard input / CRT display output
TTY	Serial Port A
LPT	External parallel port (Centronics)
UL1	Serial Port B
BAT	Input from AUXIN / output to LST

The current state of the logical to physical device assignments is kept in a byte in system memory called the IOBYTE. The IOBYTE is divided into four two-bit fields, one field for each logical device. The value of a field (0-3) indicates which physical device is assigned to its corresponding logical device. The field definitions are outlined below.

CON (Bits 0,1)	AUXIN (Bits 2,3)
0 - TTY	0 - TTY
1 - CRT	1 - PTR
2 - BAT	2 - UR1
3 - UCT	3 - UR2

AUXOUT (Bits 4,5)	LST (Bits 6,7)
0 - TTY	0 - TTY
1 - PTP	1 - CRT
2 - UP1	2 - LPT
3 - UP2	3 - UL1

It is possible to examine and set the IOBYTE by using the SETIO utility under MS-DOS. It is also possible to examine and set the IOBYTE from a program by using the following sequence:

```

; MS-DOS get IOBYTE to register al
MOV AX,DS ;set ES to data segment
MOV ES,AX
LEA BX,PB1 ;get offset of parameter block
MOV AX,16 ;get IOBYTE function
INT 223 ;Victor super-bios interrupt
MOV AX,PB1 ;get IOBYTE to al (0 to ah)
-----
PB1 DW 0 ;parameter passing area

```

```

; MS-DOS set IOBYTE from register al

MOV  AH,0    ;set upper byte to 0
MOV  PB1,AX  ;set IOBYTE in parameter block
MOV  AX,DS   ;set ES to data segment
MOV  ES,AX
LEA  BX,PB1  ;get offset of parameter block
MOV  AX,17   ;set IOBYTE function
INT  223    ;Victor super-bios interrupt
-----
PB1 DW 0     ;parameter passing area

```

## CUSTOM DEVICE DRIVERS

### NOTE:

In preparing the following section, it was discovered that some of the features mentioned did not work, or contained bugs. Specifically, function 5 for the List Vector and Console Vector calls does not work because the BX register is clobbered by the BIOS code, and the "pass-on" feature passes on one character too many (the extra character is meaningless). Victor has stated that since these are undocumented features of the BIOS, they are not directly supported except as they pertain to Victor programs. Therefore, the likelihood of the bugs being fixed is small. We decided to include the discussion of these features anyway, in case the bugs are fixed, and as a note of warning to users who might have wanted to use these features.

At some time it may be required to attach the V9 to a device that is not compatible with the standard device drivers supplied by Victor. This was the case when I plugged my old Diablo 1620 into one of the V9's serial ports. My Diablo does not support hardware handshaking, and using the Victor driver, I was forced to run at 300 baud or lose data. Because of this, I decided to write my own device driver that would support the Diablo's ETX/ACK software protocol.

It turns out that there are two ways of incorporating a different device driver into the V9 system. The first, which is only viable for MS-DOS, involves writing a replacement module for the standard one supplied by Victor as part of their system generation program. The sysgen program used MS-LINK to combine pre-assembled modules to form the MS-DOS system file. By writing a new module and changing the linker command file to cause the linker to load the new module instead of the standard one, it is possible to create a system disk incorporating the new driver. This task requires an intimate knowledge of the V9 BIOS and should only be attempted by experienced programmers.

Fortunately, the second method is somewhat easier. The V9

BIOS routes all calls to the LST device through a long vector stored in memory. The BIOS also provides a simple method by which to examine or change this vector. A device driver written to use this second method is a COM or CMD file that consists of two parts. The first part is the initialization section, which configures the port hardware, changes the list vector to point to the new list driver entry points, and then exits to the system, leaving the module permanently resident in memory. The second part is the device driver itself. The module will remain as the new device driver until the system is reset.

The second method is the method I used to implement my Diablo driver. The driver for MS-DOS is shown in Listing 1.

### LIST VECTOR

Before doing an indirect long call through the list vector, the BIOS sets the AX register to indicate the function to be performed.

AX	Function
0	Get data from list device
1	Send data to list device
2	Get list device input status
3	Get list device output status
4	Get list device input vector
5	Set list device input vector

Data to be output is passed in the CL register and input in the AL register. Status is returned in the AL register (00h is busy or no data ready, FFh is not busy or data waiting).

The list device input vector (functions 4 and 5) is a long vector containing the address of the list device input routine. The standard BIOS does not provide an input routine for the list driver and this vector points to a null subroutine that returns zero in the AX, BX, and ES registers. This vector can be used when it is desired to add an input routine for the list device, accessible through the standard list vector, without changing the list device output routine. The list input vector is set from and returned in registers ES (segment) and BX (offset).

Output to the LST device from MS-DOS produces calls through the main list vector for functions 1 and 3 only (see above). There is no way to get input from the LST device using standard system calls. In order to call the other functions, the user must make a BIOS call to get the list vector and then do a long indirect call using the address obtained. An example of getting the list input status is given below:

```

MOV  AX,DS   ;set ES to data segment
MOV  ES,AX
LEA  BX,PB1  ;set offset of parameter block

```

```

MOV AX,14      ;get device vector function
INT 223       ;Victor super-bios interrupt
MOV AX,2      ;get input status function
CALL DWORD PTR LSTV ;call through vector
               ;the input status is in AL

```

```

-----
PB1 DW 2      ;list vector function
LSTV DD 0     ;list vector is returned here
               ;first word is offset
               ;second word is segment

```

The following code sequence is used to change the list vector:

```

MOV AX,DS     ;set ES to data segment
MOV ES,AX
LEA BX,PB1    ;get offset of parameter block
MOV AX,15    ;set device vector function
INT 223      ;Victor super-bios interrupt

```

```

-----
PB1 DW 2      ;list vector function
    DW LSTOFF ;new list vector offset
    DW LSTSEG ;new list vector segment

```

### CONSOLE VECTOR

The V9's BIOS also provides a vector scheme for the console device (CON), similar to that for the list device outlined above. This scheme is very useful for adding functions to the keyboard or display. For example, it is used by Victor to implement the calculator program and graphics package. The console vector may be examined or set using the following routines:

; get console vector

```

MOV AX,DS     ;set ES to data segment
MOV ES,AX
LEA BS,PB1    ;get parameter block offset
MOV AX,14    ;get device vector function
INT 223      ;Victor super-bios interrupt

```

```

-----
PB1 DW 1      ;console vector function
CONV DD 0     ;console vector returned here
               ;first word is offset
               ;second word is segment

```

; set console vector

```

MOV AX,DS     ;set ES to data segment
MOV ES,AX
LEA BX,PB1    ;get parameter block offset
MOV AX,15    ;set device vector function
INT 223      ;Victor super-bios interrupt

```

```

PB1 DW 1      ;console vector function
    DW CONOFF ;new console driver offset
    DW CONSEG ;new console driver segment

```

As with the list driver, the AX register is used to indicate the function to be performed by the console driver.

AX	Function
0	Get data from console (i.e. keyboard)
1	Send data to console (i.e. display)
2	Get console input status
3	Get console output status
4	Get user defined console vector
5	Set user defined console vector

Data to be output is passed in the CL register and input in the AL register. Status is returned in the AL register (00h is busy or no data ready, FFh is not busy or data waiting).

The user defined console (UDC) vector (see functions 4 and 5, above) is provided to allow a simple method of switching between any two of the standard console drivers (TTY and CRT) and a custom driver. Setting the IOBYTE (see above) to map the CON device to the UCI driver causes all console I/O to be routed through the UDC vector. This reduces switching between standard and custom driver to a simple and familiar process that can be done in command mode or under program control.

There is an additional feature provided in conjunction with the use of the UDC vector. This feature is known as the pass-on feature. It allows for the "passing on" of ESCAPE sequences that are not part of the standard CRT driver. In other words, if the CRT driver determines that an ESCAPE sequence it has received is not one that it was designed to recognize, the CRT driver will pass it on through the UDC vector to a secondary, or background, driver.

Furthermore, the UDC driver can cause a flag to be set so that it directly receives all data subsequent to the ESCAPE sequence, preventing the data from going to the CRT driver. When the UDC driver decides it does not need further data, it causes the flag to be reset - thus resuming its background status. The flag is set by the UDC driver returning a non-zero value in the AL register whenever it is called. The flag is reset by the UDC driver returning zero in the AL register.

### Listing 1

#### MS-DOS Custom Device Driver Example



```

; replacement diablo driver using etx/ack protocol
iosseg equ 0E000h ;I/O base address
sioda equ 40h ;MPSC port offsets
siodb equ 41h
sioca equ 42h
siocb equ 43h
ctc0 equ 20h ;8253 port offsets
ctc1 equ 21h
ctc2 equ 22h
ctcct1 equ 23h

cseg segment public
assume cs:cseg,ds:cseg

fcb org 05Ch
db 1 dup (?) ;use file name for port spec
org 100h ;this org for COM file

public dbdinit ; initialize for port B

dbdinit proc near ;init entry point
mov ax,cs
mov ds,ax
mov es,ax
mov byte ptr portf,0 ;assume port A
mov al,fcb+1 ;get port selection
and al,0DFh ;make upper case
mov okmsg1,al ;put port in msg
cmp al,'A'
jz s0 ;port A
cmp al,'B'
jz s ;skip if valid port
jmp s9 ;port selection error

s: mov byte ptr portf,OFFh ;select port B

s0: mov bx,offset oldvec
mov ax,14 ;get print vector
int 225 ;super-bios call
mov ax,6 ;test for alt driver
call dword ptr prtvec
cmp ax,0D1ABh ;Diablo driver installed?
jnz s1 ;not installed, continue
mov dx,offset msg1 ;driver already installed
mov ah,9
int 33
xor ax,ax ;exit to system
int 33

s1: mov ax,cs ;store segment in parameter
mov vector+4,ax ;passing block
mov es,ax ;and in ES for call to bios
mov bx,offset vector ;get parameter block address
mov ax,15 ;set new device vector
int 225 ;super-bios call
mov bx,iosseg ;set print port to 1200 baud
mov es,bx
cmp byte ptr portf,0 ;which port
jnz s2

; initialize for port A
mov byte ptr es:ctcct1,56h ;init port A to 1200 baud
mov byte ptr es:ctc0,41h

mov byte ptr es:ctc0,00h
mov byte ptr es:sioca,18h ;reset channel
nop ;wait awhile
nop
nop
nop
mov byte ptr es:sioca,4 ;init 7201
mov byte ptr es:sioca,047h ;16x clk, 1 stop, even parity
mov byte ptr es:sioca,3
mov byte ptr es:sioca,0AAh ;rx enable 7 bits
mov byte ptr es:sioca,5
mov byte ptr es:sioca,0AAh ;tx enable 7 bits
mov byte ptr es:sioca,2
mov byte ptr es:sioca,010h
mov byte ptr es:sioca,1
mov byte ptr es:sioca,000h ;no interrupts
jmp s3

; initialize for port B
s2: mov byte ptr es:ctcct1,76h ;init port B to 1200 baud
mov byte ptr es:ctc1,41h
mov byte ptr es:ctc1,00h
mov byte ptr es:siocb,18h
nop
nop
nop
nop
mov byte ptr es:siocb,4 ;init 7201
mov byte ptr es:siocb,047h
mov byte ptr es:siocb,3
mov byte ptr es:siocb,041h
mov byte ptr es:siocb,5
mov byte ptr es:siocb,0AAh
mov byte ptr es:siocb,2
mov byte ptr es:siocb,010h
mov byte ptr es:siocb,1
mov byte ptr es:siocb,000h ;no interrupts

s3: mov ah,9 ;all ok message
lea dx,okmsg
int 33
mov dx,offset endall
int 39 ;exit but remain resident

s9: mov ah,9 ;invalid specification
lea dx,inmsg
int 33
xor ax,ax ;exit to system
int 33

dbdinit endp ; list call entry point
dbdent proc far
dbdel: mov bx,iosseg ;point to I/O base address
mov es,bx
cmp al,3 ;test function number
jz dbdosx ;status check
cmp al,1

```

```

jz dbd ;print char
cmp al,6 ;installation check?
jnz dbde2
mov ax,0D1ABh ;return driver ID message
ret

dbde2: xor al,al ;return to caller
dec al
ret

; see if we can output a byte and if so, do it

dbd: call dbdos ;can we send ?
or al,al
jz dbd ;loop if not
cmp byte ptr cs:portf,0 ;which port ?
jnz dbd0 ;skip if port B
mov es:si0da,cl ;send char to port A
jmp dbd1

dbd0: mov es:si0db,cl ;send char to port B

dbd1: lea bx,cs:hscnt ;check char count
dec byte ptr cs:[bx]
cmp cl,01Bh ;sent esc ?
mov al,cs:[bx]
jnz dbd3 ;skip if not esc
cmp al,2 ;force 2 chars after esc
jnb dbd2 ;ok, more than 2 to go
mov byte ptr cs:[bx],2 ;make count 2

dbd2: ret

dbd3: or al,al ;ready for handshake?
jnz dbd2 ;exit if not
mov byte ptr cs:dbdcts,1 ;flag time for shake
ret

; status check and handshake routine

dbdosx: call dbdos ;call status routine
ret

dbdent endp

; output status near procedure

dbdos proc near
cmp byte ptr cs:dbdcts,2 ;waiting for ack?
jnz dbdos1 ;skip if not

; must wait for ack, check for incoming char

call dbdsts ;get port status
and al,01h ;char received?
jz dbdosb ;exit if not
call dbddta ;get char
and al,07Fh ;strip parity
sub al,06h ;sub ack
jnz dbdosb ;not ack, exit
mov cs:dbdcts,al ;clear flag
mov byte ptr cs:hscnt,32 ;reset char counter

dbdos1: call dbdsts ;get port status
and al,04h ;can we send ?

jz dbdosb ;no, exit
mov al,byte ptr cs:dbdcts ;time to shake ?
or al,al
jnz dbdos2 ;yep, shake it
dec al ;indicate ready to send
ret

dbdos2: inc al ;inc flag indicating etx sent
mov cs:dbdcts,al
cmp byte ptr cs:portf,0 ;which port ?
jnz dbdos3 ;skip if port B
mov byte ptr es:si0da,03h ;send etx to port A
jmp dbdosb ;exit

dbdos3: mov byte ptr es:si0db,03h ;sent etx to port B

dbdosb: xor al,al ;indicate busy
ret

; get port status to al

dbdsts: cmp byte ptr cs:portf,0 ;which port ?
jnz dbdst1 ;skip if port B
mov al,es:si0ca ;get port A status
ret

dbdst1: mov al,es:si0cb ;get port B status
ret

; get port data to al

dbddta: cmp byte ptr cs:portf,0 ;which port ?
jnz dbddt1 ;skip if port B
mov al,es:si0da ;get port A data
ret

dbddt1: mov al,es:si0db ;get port B data
ret

dbdos endp

; data storage

vector dw 2 ;new driver vector block
dw dbdent
dw 0

oldvec dw 2 ;old driver vector block
prtvec label dword
prt0ff dw 0
prtseg dw 0

hscnt db 32 ;handshake counter
dbdcts db 0 ;handshake flag
portf db 0 ;port A/B flag

okmsg db 'Diablo ETX/ACK protocol established '
'on port '
okmsg1 db 'B at 1200 baud.',0dh,0ah,'$'
msg1 db 'Diablo driver already installed.',07h,0dh,0ah,'$'
inmsg db 'Invalid port specification.',07h,0dh,0ah,'$'

endall equ $
cseg ends
end dbdinit

```

=====VICTOR\*GROUP PUBLIC DOMAIN LIBRARY=====

DISK #101

Release Date: January 15, 1989

Bay Area Victor Users Group  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

This address is the one in use on the release date as noted above for this diskette. Please check the latest issue of our news publication, VictorTalk, for any possible change of address.

----- <<<<<< >>>>>> -----

This disk contains three Public Domain Communications Software Programs for the Victor 9000, including the V9FOSSIL communications driver and BINKLEY 150.

Directory of DISK101

Program Name	Ext.	Size (bytes)	Additional Information
DISK101	.TXT	7424	Information about the files on this disk.
KERM229	ARC	34048	The latest release of this popular communications program. Kermit, developed and maintained at Columbia University, is especially useful for communicating with University mainframe computers. It also works well on PC Pursuit Packet Switched Network for file transfers, although it is much slower than ZMODEM protocol.
MINITEL	ARC	93523	The popular Public Domain Communications Program written by Tom Jennings. While auto-dial capability has not been built into the program, the Victor's programmable keyboard makes it easy to automate dialing your favorite BBS, CompuServe, DowJones or other services.  The IBM version of the program is included as well as the Victor version.  MINITEL is ideal for those new to modem communications.
BEXE_150	ARC	176985	Binkley Communication Program. Requires V9FOSSIL to run. Supports the very efficient ZMODEM file transfer protocol, automatic dialing, modem speeds to 19,200 bps. Binkley may also be used as a full featured mailing system. ZMODEM is suggested for use with high speed modems (2400 bps or faster). Very powerful, well documented, but requires some knowledge to take advantage of all features.
V9FOSSIL	.EXE		V9FOSSIL is a FOSSIL communications driver written specifically for the Victor 9000. V9FOSSIL enables communications written to comply with FOSSIL interface requirements to run on the Victor 9000. Binkley is one such program.  The source code for V9FOSSIL may be licensed and used to develop your own communications programs.

# The Victor Primer/Dan Mardesich

## INTRODUCTION

In the first installment of this series we took a careful look at all the physical parts of your "new" Victor. Now we're going to see if it runs. You have all your cables plugged into the proper receptacles, and you are ready to plug the power cord into the wall. You can go ahead and do that, but most people recommend plugging it into a "surge suppressor" (see glossary). The wall outlet is safe enough for now so go ahead and do that.

Flip the rocker power switch on the back of the CPU box to the ON position. One of two things will happen: nothing or something.

If nothing happened, check your connections. If you are using a surge suppressor, bypass it and plug the computer directly into the wall. If it works, your surge suppressor is bad. If not, check the wall outlet. Is it controlled by a wall switch? Plug in a lamp or appliance to test it. If all your connections are good and the power source is OK, you have a problem. See RESOURCES at the end of this article.

## THE BOOT

More likely, you will hear a humming noise and in a few seconds you will see something at the bottom of the screen. There will be a block letter "M" (for Memory) to the left of center, a number in about the middle of the line (which may be in "decimal" or "hex") a square with a circle in it (imitating a floppy disk) on the right and, between the number and the floppy icon, a flashing block arrow pointing up. The computer is asking for an operating system.

The above process has been accomplished because the computer has certain permanent instructions in something called ROM (Read Only Memory) that wake it up and get it going. This is called "booting" since the computer "pulls itself up by its bootstraps." There is a "cold boot" when you turn the machine on or turn it off and on again, and there is a "warm boot" when you restart the computer by pressing the "reset" button or by pressing a certain combination of keys.

One of the things which the ROM instructs the computer to do during this process is to check to see how much RAM (Random Access Memory) it has. This is the number that appeared at the bottom of the screen. It will be in multiples of 64K. If it said "128K" or "256K" or some other number up to "896K" you have that number of kilobytes of memory and "new" ROMs. If it is some weird number, possibly with letters in it, it is in hex (base 16), and you have "old" ROMs. New ROMs are updated versions issued by Victor that enable the computer to handle certain tasks that the old ROMs could not accomplish, or handle them more elegantly. If you have old ROMs you should still be able to do most things. However, if you have old ROMs, you have an older machine and it would be a good idea to have someone go over the machine, clean it and install new ROMs while they are still available. You will be able to use newer software, DOS 3.1 and the IBM BIOS and good things like that which you don't need to know about at this point, and your machine will tend to maintain its value. (See RESOURCES)

## ROM and RAM

Don't confuse ROM and RAM. They are both "memory" and both are stored on "chips," but ROM memory is permanently programmed into the chip(s). RAM is impermanent, fugitive memory that is dependent upon the power staying on and into which the computer loads data and instructions as to what to do with the data.

The 8088 processing chip, which is the heart of your Victor, can "address" one megabyte of RAM. The designer of any computer "reserves" a certain portion of this one megabyte of addressable memory for certain internal functions. This leaves the balance for the loading of programs and data. In the IBM world the magic number, 640K, which you have seen in all the ads is the maximum memory available for programs and data. Victor handles memory differently and the maximum number is 896K, indicating one of the many incompatibilities between the two systems. You need a minimum of 128k of RAM in your Victor for it to do anything, and the most it can use (except for RAM disks, but

lets not talk about that just yet) is 896K. On the single-sided, double-floppy Victor this 128K minimum has been included on the "mother board" which is the main printed circuit board that houses the CPU chip as well as all the other chips and things the computer needs. On the double-sided machine, 256K is installed on the mother board.

You add RAM to your computer by installing a "card" with additional memory on it. A "card" is a printed circuit board which plugs into a socket inside the computer and which has space to plug in memory chips. You must buy a board that is specifically designed for the Victor 9000. IBM boards do not fit. RAM chips come in 64K and 256K sizes and are rated to run at various speeds from 80 to 150 nanoseconds (ns). Your Victor can use the relatively inexpensive 64K, 150ns chips or the 256K chips as fast as 100ns.

One 64K or 256K chip does not give you 64K or 256K of memory. You need eight (one for each bit in your byte, apparently.) So the general practice (naturally) is to sell them in bunches of nine. I suppose this is because of the fragile nature of the chips (they are easily permanently zapped by static electricity for instance) and it has probably proven to be easier on everybody if you buy an extra 12.5% when you are buying chips. I don't think it is any more sinister than that, but one of my editors will probably jump in somewhere with a proper explanation. [Actually, the ninth chip is for parity-checking in the IBM world. V9000s don't parity check. Victor reasoned that parity checking was superfluous as parity errors usually cause a program crash, unequivocally informing users of the problem.-Ed.]

Most memory boards are designed to accommodate a **clock chip** in addition to several blocks of RAM chips. We'll see below why you might want one. The clock is set using software which comes with the board and is kept running by a battery mounted on the board. It tells the computer the time and date whenever it is called for. The clock chip may be a few dollars extra, but the increased precision, keystrokes saved, and reduction in the level of aggravation are worth it.

#### THE OPERATING SYSTEM

The operating system is a program that runs programs. It acts as a sort of traffic cop or obses-

sive mother who tells the computer how to do things and where to put things: where in memory (RAM) to load the program, what to do with input from the keyboard, where and how store data, all that sort of thing. It keeps your files organized on your floppy or hard disk and has various "utilities" or sub-programs that allow you to manipulate your files by copying them from one place to another, deleting them, printing them on a printer, displaying them on the screen, displaying information about them, etc. There are many operating systems around. Two developed for the Victor 9000 are CP/M-86 and MS-DOS.

#### CP/M-86

CP/M stands for "Control Program/ Micro" and was developed for the Z-80 and Intel 8080 micro-processor chips. The Z-80 and 8080 chips were 8-bit chips that could address 64K of memory. When the computer explosion started in the late 1970's there was much software developed for CP/M. The two most famous programs were Word-Star and dBase II.

Intel later developed the 8086 and 8088 chips which were 16-bit micro-processors which could address more memory and run faster than the 8-bit chips. A number of machines based on the 8088 chip were introduced in the early 1980's including one from IBM and one from Victor Technologies. Naturally, everyone wanted the newer, faster machines. The authors of CP/M brought out CP/M-86 to make it easier for software developers to adapt the large base of existing programs to run on the new machines. The primary programs offered by Victor when the 9000 was released were Word-Star, dBase II and others which had been "ported over" from CP/M. Many early Victor Group Public Domain Disks are CP/M-86 and include games and utilities as well as more elaborate programs. CP/M-86 is essentially dead, but all those early programs are as usable now as they were in 1983.

Unfortunately, no program is perfect (to say the least) and only successful ones are updated and revised to correct problems and improve performance. When CP/M-86 programs were updated they tended to be reissued in MS-DOS format. This was not a problem at the time since the Victor had both operating systems and there were utilities for converting data from CP/M-86 to MS-DOS format. Today, however, if you want to upgrade,

say, an inherited CP/M-86 WordStar to a Victor MS-DOS version, I don't think you'll have much luck. Besides, there are better word processors available for the Victor than WordStar v3.3.

### MS-DOS

One of the legends of the micro-computer era is the famous story about some people from IBM who came to visit the author of CP/M to discuss developing an operating system for Big Blue who had decided to put out one of these "toy" computers. The young software developer, who had become quite wealthy from the fact the "everyone" was using his product was off flying (or was it gliding?) in his plane and stood-up the gentlemen from IBM. The story goes that this snub so annoyed them that they immediately went to Washington to talk to a young man named Bill Gates who had a company called MicroSoft and an operating system whose initials stood for MicroSoft-Disk Operating System. The rest, as they say, is history.

Both operating systems do essentially the same thing. MS-DOS tends to do them more elegantly, however, and because of its relationship to IBM has become the "standard". Each of the three versions of MS-DOS released for the Victor has improved the scope and flexibility of commands and file handling. I won't say much more about DOS, but will refer the reader to George Anderson's excellent series in Volume 6 Numbers 2 through 6 of VictorTalk. Also excellent is Van Wolverton's **Running MS-DOS**. Although it's not Victor specific and therefore omits some commands which were developed for the 9000 (e.g., CPM.-EXE) and discusses others that were not included in the Victor MS-DOS (e.g., TREE.COM), it is very useful and very readable.

### THE BOOT CONTINUES

Meanwhile, that arrow at the bottom of screen continues to blink at you, asking for an operating system. Look through the disks you received with your computer and see if there is anything labeled MS-DOS or CP/M-86. Place the disk in the left-hand drive (this is the "A" drive) and close the drive door by pushing down on the tab that is sticking out at the top of the wide vertical slot. The machine will not attempt to read a disk unless the door is closed. If you have located the proper disk, there will be whirring and grinding

noise, the arrow will stop flashing and turn into a clock face and the computer will load the operating system.

(If you have a hard disk, chances are that the operating system is installed on it. In that case the machine will boot from the hard disk when it turned on and the following not apply. If you have a hard disk that did not boot, you will have to "format" it and install the operating system so that it will boot. More of that later.)

If the operating system disk is a plain, naked one, you will see the Victor logo announcing that this is a "VICTOR 9000 BUSINESS COMPUTER" and below that a box in reverse video that will say something about which operating system has loaded and its version, the "I/O Subsystem", the keyboard and character set (they are changeable in the Victor world,) some copyright information and perhaps some other information if it is a later version of MS-DOS. CP/M does not care about the date, but each time a file is created or updated, MS-DOS marks it with the time and date. This is very useful when you have similarly named files and are looking for the one that was done on a certain date. Therefore, the next thing you will see on your screen is

Enter new date:

Enter today's date if you want to accurately date your files or hit a Return to accept 1/01/1980. The screen will then say...

Current time is 0:03:27.5

Enter new time:

The current time is the amount of time since the operating system loaded. You can leave it like that by entering a Return and you can ask for the time when you are finished to find out how long you have been working, or you can enter the actual time of day so that files will be accurately time stamped as well as dated. Time is kept on a twenty-four hour clock, i.e. 1 PM is 13:00, etc.

The screen will then present you with...

A>

You have arrived at the famous "A prompt" which

tells you that you are "logged on" to Drive A: (drives almost always are referred to with a colon following) and that the computer is ready to receive some instructions.

If you have booted up from a hard disk, chances are that the person from whom you bought the machine has set it up to do more than just present you with an A-prompt. This is also possible if you have a floppy-disk machine and the seller has given you a "boot disk" that has been "customized." The same is true on a floppy machine if you have inserted, say, a word processing disk and have moved directly into that program. Next time we will talk about how to do some real work and how to customize your disks.

#### GLOSSARY:

**Byte:** (including Kilo-, Mega-, Giga-) A byte is the amount of memory or disk storage space required to store one character. It is made up of eight "bits" which are either "1" or "0" i.e. "on" or "off." Bytes are often measured in blocks based on powers of 2. A kilobyte is not one thousand bytes as a kilometer is one thousand meters. A kilobyte is  $2^8$  or 1024 bytes. A megabyte is 1024 kilobytes or  $2^{16}$  bytes. "Giga" stands for billion and is equal to 1024 megabytes or  $2^{32}$  bytes.

**Nanosecond (ns):** a millionth of a second. (This was in general use before it was discovered by Robin Williams.)

**OEM:** Original Equipment Manufacturer. When a manufacturer puts his own identification on something from another manufacturer (legally, through some licensing agreement) it is referred to as an OEM product. E.g., The Victor 6040 letter quality printer is a Diablo 630 dressed in Victor beige. Similarly, VictorCalc was an OEM edition of Datamension's "Report Manager"

**Surge Suppressor:** a box which plugs into a wall outlet and into which electronic equipment is plugged. The box may plug directly into the wall or it may be on the end of a cord so that it looks like a high-class extension cord. The box contains circuitry that catches power line blips before they can zap your RAM or even your disk and cause you to lose data. You can pay from \$7.95 up for the higher priced ones presumably capable of handling larger and more vicious blips.

#### RESOURCES

Repair and maintenance, ROMs, hardware, etc. Call or write for prices and information. **This listing does not imply an endorsement by the author or Victor\*Group.**

Al Bullock and Associates  
39522 Benavente Place  
Fremont, CA 94538  
(415) 651-6183

Stratos Systems  
1336 Bromer Street  
Suite A1  
Santa Cruz, CA 95062  
(408) 479-1713

Victor Technologies, Inc.  
396 Phoenixville Pike  
Malvern, PA 19356  
(215) 251-5000  
FAX: (215) 251-5049  
(The only Victor 9000 product they have to sell is MS-DOS ver. 3.1.)

#### CP/M-86:

Try your local public library for books on CP/M and CP/M-86. There is a very fast turn-over in computer books and they tend to become stale or out-of-date rather quickly. Things tend to stay on the shelves longer at the library than they do at the book store. If you can find it, a good one is

Hogan, Thom.

**Osborne CP/M User Guide, 2nd Edition**  
Osborne/McGraw-Hill, 1982

#### MS-DOS:

There are lots of books on MS-DOS. Some are for you and me (like the Wolverton book) and some are for people who are interested in "interrupts" and things like that. Look them over before you buy. Two excellent and available sources are

Anderson, George.

"Exploring MS-DOS", Series in VictorTalk Volume 6, No. 2,3,4,5,6 (Specific to your Victor 9000.)

Wolverton, Van.

**Running MS-DOS, 2nd Edition**, MicroSoft Press  
Bellevue, Washington, 1984, 1985  
(General MS-DOS and PC-DOS, but with good descriptions of how to use the operating system and its utilities to do things you thought you needed a word processor or a database to do.)

\* \* \* \* \*

# AUDIO ERRATICA /Hugh Connelly

[The following article corrects and supplements the audio board article in our last issue. -ED]

Thanks to Al Bullock, who is building the preamp board, a number of errors have been discovered in my article on building the audio preamplifier that appeared in the latest issue of VictorTalk. The first three errors (items #1 and #2) also occur in the original article contained in Victor Circle Newdisk 36. The rest are from editorial or publishing mixups.

1. Step 2 of the step-by-step instructions has two incorrect coordinates. Pin 5 of the 6-pin jack should connect to 12C5 (not 12D2 as stated). Also points 1A1 and 2A1 (not 1A1 and 1A2 as stated) should be soldered together on the bottom side of the board.
2. Step 39 of the step-by-step instructions has the power connections reversed. The +12V supply lead goes to Pin #1 (not pin #6 as stated) and the +6V bias goes to Pin #6 (not pin #1 as stated). (The schematic accompanying the articles does show the correct power supply to pin connections.)

The remaining corrections apply only to the VictorTalk article:

3. Figure 3 was omitted from the article. It is available from the author on request.
4. Figure 2 (the schematic) omitted most pin number references and some connections for the LM324 quad. op-amp. The two sections of the LM324 shown in the upper left of the schematic (when the schematic is turned horizontally) should show the following connections and pin number references:

The far left section should have:

Pin 2 -- negative input  
Pin 3 -- positive input  
Pin 1 -- output  
Pin 4 -- to +12 Volt supply  
Pin 11 -- to ground

The section to the right should have

Pin 6 -- negative input  
Pin 5 -- positive input  
Pin 7 -- output

The section at the bottom right should have

Pin 13 -- negative input  
Pin 12 -- positive input  
Pin 14 -- output

The section at the upper far right has the pin number references and is correct.

5. Figure 2 shows resistor R14 as 16K. It should be 10K (as indicated in the parts list).
6. The parenthetical statement under the upper two 15-hole rows in Figure 1 should read: (THE NUMBERS BELOW THE LETTERS ARE NOT PRINTED ON THE BOARD) The statement refers to the numbers shown beneath A,B,C,D,E & F.
7. In column 2 of page 3 of the article, the step-by-step instructions say that the two rows of holes with numbers assigned from 1 to 15 are not printed on the circuit board. The numbers 1, 5, 10, and 15 are printed on the board... the intervening numbers aren't.
8. The parts list on page 4 shows R17 as 330K ohms. It should be 330 ohms (the schematic shows the correct value).

Although not an error, a warning is needed regarding the Radio Shack circuit board. Apparently, Radio Shack now has a revised version of the (276-168) board. The new board is part number 276-168A. Unfortunately, this version - according to Al Bullock - has the letters at the top of the columns going from left to right (viewing from the component side) rather than right to left as on the 276-168 board described in the article. If the 276-168A board is used, I suggest that you apply a strips of tape or cut a floppy label into strips and write the original lettering sequence on them so you don't lose your mind trying to translate the instructions.

My apologies for the errors./Hugh Connelly

```
-----  
;V I N  
; - - -  
; PURPOSE:  
; This subroutine allows a BASIC program to
```



; input an array of CODEC data to create  
 ; arbitrary sound patterns. The interface to  
 ; BASIC is through the CALL statement:

```
100 CALL VIN(LNG%,DAT%(0),SPD%,VOL%)
```

Where:

LNG% is the length of the DAT% CODEC  
 data array (in xxx bytes).

This variable must be INTEGER (%).

SPD% is the CODEC Record/Playback speed.

VOL% is the CODEC volume.

;Typically these items would be defined as follows:

```
10 DIM DAT%(4096)      '4K WORDS OF DATA
20 WLNQ%=4096         'DAT% LENGTH 4K WORDS
30 BLNQ%=8192         'DAT% LENGTH 8K BYTES
40 SPD%=8H20          '16KHZ SAMPLE SPEED
50 VOL%=0             'MAXIMUM VOLUME
60 CALL VIN(BLNQ%,DAT%(0),SPD%,VOL%)
```

;INPUT DATA

```
70 END
```

;ON ENTRY:

1. VOL% POINTER IS AT SP+4.
2. SPD% POINTER IS AT SP+6.
3. DAT%(0) POINTER IS AT SP+8.
4. LNG% POINTER IS AT SP+10.

;PROCESSING:

1. SAVE ES.
2. INITIALIZE THE CODE VIA VINIT.
3. GET THE STACK BASE POINTER INTO BP.
4. SET ES TO THE I/O SEGMENT BASE.
5. SET UP THE 6852 SSDA FOR INPUT MODE:
  - A. SET AC2/1 TO 00 FOR C2 ON NEXT WRITE.
  - B. WRITE C2 DATA (5CH) TO FORCE DTR TO INPUT DIRECTION.
  - C. SET AC2/1 TO 11 TO SELECT TRANSMITTER FIFO ON NEXT WRITE.
6. SET UP THE CODEC VOLUME:

- A. USING BP PULL OUT THE VOL% POINTER FROM STACK.
- B. USING THE VOL% POINTER GET THE VOL% DATA.
- C. STORE THE VOL% DATA INTO THE VOLUME SHIFT REG.

7. SET UP THE CODEC CLOCK SPEED:

- A. USING BP PULL OUT THE SPD% POINTER FROM STACK.
- B. USING THE SPD% POINTER GET THE SPD% DATA.
- C. STORE THE SPD% DATA INTO THE TIMER 2 CODEC CLOCK.

8. USING BP PULL OUT THE D%(0) POINTER INTO SI.

9. PUT THE LNG% DATA INTO CX:

- A. USING BP PULL OUT THE LNG% POINTER.
- B. USING THE LNG% POINTER PUT THE LENGTH INTO CX.

10. DISABLE INTERRUPTS (PREVENTS OVER-RUNS).

11. TEST 6852 SSDA STATUS FOR RECEIVE READY (BIT 0).

12. IF RECEIVE DATA READY, RECEIVE A BYTE OF DATA:

- A. READ DATA FROM SSDA.
- B. STORE DATA INTO BUFFER.
- C. INC SI.

13. LOOP TO STEP 11 UNTIL LENGTH IN CX EXPIRED.

14. ENABLE INTERRUPTS.

15. RESTORE ES.

16. RETURN.

;ON EXIT:

CODEC DATA IS RECEIVED.

#### ASSEMBLY CODE

Code for the VIN program follows:

```
PUBLIC VIN
NAME VIN
EXTRN VINIT:FAR
CSEG SEGMENT 'CODE'
CSEG ENDS
DSEG SEGMENT PUBLIC 'DATA'
DSEG ENDS
```

```

SSEG SEGMENT STACK 'STACK'
SSEG ENDS
ESEG SEGMENT AT 0E800H
      ORG 02AH
VOLUME DB ? ;VOLUME CONTROL SR
      ORG 060H ;6852 BASE AT E8060
SSDCTL DB ? ;6852 SSDA CTL
SSDDAT DB ? ;6852 SSDA DATA
      ORG 084H
CDCLK DW ? ;CODEC CLOCK
ESEG ENDS
CGRP GROUP CSEG
DSGRP GROUP DSEG, SSEG
SSGRP GROUP SSEG
ESGRP GROUP ESEG
-----
CSEG SEGMENT
      ASSUME
CS:CGRP,DS:DSGRP,ES:ESGRP,SS:SSGRP
-----
;VIN (Note: Do NOT type in the line numbers
;-- or leading semicolons below)
-----
VIN PROC FAR
; 1. PUSH ES ;SAVE ES
; 2. CALL VINIT ;INITIALIZE CODEC
; 3. MOV BP,SP ;BP=SP BASE
; 4. MOV AX,ESEG ;PUT I/O BASE
; MOV ES,AX ;.. IN ES
; 5. A. MOV SSDCTL,000H ;SET AC2/1=0
; 5. B. MOV SSDDAT,05CH ;WRITE C2 FOR DTR IN
; 5. C. MOV SSDCTL,000H ;SET AC2/1=11 (XMIT FIFO)
; 6. A. MOV BX,[BP]+4+2 ;GET VOLUME PTR
; 6. B. MOV AL,[BX] ;GET VOLUME DATA
; 6. C. MOV VOLUME,AL ;MOVE VOLUME DATA
; 7. A. MOV BX,[BP]+6+2 ;GET CODEC SPEED PTR
; 7. B. MOV AX,[BX] ;GET SPEED WORD
; 7. C. MOV CDCLK,AX ;MOVE SPEED DATA
; 8. MOV DI,[BP]+8+2 ;GET DATA POINTER
; 9. A. MOV BX,[BP]+10+2 ;PULL LNG% PTR
; 9. B. MOV CX,[BX] ;GET CX LENGTH
;10. CLI ;(PREVENT OVER
; ;RUNS);CLEAR INT.
; ;ENABLE VLOOP:
;11. TEST SSDCTL,001H ;CHECK FOR RECEIVE
; ;DATA READY
; JZ VLOOP ;WAIT TIL READY
;12. A. MOV AL,SSDDAT ;GET CODEC DATA
;12. B. MOV [DI],AL ;SAVE DATA
;12. C. INC DI ;MOVE BUFFER PTR
;13. LOOP VLOOP ;INPUT TIL DONE
;14. STI ;SET INT. ENABLE

```

```

;15. POP ES ;RESTORE ES
;16. RET 8 ;RETURN PAST 4 PARAMS
;
VIN ENDP
CSEG ENDS

```

-----  
END VIN  
-----

VOUT

-----  
PURPOSE:

This subroutine allows a BASIC program to output an array of CODEC data to create arbitrary sound patterns. The interface to BASIC is through the CALL statement:

100 CALL VOUT(LNG%,DAT%(0),SPD%,VOL%)

Where:

LNG% is the length of the DAT% CODEC data array (in bytes). This variable must be INTEGER (%).

SPD% is the CODEC Record/Playback speed.

VOL% is the CODEC volume.

Typically these items would be defined as follows:

```

10 DIM DAT%(4096) ;4K WORDS OF DATA
20 WLN%=4096 ;DAT% LENGTH 4K WORDS
30 BLN%=8192 ;DAT% LENGTH 8K BYTES
40 SPD%=8H20 ;16K-HZ SAMPLE SPEED
50 VOL%=0 ;MAXIMUM VOLUME
60 CALL VOUT(BLN%,DAT%(0),SPD%,VOL%)

```

OUTPUT DATA

70 END

ON ENTRY:

1. VOL% POINTER IS AT SP+4.
2. SPD% POINTER IS AT SP+6.
3. DAT%(0) POINTER IS AT SP+8.
4. LNG% POINTER IS AT SP+10.

PROCESSING:

1. SAVE ES.

2. INITIALIZE THE CODE VIA VINIT.
3. GET THE STACK BASE POINTER INTO BP.
4. SET ES TO THE I/O SEGMENT BASE.
5. SET UP THE 6852 SSSA FOR OUTPUT MODE:

- A. SET AC2/1 TO 00 FOR C2 ON NEXT WRITE.
- B. WRITE C2 DATA (5CH) TO FORCE DTR TO OUT DIRECTION.
- C. SET AC2/1 TO 11 TO SELECT TRANSMITTER FIFO ON NEXT WRITE.

6. SET UP THE CODEC VOLUME:

- A. USING BP PULL OUT THE VOL% POINTER FROM STACK.
- B. USING THE VOL% POINTER GET THE VOL% DATA.
- C. STORE THE VOL% DATA INTO THE VOLUME SHIFT REG.

7. SET UP THE CODEC CLOCK SPEED:

- A. USING BP PULL OUT THE SPD% POINTER FROM STACK.
- B. USING THE SPD% POINTER GET THE SPD% DATA.
- C. STORE THE SPD% DATA INTO THE TIMER 2 CODEC CLOCK.

8. USING BP PULL OUT THE D%(0) POINTER INTO SI.
9. PUT THE LNG% DATA INTO CX:

- A. USING BP PULL OUT THE LNG% POINTER.
- B. USING THE LNG% POINTER PUT THE LENGTH INTO CX.

10. MAKE STRINGS GO FORWARD.
11. DISABLE INTERRUPTS (PREVENTS OVER-RUNS).
12. TEST 6852 SSSA STATUS FOR TRANSMIT READY (BIT 1).
13. IF READY, SEND A BYTE OF DATA:
  - A. GET BYTE OF DATA FROM BUFFER & INC SI.
  - B. MOVE DATA TO 6852 SSSA TRANSMIT FIFO.
14. LOOP TO STEP 13 UNTIL LENGTH IN CX EXPIRED.
15. ENABLE INTERRUPTS.
16. RESTORE ES.
17. RETURN.

; SYNCS (AA).

## ASSEMBLY CODE

Assembly code for VOUT follows:

```

PUBLIC VOUT
NAME VOUT
EXTRN VINIT:FAR
CSEG SEGMENT 'CODE'
CSEG ENDS
DSEG SEGMENT PUBLIC 'DATA'
DSEG ENDS
SSEG SEGMENT STACK 'STACK'
SSEG ENDS
ESEG SEGMENT AT 0E800H
ORG 024H
VOLUME DB ? ;CODEC VOLUME CONTROL SR
ORG 060H ;6852 BASE AT E8060
SSDCTL DB ? ;6852 SSSA CTL
SSDAT DB ? ;6852 SSSA DATA
ORG 084H
CDCLK DW ? ;CODEC CLOCK
ESEG ENDS
CSGRP GROUP CSEG
DSGRP GROUP DSEG,SSEG
SSGRP GROUP SSEG
ESGRP GROUP ESEG
;-----
CSEG SEGMENT
ASSUME
CS:CSGRP,DS:DSGRP,ES:ESGRP,SS:SSGRP
;-----
;VOUT (Note: Do NOT type in the line numbers
;-----
; or leading semicolons below)
;-----
VOUT PROC FAR
; 1. PUSH ES ;SAVE ES
; 2. CALL VINIT ;INITIALIZE CODEC
; 3. MOV BP,SP ;BP=SP BASE
; 4. MOV AX,ESEG ;PUT I/O BASE
; 5. MOV ES,AX ;... IN ES
; 5. A. MOV SSDCTL,000H ;SET AC2/1=0
; 5. B. MOV SSDAT,05EH ;WRITE C2 FOR DTR OUT
; 5. C. MOV SSDCTL,0C0H ;SET AC2/1=11 (XMIT FIFO)
; 6. A. MOV BX,[BP]+4+2 ;GET VOLUME PTR
; 6. B. MOV AL,[BX] ;GET VOLUME DATA
; 6. C. MOV VOLUME,AL ;MOVE VOLUME DATA
; 7. A. MOV BX,[BP]+6+2 ;GET CODEC PEED PTR
; 7. B. MOV AX,[BX] ;GET SPEED WORD
; 7. C. MOV CDCLK,AX ;MOVE SPEED DATA

```

;ON EXIT:

CODEC DATA IS SENT, CODEC IS QUIETLY SENDING

```

; 8.  MOV  SI,[BP]+8+2 ;GET DATA POINTER
; 9. A. MOV  BX,[BP]+10+2 ;PULL LNG% PTR
; 9. B. MOV  CX,[BX] ;GET CX LENGTH
;10.  CLD ;AUTO INCREMENT SI
;11.  CLI ;CLEAR INT. ENABLE
;      (PREVENT OVER-RUNS)
VLOOP:
;12.  TEST  SSDCTL,002H ;CHECK FOR TRANSMIT
;      READY/EMPTY FLAG
;      JZ  VLOOP ;WAIT FOR PTY/READY
;13. A. LODS  BYTE PTR [SI] ;GET CODEC DATA
;13. B. MOV  SSDDAT,AL ;OUTPUT DATA TO
;      SSDA/CODEC
;14.  LOOP  VLOOP ;OUTPUT TIL DONE
;15.  STI ;SET INT. ENABLE
;16.  POP  ES ;RESTORE ES
;17.  RET  8 ;RETURN PAST 4
;      PARAMS
VOUT  ENDP
CSEG  ENDS

```

-----

END VOUT

-----

VINIT

PURPOSE:

THIS SUBROUTINE ALLOWS A BASIC OR ASSEMBLY LANGUAGE PROGRAM TO INITIALIZE THE CODEC SUBSYSTEM PRIOR TO SOUND GENERATION. THE INTERFACE TO THIS ROUTINE IS THROUGH THE CALL STATEMENT:

100 CALL VINIT

ON ENTRY:

SYSTEM INITIALIZED BY MSDOS BOOT.

PROCESSING:

1. DISABLE INTERRUPTS.
2. SAVE ES.
3. SET ES TO I/O SEGMENT BASE AT E800H.
4. INITIALIZE 6852 SSDA:
  - A. SET AC2/1 IN C1 TO 0 TO SELECT C2 ON NEXT WRITE.
  - B. WRITE C2 CONFIGURATION DATA (5EH) TO SELECT NO ERR INT., SEND SYNC ON TUF, 8 BITS-NO PARITY, 1 BYTE XFER, AND DTR TO 0 (OUT).
  - C. SET AC2/1 IN C1 TO 01 TO SELECT C3 ON

NEXT WRITE.

- D. WRITE C3 CONFIGURATION DATA (0DH) TO SELECT CLEAR TUF, CLEAR CTS, 2 SYNCs, AND EXTERNAL SYNC MODE.
- E. SET AC2/1 IN C1 TO 10 TO SELECT THE SYNC REGISTER ON NEXT WRITE.
- F. WRITE THE CODEC SILENCE PATTERN (AAH) INTO THE SYNC REGISTER.

5. INITIALIZE THE 6522 VIA CODEC CLOCK GENERATOR:

- A. ADD FLAGS (0DH) TO ACR REGISTER TO SELECT TIMER 1 IN FREE RUNNING MODE AND OUTPUT TO PB7.
- B. WRITE THE TIMER 1 LATCH VALUE (20H) TO GENERATE A 16KHZ CODEC CLOCK RATE.

6. INITIALIZE THE 6522 VIA VOLUME CONTROL:

- A. CHANGE THE ACR FLAGS (B2-4=100) TO SELECT SHIFT REGISTER SHIFT OUT AT FREE-RUNNING TIMER 2 RATE.
- B. SET THE SHIFT REGISTER FOR MAXIMUM VOLUME (0).
- C. SET THE VOLUME TIMER 2 TO 1 IN ORDER TO PRODUCE 20KHZ AT CB2.

7. RESTORE ES.

8. RETURN.

ON EXIT:

CODEC INITIALIZED AND QUIETLY SENDING SYNCs (AA) DUE TO UNDER-RUN.

-----

### ASSEMBLY CODE

Assembly code for VINIT follows:

```

PUBLIC VINIT
NAME VINIT
CSEG SEGMENT 'CODE'
CSEG ENDS
DSEG SEGMENT PUBLIC 'DATA'
DSEG ENDS
SSEG SEGMENT STACK 'STACK'
SSEG ENDS
ESEG SEGMENT AT 0E800H
ORG 028H ;6522 VIA BASE
VOLCLK DW ? ;CODEC CLOCK/SHIFT REG
ORG 02AH ;6522 VIA BASE

```

## New User's Corner/

### H. PAUL SCHMIELEWSKI

For many of you who are reading this, your Victor computers are old and reliable friends with whom you have shared countless hours of working together on a variety of projects with mixed results. There are many stories to be told and between all of you there is a wealth of information to be shared.

As a newcomer to the Victor 9000 computer I have already met and talked with many of you and I am grateful for all the help. Not only is the Victor new to me, I am learning MSDOS after working only with 8-bit computers (mostly the Commodore 64, C+4, and 128). After recently crashing my hard drive, the value or a little help from your fellow computer users was more than evident. Andrew (Rudiak) missed a good portion of the meeting helping me with the HDSETUP and Franz and Carol (Hirner) stayed late to add further assistance. Al (Bullock) offered his help before he left and Boardman (Moore) spent a couple of hours sharing some tips on how to work within my 256K Victor and demonstrated the abilities of an upgraded version when I brought my system to him for help.

I see my Victor as a learning machine that will allow me to explore the productivity software that was previously not available to me on an 8-bit system. Wordstar, Lotus, dBase, Multiplan, Wordperfect and many of the programs I have read about and researched I can now experience. The games and sound features of my Commodore 128 along with my Q-Link membership and the networking remain for entertainment while I practice and learn serious computing on the Victor 9000. Hard disk management is another area that is fascinating due to the speed and availability not to be found on 8-bit machines.

After receiving my first VictorTalk and having had the opportunity to meet other members at a recent meeting, along with the assistance I have received in person or over the phone from so many of the members, I am very pleased to be a member of the Bay Area Victor User's Group and look forward to mastering a whole new world of computing that this computer has made available to me. From time to time, I will continue to have questions that may seem simple or even irrelevant to you. Thanks for your patience as I graduate from the world of 8-bit to 16-bit computers.


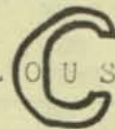
I am concerned about the lack of Victor specific software, but am reassured that conversions to Victor format can be easily done by fellow members, or at the meetings. I am not too concerned about repairs, since I know that fine technical assistance is available within the group and from what I've seen will seldom be needed, as the Victor is one of the best constructed (but most poorly merchandised) computers available this decade.

```
VOLUME DB ? ;VOLUME SHIFT REG
VOLACR DB ? ;ACR TO SET MODE
        CRG 060H ;6852 BASE AT E8060
SSDCTL DB ? ;6852 SSDA CTL
SSDAT  DB ? ;6852 SSDA DATA
        CRG 084h ;6522 VIA BASE
CDCLK  DW ? ;TIMER 1 FOR CLOCK
        CRG 08BH ;ACR FOR TIMER MODE
ACRCLK DB ? ;...
ESEG   ENDS
CSGRP  GROUP CSEG
DSGRP  GROUP DSEG,SSEG
SSGRP  GROUP SSEG
ESGRP  GROUP ESEG
;-----
CSEG   SEGMENT
        ASSUME
CS:CSGRP,DS:DSGRP,ES:ESGRP,SS:SSEG
;-----
; VINIT (Note: Do NOT type in line numbers
;-----
;-----
VINIT  PROC FAR
; 1. CLI ;CLEAR INTERRUPTS
; 2.A PUSH DS ;SAVE DS
; 2.B PUSH ES ;SAVE ES
; 3. MOV AX,ESEG ;SET UP ES
; MOV ES,AX ;TO IOSEG BASE
; 4.A MOV SSDCTL,0 ;SET AC2/1 TO 01
; 4.B MOV SSDAT,5EH ;WRITE C2 CONFIG
; 4.C MOV SSDCTL,40H ;SET AC2/1 TO 01
; 4.D MOV SSDAT,00H ;WRITE C3 CONFIG
; 4.E MOV SSDCTL,80H ;SET AC2/1 TO 10
; 4.F MOV SSDAT,0AAH ;WRITE SILENT SYNC
; 5.A CR ACRCLK,0C0H
; 5.B MOV CDCLK,20H ;WRITE TIMER 2 VALUE
; 6.A MOV AL,VOLACR ;GET CURRENT ACR
; AND AL,NOT 1GH ;STRIP OUT B2-4
; CR AL,10H ;INSERT NEW B2-4
; MOV VOLACR,AL ;UPDATE ACR
; 6.B MOV VOLUME,0 ;SET S/R MAX VOLUME
; 6.C MOV VOLCLK,1 ;SET VOLUME CLOCK
; 7.A POP ES
; 7.B POP DS
; 8. RET
VINIT  ENDP
CSEG   ENDS
;-----
END VINIT
```

\* \* \* \* \*

# Pop Quiz

See if you've got what it takes to be a *VictorTalk* reader (besides \$35). Translate the following. Hints: #1 = Sandbox. #24 = Just between you and me. Answers next issue.

<p>[1] <i>Sandbox</i></p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">SAND</div>	<p>[2] <i>Man on Board</i></p> <p style="text-align: center;">MAN — BOARD</p>	<p>[3]</p> <p style="text-align: center;">STAND — I</p> <p><i>I understand</i></p>	<p>[4] <i>one letter at a time</i></p> <p style="text-align: center;">R   E   A   D   I   N   G</p>
<p>[5] <i>long underwear</i></p> <p style="text-align: center;">WEAR — LONG</p>	<p>[6] <i>crossroads</i></p> <p style="text-align: center;">R ROADS A D S</p>	<p>[7] <i>down town</i></p> <p style="text-align: center;">T O W N</p>	<p>[8] <i>Tricycle</i></p> <p style="text-align: center;">CYCLE CYCLE CYCLE</p>
<p>[9] CHAIR</p> <p><i>high chair</i></p>	<p>[10]</p> <p style="text-align: center;">DICE DICE</p> <p><i>pair of dice</i></p>	<p>[11]</p> <p style="text-align: center;">T O U C H</p> <p><i>touch down</i></p>	<p>[12]</p> <p style="text-align: center;">GROUND</p> <p style="text-align: center;">Feet Feet Feet Feet Feet Feet</p> <p><i>my feet under ground</i></p>
<p>[13] <i>not level</i></p> <p style="text-align: center;">LE VEL</p>	<p>[14]</p> <p style="text-align: center;">0</p> <p><i>3 layers behind zero</i></p> <p style="text-align: center;">M.D. Ph.D. B.S.</p>	<p>[15]</p> <p style="text-align: center;">KNEE LIGHTS</p>	<p>[16]</p> <p style="text-align: center;">I I I I</p> <div style="text-align: center;">  </div>
<p>[17] <i>mind over matter</i></p> <p style="text-align: center;">MIND — MATTER</p>	<p>[18] <i>He's higher than himself</i></p> <p style="text-align: center;">HE'S — HIMSELF</p>	<p>[19] <i>Backward glance</i></p> <p style="text-align: center;">ECNALG</p>	<p>[20]</p> <p style="text-align: center;">DEATH LIFE</p>
<p>[21]</p> <p style="text-align: center;">G.I.</p> <p style="text-align: center;">C C C C C C</p>	<p>[22] <i>Blank program</i></p> <p style="text-align: center;">PROGRAM</p>	<p>[23] <i>see through house</i></p> <p style="text-align: center;">B L O U S E</p> <div style="text-align: center;">  </div>	<p>[24] <i>Just between you and me</i></p> <p style="text-align: center;">J U S T YOU ME</p>

# Issue News

[The following is a word-for-word transcription of the March 1989 Issue News from Holland. The current exchange rate for Dutch Guilders is about \$.45, should you order anything. - Ed.]

This edition of the **Issue newsletter** for Sirius and Vicki users is dedicated to the documentation of various bugs and how to get around them. It is hoped that the (sometimes technical) descriptions will give an insight into the mysteries of the excellent Sirius architecture and satisfy some of the thirst for technical information expressed to us in the recent past.

## BUGS AND FIXES

Issue's best bug yet was with release 1.0 of NBI, the pop-up software which came with the Clock+ Card. On or after the 13th of any month it refused to believe that the clock chip had been started and would only display a Clock+ instruction screen. NBI has now been reincarnated as FLASH with five separate pop-up windows including binary/hex/decimal/ASCII conversion - the new Clock+ Card disc has Issue DOS, FLASH and ASM source files donated by users. Release history of RAMDISC: first general release 1.4, RD reset request verify added with 1.5, improved command format 1.6, RD survives machine reset 1.78, improved reinstallation after machine reset 1.8 and improved RD reset with 1.9. Starpatch is now on version 1.2 due to the occasional appearance of an extra cursor, now solved, and automatically patches both Wordstar 3.3 and 3.21 for direct, high speed screen access. WordPerfect is in high fashion at the moment with 4.2 available to Sirius users with lots of money and memory but we're sticking to Wordstar, the WP you either love or love to hate.

If you've ever overwritten a disc directory, rendering the disc completely useless, you might be interested to learn how this comes about. It's Victor and Microsoft's fault for BIOS versions 290-292 and Microsoft alone's thereafter, at least as best as we can gather. MS-DOS normally keeps a copy of the disc directory in RAM to minimize disc accesses and this is updated and written back to the disc as and when appropriate. Early BIOS versions did not check the drive door often enough to see if the disc may have changed and DOS itself has an ambiguous "Don't Know" status of the Media Check Byte. (CP/M got around this inherent difficulty by "logging" a new disc with ALT-C). The problem can occur on any machine running MS-DOS/PC-DOS. Changing discs during an Abort, Retry, Ignore? message particularly is not recommended, and BIOS 293 (or later) is the one to use. We know of at least six different Operating Systems on the Sirius - MS-DOS 1.25, 2.11 and 3.1, CP/M 86, CP/M 80 (for Z80 card) and the UCS0 p-system. We've yet to see the latter.

The Super BIOS is a sort of unfriendly user interface to the BIOS which is called with INT DF (hex). One of the functions

which doesn't work fully is function 0 (get BIOS Version) and we currently check for the ability to reset from the keyboard with CAPS LOCK+ALT+NUMKEY DEC,PT to establish whether the system is running BIOS 2.93 or later. The new PD/2 public domain disc features KERMIT and as much documentation as we've been able to cram, with specific information for Vicki and VI users.

No doubt there are more bugs in the BIOS but ones we are aware of are that the console output interrupt MOV AH, as char INT 29h is unreliable when used within loops, although it works OK sending individual characters like in escape sequences (this may be a re-entrancy problem). One that caused us some grief (because we thought at first it was a ROM bug) concerns high resolution graphics displays using the DOS 3 protocol for reserving the 40K bit-mapped screen image. One of the few hardware limitations of the Sirius is that this 40K memory must reside wholly within the first or second 64K segment. Loading a big Operating System (DOS 3) and/or a very large TSR can result in this memory no longer being available for graphics. Victor's solution with DOS 3 is to reserve the graphics memory with GETSCRN early on booting and set a 4-byte flag at 0:364h, while the flag can be cleared and the memory released with KILLSCRN. Unfortunately, the BIOS did not reset this flag on boot-up, relying on ROM code to do so. Since Issue ROM does not destroy memory contents (permitting RAMDISC survival and other marvels) this could cause a recurrent problem if a reset was made during a graphics application. Graphics software following the DOS 3 protocol will always run happily if using either Issue ROMs with fast memory test on re-boot or the updated Issue DOS/BIOS 2.94. However, the recommended method for implementing graphics on the Sirius and Vicki is to ignore the DOS 3 protocol whenever the memory can be grabbed without difficulty and release it on exit, else leave flag-setting and memory reservation to GETSCRN or a similar utility.

Besides an early bug in IBM's FORMAT which defaulted to the hard disc and went ahead even after hitting reset, the last word on software/firmware bugs has to go to one in a (thankfully non-nuclear) guided torpedo. Said torpedo had a "fail-safe" routine such that if it turned 180° after firing, it was assumed to be out of control and would self-detonate. During naval exercises one of these intelligent weapons of destruction misfired and got stuck in the torpedo tube, whereupon the Captain turned 180° to head for home. You can guess what happened next...

ISSUE Postbus 3707 1001 AM AMSTERDAM  
THE NETHERLANDS  
ISSUE P.O. BOX 222 BRIGHTON BN1 3BR UK

**This issue's contributors:**

**Issue Products for the Victor Sirius/V9000/S1, VI and Vicki  
International Price List  
June 1988**

The Issue Boot ROMs for the Victor Sirius are 100% compatible with the latest Victor ROMs but with full on-screen diagnostics and other features **HFL 125.00**

The Issue RAMDISC software provides an additional disc drive of programmable size for reduced risk of disc corruption and much faster operating speed. With DOS **HFL 150.00**

INDICATORS is a set of three memory-resident Caps Lock indicator programs, particularly suitable for use with word-processors. With DOS **HFL 75.00**

The Issue Clock+ (Clock Plus) Card is a real-time clock for the Sirius and VI with additional features. With DOS and 'pop-up' software **HFL 265.00**

The Issue 512 RAM Board is an advanced design memory board using the latest technology for low power consumption and full performance. With DOS and RAMDISC **HFL 990.00**

512K RAM Board, as above, with software but without memory chips (tested) **HFL 500.00**

The Issue 8087 Board enables an Intel Numeric Co-processor to be used in the Sirius, VI and Vicki for high speed "number crunching" and scientific applications **HFL 750.00**

SOFTWARE UPDATES on return of the original master disc(s) accompanied by **HFL 25.00**

ISSUE PUBLIC DOMAIN DISC **HFL 30.00**

The DOS provided with several of the above items is a modified and improved version of MS-DOS 2.11/BIOS 2.93. State HARD/FLOPPY/COUNTRY. Prices are in Dutch Guilders and are fully inclusive, but may not include local taxes. Payment by Eurocheque(s), Bankers' Draft or Post Office International Money Order should normally accompany orders. All items shown are available ex-stock and carry a full warranty.

Hugh Connelly  
13441 Pt. Pleasant Drive  
Chantilly, VA 22021  
(703) 378-1912

Dan Mardesich  
1569 Fourth Street  
San Rafael, CA 94901  
(415) 454-6661

CHANNEL 9000  
James Leshar  
(no longer published)

Paul Schmielewski  
3516 Quigley Street  
Oakland, CA 94619  
(415) 531-9661

\* \* \* \* \*

**VictorTalk**

**Address inquiries to:**

BAY AREA VICTOR USERS GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

**Subscriptions:** Carol Hirner

**Editor:** George Anderson  
782 Jordan Ave.  
Los Altos, CA 94022  
(415) 941-3029

**Layout:** Andrew Rudiak

**VictorTalk** is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in **VictorTalk**. Copyright 1988 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

**VictorTalk** is an independent publication not affiliated in any way with Victor Technologies, Inc. **Victor** is a registered trademark of Victor Technologies, Inc.

**Submit articles** on MSDOS disks in ASCII or Spellbinder format (either single-sided or double-sided, in Victor or IBM format), with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

BBS is on line 24 hours a day: (415) 574-7914, Franz Hirner, Sysop.

**Victor\*GROUP meetings** are normally held at 7:30 P.M. on the first Wednesday of the month at the Holiday Inn, Powell Street, Emeryville, CA (adjacent to Highway I-80).



PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
City _____ State _____ & Zip _____	
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Un-archived versions are available where "A", "AB", or "ABC" appears below number. If originally released (archived) version is desired, cross out corresponding "A", "AB", or "ABC". (Original releases of all other disks contain no archived files.) Note that pricing below is per "disk number" rather than per "disk" (re-released unarchived "disk numbers" may be comprised of one disk (A), two disks (AB), or three disks (ABC). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No. Disks	Vol. No. Disks	Vol. No. Disks	Vol. No. Disks
4 2 61-64	5 1 78-81	5 5 87-90	6 3 96-97
4 3 65-71	5 2 82-84	5 6 91-92	6 4 98
4 6 72-77	5 4 85-86	6 2 93-95	6 5 99

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56*	57*	--	59	60
													A						AB
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
												A					AB	AB	AB
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	101
AB	AB	AB	AB	AB	AB	AB	ABC	ABC	AB	AB	ABC	ABC	AB	ABC	AB	AB			

\* 34 & 35 are a two-disk set; 56 & 57 are also a two-disk set

1988 Victor\*Group member: No  Yes  Date joined/renewed \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, 1988, sets of 6, \$12.00 /set.. \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1989 membership fee \$35 (receive all 1989 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

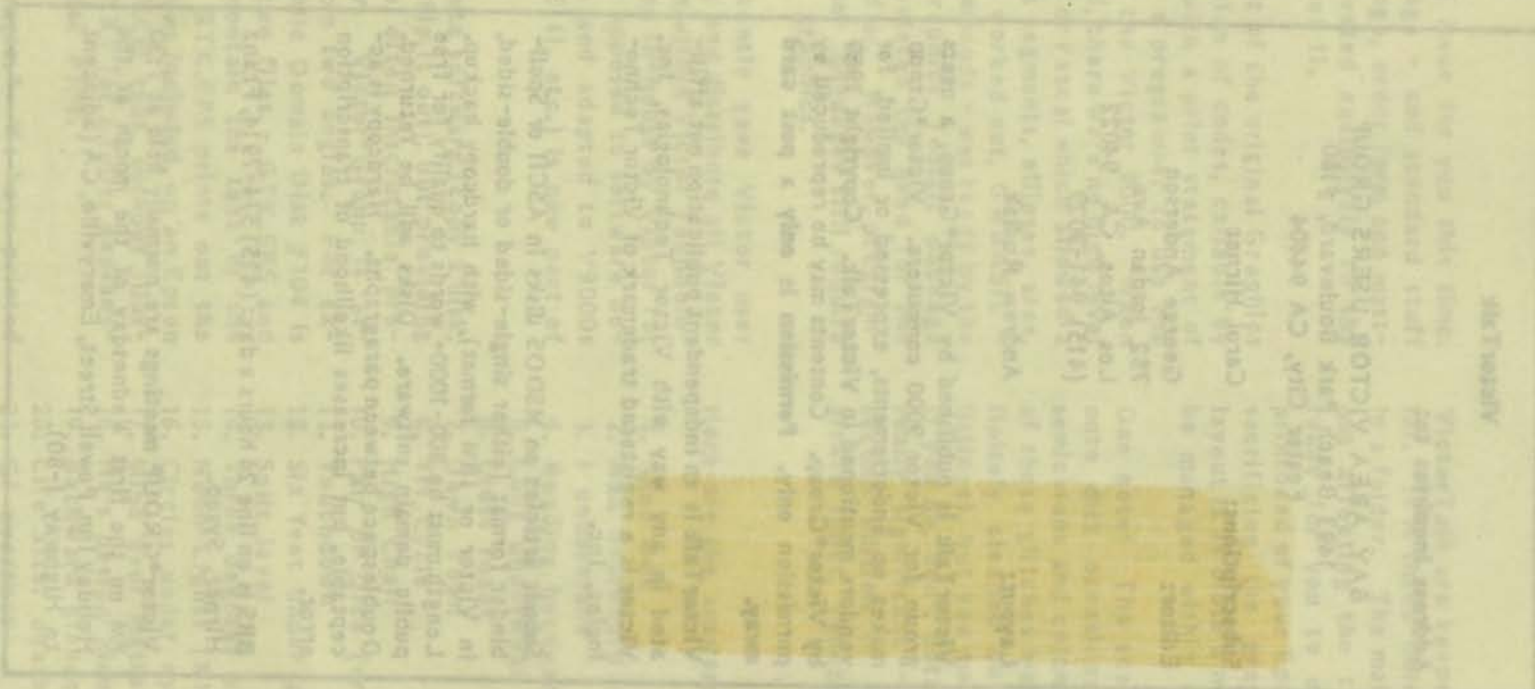
TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_

# VICTOR\*GROUP

1493 Beach Park Blvd., No. 180

Foster City, CA 94404

BULK RATE  
U.S. POSTAGE  
PAID  
LAFAYETTE, CA 94549  
PERMIT NO. 39



PRINTED MATTER

VICTOR TALK

## VictorTalk

### Address inquiries to:

BAY AREA VICTOR USERS GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

**Subscriptions:** Carol Hirner

**Editor:** George Anderson  
782 Jordan Ave.  
Los Altos, CA 94022  
(415) 941-3029

**Layout:** Andrew Rudiak

**VictorTalk** is published by Victor\*Group, a users group for Victor 9000 computers. Victor\*Group makes no endorsements, expressed or implied, for vendors mentioned in **VictorTalk**. Copyright 1989 by Victor\*Group. Contents may be reproduced by permission only. Permission is only a post card away.

**VictorTalk** is an independent publication not affiliated in any way with Victor Technologies, Inc. **Victor** is a registered trademark of Victor Technologies, Inc.

**Submit articles** on MSDOS disks in ASCII or Spellbinder format (either single-sided or double-sided, in Victor or IBM format), with hardcopy backup. Length must be 800-1000+ words to qualify for free public domain software. Disks will be returned. Double-space between paragraphs. Hardcopy is acceptable, but increases likelihood of transcription error.

**BBS** is on line 24 hours a day: (415) 574-7914, Franz Hirner, Sysop.

**Victor\*GROUP meetings** are normally held at 7:30 PM on the first Wednesday of the month at the Holiday Inn, Powell Street, Emeryville, CA (adjacent to Highway I-80).

# VICTOR<sup>®</sup> TALK

Bi-Monthly Magazine for Victor 9000 users

Volume 7, Number 2, May-June 1989

**W**e have an unusual issue for you this time. Except for this page - and standard stuff such as the mailcover, logo, order form and masthead - this issue is devoted exclusively to the manual for Graftix Kernel II.

The Graftix Kernel is part of the original Graphics Toolkit. It is a collection of codes, callable by any computer language with a print statement of some sort, which enable a programmer to use the graphics capabilities of the Victor 9000. When the Kernel was developed, there were no high level languages such as BASIC or Pascal which allowed a programmer to draw lines segments, ellipses, arcs, and the like. As things worked out, GWBASIC - which contained such commands - was ready by the time Victor put the Graphics Toolkit package together and largely obviated the need for the Graftix Kernel for many of us. However, programmers working in most other languages still require the Kernel to generate graphics.

Victor Technologies recently gave Victor user groups permission to copy and distribute all Victor software originating from within Victor Technologies. Software developed in-house by Victor for its V9000 computers (as opposed to software developed by third parties and adapted to V9000s such as MS-DOS, dBASE II, etc.) is now part of Victor public domain.

It's easy for us to distribute software (copy a disk, stuff it in a mailer and ship it out), but providing documentation for these often inscrutable programs is another matter. Victor\*Group is attacking this problem from two angles. For example, Victor\*Group Public Domain Disk #104 is devoted exclusively to two files - GTK.EPS and GTK.ASC. Both files contain the text of the Graftix Kernel manual. GTK.EPS prints out the manual exactly, **illustrations included**, on an Epson printer. GTK.ASC is ASCII text, without illustrations. See DISK104.TXT for a fuller explanation of the nature of disk #104.

This issue is the other "angle." We have printed GTK.ASC for you, with slightly enhanced formatting for clarity, plus illustrations photocopied directly from the original manual. All issues of

**VictorTalk** are keepers, but this one is in particular. It is three-hole punched so you can store it in a folder. The mail cover can be removed without affecting the text. It's neatly bound and ready for you to use, without burning out the printhead of your Epson and your ears, and without sacrificing the graphic illustrations. And the layout is as close a match for the original as could be managed within our 8.5"x11" format.

One note: The scanner used to read GTK.ASC onto disk occasionally confused "j"s and "l"s, semicolons and colons, periods and commas. Most of these glitches were eliminated (they were definitely cleaned out of the BASIC and Pascal listings), but you may run across a few we missed. If you do, don't bother telling us unless they are **really** significant - there ain't going to be a second edition of this baby. We may run an errata sheet, though, if absolutely necessary.

\*\*\*\*\*

Here are the answers to the quiz in Vol. 7, No. 2:

1. Sand Box
2. Man overboard
3. I understand
4. Reading between the lines
5. Long underwear
6. Cross roads
7. Down Town
8. Tri Cycle
9. High Chair
10. Pair of Dice (or paradise)
11. Touchdown
12. Six feet under ground
13. Split level
14. Three degrees below zero
15. Neon Lights (KNEE on LIGHTS)
16. Circles under eyes
17. Mind over matter
18. He's beside himself
19. Backward glance
20. Life after death
21. GI overseas
22. Space program
23. See-through blouse
24. Just between you and me.

# NOTES

VictorTalk, Vol. 7, No. 3, May-Jun 1989

VictorTalk, Vol. 7, No. 3, May-Jun 1989

VictorTalk is a report, but this one is so different. The information is presented as you can store it in a folder. The only caveat can be removed with the exception of the text. It's really hard and easy to use. It's without having to be a member of your own club and you can, and without revealing the source. It's really hard and easy to use. It's without having to be a member of your own club and you can, and without revealing the source. It's really hard and easy to use. It's without having to be a member of your own club and you can, and without revealing the source.

One more. The reason you can use it is that it's really hard and easy to use. It's without having to be a member of your own club and you can, and without revealing the source. It's really hard and easy to use. It's without having to be a member of your own club and you can, and without revealing the source.

1. VictorTalk
2. VictorTalk
3. VictorTalk
4. VictorTalk
5. VictorTalk
6. VictorTalk
7. VictorTalk
8. VictorTalk
9. VictorTalk
10. VictorTalk
11. VictorTalk
12. VictorTalk
13. VictorTalk
14. VictorTalk
15. VictorTalk
16. VictorTalk
17. VictorTalk
18. VictorTalk
19. VictorTalk
20. VictorTalk
21. VictorTalk
22. VictorTalk
23. VictorTalk
24. VictorTalk
25. VictorTalk
26. VictorTalk
27. VictorTalk
28. VictorTalk
29. VictorTalk
30. VictorTalk

VictorTalk is a report, but this one is so different. The information is presented as you can store it in a folder. The only caveat can be removed with the exception of the text. It's really hard and easy to use. It's without having to be a member of your own club and you can, and without revealing the source. It's really hard and easy to use. It's without having to be a member of your own club and you can, and without revealing the source.

One more. The reason you can use it is that it's really hard and easy to use. It's without having to be a member of your own club and you can, and without revealing the source. It's really hard and easy to use. It's without having to be a member of your own club and you can, and without revealing the source.

1. VictorTalk
2. VictorTalk
3. VictorTalk
4. VictorTalk
5. VictorTalk
6. VictorTalk
7. VictorTalk
8. VictorTalk
9. VictorTalk
10. VictorTalk
11. VictorTalk
12. VictorTalk
13. VictorTalk
14. VictorTalk
15. VictorTalk
16. VictorTalk
17. VictorTalk
18. VictorTalk
19. VictorTalk
20. VictorTalk
21. VictorTalk
22. VictorTalk
23. VictorTalk
24. VictorTalk
25. VictorTalk
26. VictorTalk
27. VictorTalk
28. VictorTalk
29. VictorTalk
30. VictorTalk

# GRAFIX

## Reference Manual

Transcribed May 7, 1989  
by Franz Hirner

**COPYRIGHT** ©1983 by VICTOR

All rights reserved. This manual contains proprietary information which is protected by copyright. No part of this manual may be reproduced, transcribed, stored in a retrieval system, translated into any language or computer language, or transmitted in any form whatsoever without the prior written consent of the publisher. For information contact:

VICTOR Publications  
380 El Pueblo Road  
Scotts Valley, California 95066  
(408) 438-6680

### TRADEMARKS

VICTOR is a registered trademark of Victor Technologies, Inc.  
GRAFIX is a trademark of Victor Technologies, Inc.

### NOTICE

VICTOR makes no representations or warranties of any kind whatsoever with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. VICTOR shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing performance, or use of this material.

VICTOR reserves the right to revise this publication from time to time and to make changes in the content hereof without obligation to notify any person of such revision or changes.

Second VICTOR printing April, 1984.

ISBN 0-88182-102-0

## CONTENTS

1. Introduction
2. Memory Requirements
3. Installing GRAFIX
  - 3.1 Character Set Parameter
  - 3.2 Screen Parameter.
4. Using High Resolution Graphics
  - 4.1 The High Resolution Screen
    - 4.1.1 The Display Screen and The Work Screen
    - 4.1.2 The Text Window
    - 4.1.3 The Aspect Ratio
  - 4.2 The Character Set
    - 4.2.1 Printing Vertically
5. GRAFIX Commands
  - 5.1 Parameter input Commands
  - 5.2 Parameter Return Commands
  - 5.3 Action Commands
  - 5.4 File Maintenance Commands
  - 5.5 Using The GRAFIX Escape Sequences
  - 5.6 Command Descriptions
6. Hi-Res Printing
  - 6.1 Character Sets
  - 6.2 The Hi-Res Print Function
  - 6.3 Proportional Printing
7. The Cursor
  - 7.1 Positioning the Cursor
  - 7.2 Pointing The Cursor
  - 7.3 Multiple-Cursor Problems
8. Using Windows and Screens
  - 8.1 Creating a Window
  - 8.2 Moving a Window
  - 8.3 Automatic Clipping
  - 8.4 Saving Time
  - 8.5 Using Screens

- 9. Using the Combination Rules
  - 9.1 Exchange of Screens
  - 9.2 Blend or Fade Effect
  - 9.3 Hi-Res Printing
  - 9.4 Line Draw Functions
  - 9.5 The Combination Rules

**APPENDIXES**

- A. Escape and Control Sequences
- B. Error Messages
- C. GRAFIX Functions Reference List

**FIGURES**

- 4-1: Screen Coordinates
- 4-2: The Aspect Ratio
- 4-3: Character Orientation
- 7-1: Total Cursor Space
- 8-1: Moving a Window
- 9-1: Combination Rules
- A-1: Window Redefinition Parameter

**TABLES**

- 2-1: Approximate User Space with GRAFIX
- 7-1: Targets of GRAFIX Cursors

---

---

**CHAPTERS**

- 1. Introduction
- 2. Memory Requirements
- 3. Installing GRAFIX
- 4. Using High Resolution Graphics
- 5. GRAFIX Commands
- 6. Hi-Res Printing
- 7. The Cursor
- 8. Using Windows and Screens
- 9. Using the Combination Rules

Appendix A: Escape and Control Sequences

Appendix B: Error Messages

Appendix C: GRAFIX Functions Reference

NOTES

Reference Manual

GRAFIX

CONTENTS

## INTRODUCTION

GRAFIX gives you a powerful set of commands to use the high resolution screen fully. GRAFIX provides such functions as:

- \* Drawing of multiple line types and widths
- \* Drawing of circles and arcs
- \* Filling of regions and bars
- \* Definition of cursor and fill patterns
- \* Logical combination of screens and windows
- \* Columnar printing

Screen file routines let you create screens or portions of screens, and then save them on disk. You can call up screen files whenever you need them. Furthermore, character sets created with the EFONT program can be read from disk and used in various modes, including double-size, reverse Video, and proportional printing.

Since high resolution graphics requires a substantial amount of memory, the standard memory configuration of 128K bytes might not be sufficient for your needs. Refer to Chapter 2, "Memory Requirements", to see if you need additional memory for your application.

# 1

## MEMORY REQUIREMENTS

# 2

With GRAFIX you can use up to ten different character sets of 128 characters each, and up to eight full screens. A character set uses slightly over 4K bytes: a screen uses exactly 40,000 bytes.

The specific number of screens that you can use is determined by the total amount of available memory in the system. Table 2-1 gives some examples which assume that you are using four character sets.

Table 2-1: Approximate User Space with GRAFIX

SYSTEM RAM CAPACITY	SPACE AVAILABLE TO USER			
	128K	256K	384K	512K
No. of Screens				
1	32K	180K	288K	416K
2	-	141K	249K	377K
3	-	102K	210K	338K
4	-	63K	171K	299K
5	-	23K	132K	260K
6	-	-	93K	221K
7	-	-	43K	182K
8	-	-	14K	141K

NOTE: The Above Values represent memory available for the user after allocating memory to the operating system (30K), GRAFIX (20K), and four character sets (4K each).

In a system with 512K bytes of RAM, you can use all ten character sets and all eight screens; 128K of RAM remain for your own use. For smaller RAM configurations, you must decrease the number of screens and character sets by an appropriate amount. In an application, you must ensure that enough memory is left in RAM for both your program and your runtime package.

You can use GRAFIX with any language that is capable of executing a print statement.

# 3



# INSTALLING GRAFIX

## 3

To install GRAFIX, type **GRAFIX**, followed by a space, a dollar sign, and the letter **S**. Then type the number of screens you want to use, the letter **C**, and the number of extra character sets you want to activate.

To install GRAFIX with two character sets and two screens, for example, type the following command:

```
grafix $S2C2
```

To specify a printer, enter a third parameter, **P**, and one of the following:

```
F or M  Epson
T       Tally
C or S  C. Itoh
O       Okidata
N       No printer
```

For example, to install GRAFIX with one character set, three screens, and the printer driver for an Epson printer, type the following command:

```
grafix $S3C1PE
```

If you type the GRAFIX command without any parameters, you install GRAFIX with only one screen and the standard system character sets. Once GRAFIX is installed, you must reboot the system to return to any other mode.

---

### 3.1 CHARACTER SET PARAMETER

The minimum number of character sets available is two. The first 128 ASCII characters and the upper 128 graphics characters are the two system character sets; neither of these sets is ever overwritten. The maximum number of character sets that you can specify is eight (a total of ten, including the two system character sets).

If you specify one character set (for a total of three) you can load only one additional character set at a time. Each time you load a new character set, the previously loaded character set is overwritten.

The only reason to specify an additional character set is to minimize disk access. If an application requires many different character sets, you can load all the sets once. Then the sets are resident in RAM, and you can access them at any time without repeatedly loading from disk.

### 3.2 SCREEN PARAMETER

The minimum number of screens you can specify is 1, and the maximum is 8. Refer to Table 2-1 to select the number of screens.

MEMORIAL KEONIKEMENIS

# USING HIGH RESOLUTION GRAFIX

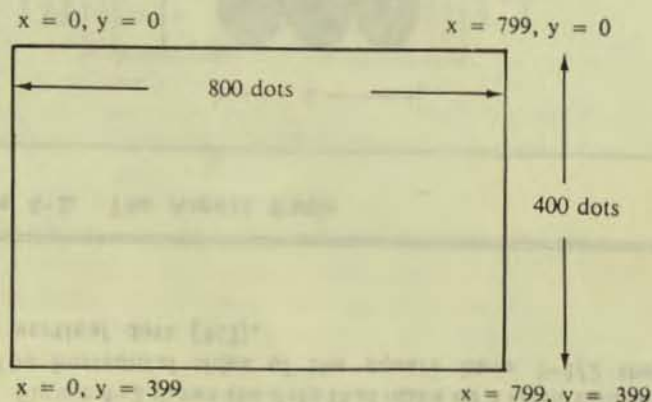
## 4

### 4.1 THE HIGH RESOLUTION SCREEN

The high resolution screen consists of 320,000 individual dots. There are 800 dots in each row and 400 in each column.

You can describe any point on the screen by using vertical (y) coordinates. The coordinates range 0,0 in the upper left corner to 799,399 in the lower right corner. Figure 4.1 displays the coordinates of the corners of the screen.

Figure 4.1: *Screen Coordinates*



### 4.1.1 THE DISPLAY SCREEN AND THE WORK SCREEN

There are two types of screens in GRAFIX.

- DISPLAY SCREEN--currently displayed on the video screen
- WORK SCREEN--where you send GRAFIX commands

If you make the Display Screen and the Work Screen the same screen then the effects of the GRAFIX commands appear on the video screen. If they are not the same, the effects of the commands do not appear on the Display Screen until the Display Screen number is switched to match the Work Screen number or vice versa. Regardless of how many screens you install when booting GRAFIX you can have only one Display Screen and one Work Screen. But you can easily reassign Display Screen or Work Screen status to any other screen. The escape sequence used to change the Display Screen is **Esc 5 B**. The sequence used to change the Work Screen is **ESC 5 A**.

### 4.1.2 THE TEXT WINDOW

With this feature, you can define a text window with any number of lines anywhere on the screen. The text window simulates normal VT52 terminal operation under Hi-Res and implements most of the normal functions and their escape sequences (see Appendix A). The only difference from normal character mode operation is that all cursor positioning, insert, and delete functions refer to the text window rather than to the whole screen. When the bottom of the text window is reached, the text scrolls up. Any normal printout, such as an error message, uses the normal system character set and is directed to the text window. For example, if you print **Esc H** (which normally sets the cursor at Home Position), you will set the cursor to the upper left corner of whatever text window you have defined.

The text window can range from rows 3 to 24: the default is lines 21 to 24 at the bottom of the display. The text window is always on the currently viewed screen. The escape sequence used to specify the range of the text window is **Esc m2 p1 p2** (described in Appendix A).

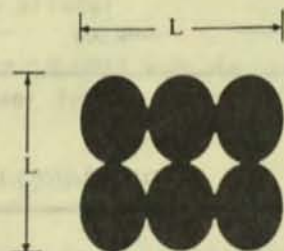
When you debug programs using Hi-Res graphics, do all Hi-Res printing and drawing on a Work Screen (for example, Screen 1) and leave Screen 0 totally in text mode. It is easy to select the Work Screen as the

Display Screen in order to view the Hi-Res graphics. Select Screen 0 as the Display Screen to return to programming. (See Chapter 5 for descriptions of the Select Display Screen and Select Work Screen functions.) Program listings and error messages are printed in the text window in Screen 0: they do not disturb the graphics.

#### 4.1.3 THE ASPECT RATIO

The aspect ratio is the ratio of x units to y units needed to provide equal length in both directions. The video screen has an aspect ratio of 3 to 2. Figure 4-2 shows the dots that make up a small square on the screen. The horizontal sides of the square have 1-1/2 times the number of vertical dots (3:2).

Figure 4-2. The Aspect Ratio



If you want to draw a ten-unit square on the screen, follow these steps: draw a 15-unit line in the x direction, a ten-unit line in the y direction, a 15-unit line in the negative x direction, and finally a ten unit line in the negative y direction. To display a circle on the screen, you must draw an ellipse.

#### 4.2 THE CHARACTER SET

A character set contains 128 characters. Each character is defined within an array which is 10 dots wide and 16 dots high.

The Character Height parameter specifies the height of the character: it is measured in dots starting from the bottom row of the array. This number is constant for all the characters within a character set.

The Character Width parameter specifies the width of the character, measured in dots starting in the leftmost column of the array and moving to the right of a given character set. This parameter can be fixed or can take on a different value for each character.

Each character set file contains the parameters used by GRAFIX to print characters correctly from that character set. The character set file headers contain flags specifying whether a character is:

- Normal or Special
- Horizontal or Vertical
- Fixed-width or Proportional

A normal character set is 10 dots wide and 16 dots long. It is booted at the same time as the system.

Once a character from any character set has been placed on the Hi-Res Screen, the character set doesn't need to remain in RAM to be displayed on the screen. (In normal mode, the character set must reside in RAM to display any character from that set.)

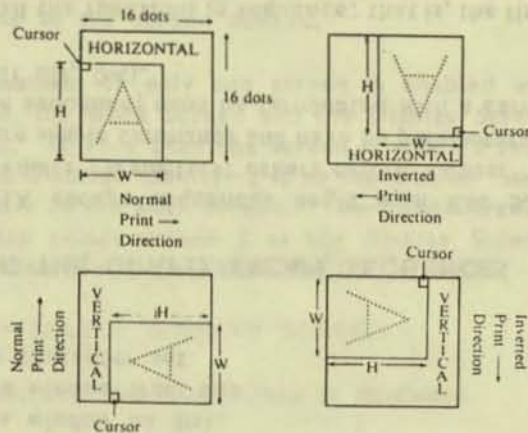
##### 4.2.1 PRINTING VERTICALLY

Because of the screen's 3:2 aspect ratio, you must use different character sets for horizontal and vertical printing. A horizontal character set is normally printed left to right and a vertical character set is normally printed bottom to top.

To print vertically, first specify a vertical character set. Then enter both the Hi-Res print escape sequence and the string of characters to be printed. Printing will be vertical until you return to the horizontal printing.

The portion of the 16-by-16-dot character that is printed, together with the direction of print, is shown in Figure 4-3.

Figure 4.3: Character Orientation



## GRAFIX COMMANDS

5

GRAFIX has the following four types of commands:

1. Parameter Input
2. Parameter Return
3. Action-Screen Draw, Print, or Move
4. File Maintenance

### 5.1 PARAMETER INPUT COMMANDS

- \* Select the screen to draw on
- \* Select the screen to be displayed
- \* Select the fill pattern
- \* Define a screen window
- \* Select a character set
- \* Select a cursor type
- \* Select a combination rule
- \* Set relative and absolute cursor positioning
- \* Set line width
- \* Set line type
- \* Set left margin
- \* Set a dot
- \* Set and reset superscript mode
- \* Set and reset subscript mode
- \* Set and reset invert character and print direction
- \* Set and reset double character size mode
- \* Define text window
- \* Define user cursor
- \* Define user fill pattern
- \* Enable and disable cursor
- \* Enable and disable shadow print
- \* Set and reset reverse video mode
- \* Set and reset underline mode
- \* Set GRAFIX cursor position

### 5.2 PARAMETER RETURN COMMANDS

- \* Get enabled screen number
- \* Get displayed screen number
- \* Get window parameters
- \* Get dot

- \* Get character width
- \* Get character height
- \* Get character type
- \* Get GRAFIX cursor

### 5.3 ACTION COMMANDS

- \* Fill a region
- \* Fill a bar
- \* Draw a circle
- \* Draw an arc
- \* Absolute and relative line draw
- \* Move a window or B screen
- \* Initialize
- \* Hi-Res print
- \* Clear screen
- \* Return GRAFIX cursor to previously saved position
- \* Toggle text window screen

### 5.4 FILE MAINTENANCE COMMAND

- \* Save a window on disk.
- \* Load a window from disk
- \* Select character set

### 5.5 USING THE GRAFIX ESCAPE SEQUENCES

All GRAFIX escape sequences begin with Esc 5. Many of the functions require parameters; others return values. The rest of the functions are simple commands and have no parameters. Any function call (escape sequence) must be terminated with a carriage return/line feed (ASCII 0D, 0A).

You must call the functions in sequence: that is, the first function must end before you call the next one. Note that only integer parameters are accepted.

The following program segments show how each of the four types of GRAFIX functions is used (Parameter Input, Parameter Return, Action and File Maintenance).

---

#### In BASIC

```

10 E$ = CHR$(27) : G$ = E$ + "5" 'initialize grafix call string
20 PRINT G$;"2" 'clear the Hi-Res screen
30 PRINT G$;"Q";400;200 'put cursor at 400,200
40 PRINT G$;"P";100 'draw circle with radius 100
50 PRINT G$;"u" 'get graphics cursor position
60 INPUT X 'GRAFIX will return values
70 INPUT Y 'into the input statements
80 PRINT G$;"iscript" 'load new font from disk (script)
90 PRINT G$;"pEND" 'print to Hi-Res screen
100 END

```

---

#### In Pascal

```

PROCEDURE grafix_example;

CONST
    escape = chr (27);
    grafix = escape * '5';

VAR
    screen_x : 0..799;
    screen_y : 0..399;

BEGIN
    (* all statements perform the same functions as before *)
    writeln(grafix,'2!'); (* Action Command *)
    writeln(grafix,'Q',400,200); (* Parameter Input *)
    writeln(grafix,'P',100); (* Parameter Input *)
    writeln(grafix,'u'); (* Parameter Return *)
    readln(screen_x); (* returned parameter *)
    readln(screen_y); (* returned parameter *)
    writeln(grafix,'iscript!'); (* File Maintenance *)
    writeln(grafix,'pEND. '); (* Parameter Input *)
END (* grafix_example *);

```

## 5.6 COMMAND DESCRIPTION

### ESC 5 A - SELECT WORK SCREEN

**Action:** Selects the screen specified as the Work Screen, All line-draws and Hi-Res prints are directed to the work screen.

**Parameters passed:** Single ASCII numbers between 0 and 7 corresponding to the screen desired.

**Explanation:** If only one screen is enabled when GRAFIX is loaded, the Work Screen and the Display Screen are the same screen. If more than one screen is enabled you can program on Screen 0 (after specifying it as the Display Screen), but select screen 1 as the work screen. Then, once drawing is completed, you can select screen 1 as the Display Screen and view the completed picture, rather than its formation.

### ESC 5 B - SELECT DISPLAY SCREEN

**Action:** The specified screen is displayed.

**Parameters passed:** An ASCII number specifying which screen is to be viewed on the CRT. Current hardware supports selection of Screen 0 or Screen 1.

**Explanation:** As before, if only one screen is specified when GRAFIX is installed, Screen 0 is both the Work and Display Screen at all times. If multiple screens are enabled on installation, Screen 0 or 1 may be selected as the Display Screen.

### ESC 5 C - SET SUPERSCRIPIT SHIFT MODE

**Action:** Shifts the cursor up by the number of dots specified in the currently enabled character set.

**Parameters passed:** None.

**Explanation:** The cursor shifts up a number of dots, usually equal to about one-half the character height stored in the currently enabled character set. This function is used to print superscripts.

### ESC 5 D - RESET SUPERSCRIPIT SHIFT MODE

**Action:** Moves cursor down the number of dots specified in the table for the currently enabled character set.

**Parameters passed:** None.

**Explanation:** This is the complement to Set Superscript Shift Mode. A reset is required to return to printing on the normal line.

### ESC 5 E - SET SUBSCRIPT SHIFT MODE

**Action:** Moves cursor down the number of dots specified in the table for the currently enabled character set.

**Parameters passed:** None.

**Explanation:** Similar to Set Superscript Shift Mode, except that the shift is downward for the printing of subscripts.

### ESC 5 F - RESET SUBSCRIPT MODE

**Action:** Moves cursor up the number of dots specified in the table for the currently enabled character set.

**Parameters passed:** None.

**Explanation:** Similar to Reset Superscript Shift Mode.

### ESC 5 G - SET DOUBLE CHARACTER SIZE MODE

**Action:** Causes all further Hi-Res print to be double the normal size.

**Parameters passed:** None.

**Explanation:** Doubles the width and the height of each character regardless of the character set enabled (a single character dot is blown up into four dots).

### ESC 5 H - RESET DOUBLE CHARACTER SIZE MODE

**Action:** Returns all Hi-Res print to normal size.

**Parameters passed:** None.

#### ESC 5 I - DEFINE SCREEN WINDOW

**Action:** Stores current window dimensions in a table for subsequent move instructions.

**Parameters passed:** Two ASCII numbers: One specifies the width of the window (x extent) and the other specifies the height of the window (y extent). The cursor position specifies the upper left corner of the window.

**Explanation:** This function is used prior to the Move function or to the Save Window function. It is not a clipping window.

#### ESC 5 J - SET INVERT CHARACTER AND PRINT DIRECTION

**Action:** Changes print direction and inverts characters

**Parameters passed:** None.

**Explanation:** If the currently enabled character set is horizontal, this function flips the characters upside down and prints from right to left. If the current character set is vertical, this function flips it over and prints from the top down.

#### ESC 5 K - RESET INVERT CHARACTER AND PRINT DIRECTION

**Action:** Returns to normal print direction.

**Parameters passed:** None.

**Explanation:** This function restores the default print direction.

#### ESC 5 L - SELECT FILL PATTERN

**Action:** Stores the selected fill pattern number in a table for use by any subsequent fill command.

**Parameters passed:** A single ASCII number specifying the number for the fill pattern. The number must be between 0 and 8.

**Explanation:** Fill patterns 0-7 are preprogrammed. The density

of the fill pattern increases with the fill pattern number. Fill pattern number 8 is blank. Each of the nine fill patterns can be specified by selecting the Define User Fill Pattern function.

#### ESC 5 M - FILL REGION

**Action:** Fills the bounded region (specified by the cursor position within the region) with the currently enabled pattern.

**Parameters passed:** None.

**Explanation:** The region to be filled is specified by the position of the cursor if the cursor is pointing to a dot that is ON when the Fill Region command is executed, no error is generated and no region is filled.

The region need not be bounded on all sides; the edges of the screen act as boundaries. Complex regions can be filled; however, if the region to be filled contains more than 64 discontinuities, a table overflow error will result.

#### ESC 5 N - FILL BAR

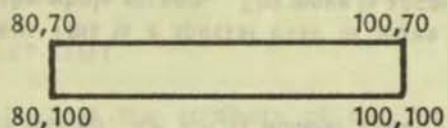
**Action:** The fill is accomplished using the currently enabled fill pattern

**Parameters Passed:** Two ASCII numbers specifying the relative x and y coordinates of the corner of the bar to be filled.

**Explanation:** The bar is a rectangular region. A reference corner is specified by the current position of the cursor. The diagonally opposite corner of the rectangular region is specified by the parameters passed to the Fill Bar function.

**Note:** The coordinates passed to the function are relative to the current cursor position. The x and y values may be positive, negative, or any combination.

**Example:** If the cursor is positioned at 100,100 and the Fill Bar function is called with the parameters -20 and -30, a bar with the following coordinates is drawn:



The bar will be filled using the currently selected fill pattern. None of the combination rules apply—the function erases the portion of the screen and fills it with the selected fill pattern.

#### ESC 5 O - SET LEFT MARGIN

**Action:** Stores left margin.

**Parameters passed:** An ASCII number ( $0 \leq n \leq 799$ ).

**Explanation:** Sets the left margin to be used with Hi-Res print.

#### ESC 5 P - DRAW CIRCLE

**Action:** Draws a circle centered at the current cursor position.

**Parameters passed:** An ASCII number specifying radius in y units.

**Explanation:** To display a circle on the screen, this function draws an ellipse with an aspect ratio of 3:2.

**Example:** If the cursor position is 200,200 and the function is called passing a radius of 100. A circle is drawn with a radius of 100 dots in the y direction and a radius of 150 dots in the x direction.

#### ESC 5 Q - ABSOLUTE POSITION GRAFIX CURSOR

**Action:** Moves the GRAFIX cursor to the absolute x,y position.

**Parameters passed:** Two ASCII numbers specifying the x and y positions (coordinates) on the screen.

**Explanation:** The GRAFIX cursor can be positioned anywhere

within the cursor space as shown in Figure 7-1. Attempting to position the cursor outside that space generates an error.

#### ESC 5 R - RELATIVE POSITION GRAFIX CURSOR

**Action:** Moves the GRAFIX cursor relative to the current cursor position by the increments specified.

**Parameters passed:** Two ASCII numbers specifying the x and y increments.

**Explanation:** Same as Absolute Position GRAFIX Cursor function except the cursor is positioned relative to its prior location.

#### ESC 5 S - SAVE WINDOW ON DISK

**Action:** Saves the contents of the defined window from the Work Screen in the named disk file with extension ".SCR".

**Parameters passed:** Up to 64 ASCII characters specifying a valid pathname.

**Explanation:** Before calling this function, move the cursor to the upper left corner of the desired window. Then specify the extent of the window using the Define Screen Window function. A whole screen can be defined as the window and saved on disk by using this function, but the window must be entirely on the screen.

#### ESC 5 T - LOAD WINDOW FROM DISK

**Action:** Loads the contents of the named file onto the currently enabled screen. The upper left corner of the window coincides with the current cursor position. The file can contain a whole screen or a partial screen. If the file exceeds the screen boundary, it is clipped.

**Parameters passed:** Up to 64 ASCII characters specifying a valid pathname.

**Explanation:** The file name must not contain an extent. The named file must be on a specified drive and must have the extension ".SCR". The cursor must be on the screen when this routine is called.



## ESC 5 U - DRAW LINE (ABSOLUTE)

**Action:** Draws a line from the cursor position to the specified x and y coordinates using the current combination rule, line width, and line type. The cursor is moved to the new x,y position.

**Parameters passed:** Two ASCII numbers corresponding to the absolute x and y position on the screen.

**Explanation:** This function draws a line from the previous cursor position to the absolute x and y coordinates specified, and moves the cursor to the end of the line.

## ESC 5 V - MOVE WINDOW

**Action:** Moves the window using the currently enabled combination rule.

**Parameters passed:** An ASCII number specifying the destination screen.

**Explanation:** Moves a region from the Work Screen defined by the enabled window to the destination screen, using the cursor position as the upper left corner of the window. The function selects the specified screen as the Work Screen when the move is done. The window can be moved from one location to another on the same screen and from one screen to another. The current combination rule remains in effect.

## ESC 5 W - MOVE SCREEN

**Action:** Moves the current Work Screen to the screen specified, using the current combination rule. Selects specified screen as the Work Screen when move is done.

**Parameters passed:** An ASCII number specifying the destination screen.

**Explanation:** This is a special case of Move Window, where the window is the whole screen. The move is accomplished faster than with the Move Window function.

## ESC 5 X - SET COMBINATION RULE

**Action:** Stores the currently selected combination rule.

**Parameters passed:** An ASCII number corresponding to one of 16 combination rules. These are described in Chapter 9.

**Explanation:** This rules applies to all moves, character prints, and line draws, but not to arc draws, circle draws, or fills.

## ESC 5 Y - SET LINE WIDTH

**Action:** Stores specified line width value.

**Parameters passed:** An ASCII number corresponding to the dot width (y units) desired for the line drawing routine. The supported numbers are 1, 2, 4, and 6. Lines other than horizontal lines are automatically scaled to give equal widths.

**Explanation:** The specified value is used in all line, circle, and arc drawings.

## ESC 5 Z - SET LINE TYPE

**Action:** Stores line type value in table.

**Parameters passed:** An ASCII number corresponding to one of five types of lines:

- Type 1 - Solid line.
- Type 2 - Dashed line: 4 dots ON, 4 dots OFF.
- Type 3 - Dashed line: 8 dots ON, 4 dots OFF.
- Type 4 - Dot, dash: 4 dots ON, 2 dots OFF
- Type 5 - Dot, dash: 8 dots ON, 3 dots OFF, 3 dots ON, 3 dots OFF

**Explanation:** The line type applies to all line, circle, and arc drawing.

## ESC 5 a - GET CHARACTER WIDTH

**Action:** Reads the width of the specified character and returns it as an ASCII number.

**Parameters Passed:** A single ASCII character

**Explanation:** This is used with proportional printing.

#### ESC 5 b - GET DOT

**Action:** Returns the attribute of the dot pointed to by the GRAFIX cursor.

**Parameters passed:** None.

**Explanation:** If the dot at the cursor position is ON, the value returned is 1; if it is OFF, the value returned is 0.

#### ESC 5 c - SET DOT

**Action:** Specifies the source dot, and combines it with the dot pointed to by the cursor, according to the set rule.

**Parameters passed:** A single ASCII character (1 or 0) specifying the source dot as ON or OFF.

**Explanation:** If an ASCII 1 is passed, the source dot is specified as being ON; if an ASCII 0 is passed, the source dot is OFF.

#### ESC 5 d - INITIALIZE

**Action:** Resets all functions to their initial values.

**Parameters passed:** None.

**Explanation:** The initial parameter values are:

Cursor Position	0,0
Window Position	0,0
Screen Window	800,400
Character Set	NORMAL
Combination Rule	7 (OR)
Work Screen	0
Display Screen	0
Line Type	SOLID
Line Width	1
Left Margin	0

Shadow Print	OFF
Underline	OFF
Reverse Video	OFF
Double Size	OFF
Print Direction	NORMAL
Cursor	ON
Cursor Type	ARROW
Cursor Saved	0,0
User Cursor x Offset	0
User Cursor y Offset	0
Fill Pattern	0

#### ESC 5 e - GET WINDOW

**Action:** Returns four ASCII numbers specifying the current GRAFIX window. The first pair of numbers specifies the x and y coordinates of the upper left corner of the window; the next pair specifies the width and height of the window.

**Parameters passed:** None.

#### ESC 5 f - DRAW LINE (RELATIVE)

**Action:** Draws a line from the cursor position to the dot specified by the increment arguments, x and y.

**Parameters passed:** Two ASCII numbers specifying the x and y increments.

#### ESC 5 h - DRAW ARC

**Action:** Draws a circular arc between two endpoints in a counter-clockwise direction.

**Parameters passed:** Five ASCII numbers specifying the radius (in y units) and the x and y coordinates of the two endpoints of the arc.

**Explanation:** An arc with the specified radius is drawn between the two endpoints. An error is generated if the x and y coordinates for the endpoints do not fall on a circle of the specified radius.

## ESC 5 i - SELECT CHARACTER SET

**Action:** Selects the named character set for all subsequent Hi-Res print.

**Parameter passed:** File name of the desired character set.

**Explanation:** If the character set is not in memory, it is loaded from disk. The least recently accessed character set is overwritten if all allocated set locations are used. The system character set is protected from overwriting and is accessed by selecting NORMAL. The system graphics character set (ALT) is also protected. A drive may be specified when you enter the file name. If no drive is specified, the default drive is assumed. File names are assumed to have a CHR extension which can be neither specified nor overridden. The same character set name should not be used from more than one disk drive.

## ESC 5 j-GET WORK SCREEN NUMBER

**Action:** Returns the ASCII number of the currently selected Work Screen.

**Parameters passed:** None.

## ESC 5 k - GET DISPLAY SCREEN NUMBER

**Action:** Returns the ASCII number (0 or 1 for current hardware) indicating the current Display Screen.

**Parameters passed:** None.

## ESC 5 l - GET CHARACTER HEIGHT

**Action:** Returns the ASCII value specifying the height of the currently selected character set.

**Parameters passed:** None.

**Explanation:** All characters in the character set have the same height.

## ESC 5 m - DEFINE USER CURSOR

**Action:** Stores the specified pattern as user cursor in the table for future use.

**Parameters passed:** 34 ASCII numbers. The first two numbers represent the desired x and y offset from the upper left corner to the cursor dot within the pattern. The last 32 numbers represent the cursor pattern ( $0 \leq n < 255$ ).

**Explanation:** The GRAFIX cursor, whatever its shape, always points to a single dot on a screen. The cursor character is a pattern 16 dots wide by 16 dots high. The cursor dot is specified by the x and y offsets with respect to the upper left hand corner of the cursor pattern.

Each of the last 32 ASCII numbers defines 8 dots of the cursor pattern. Each pair of numbers defines a row of the cursor pattern. For example, if you want to define an arrow, specify the following pattern:

	Number Pair	Pattern
Row 1	255,255	11111111 11111111
Row 2	192,0	11000000 00000000
Row 3	160,0	10100000 00000000
Row 4	148,0	10010000 00000000
Row 5	136,0	10001000 00000000
Row 6	132,0	10000100 00000000
Row 7	2,0	00000010 00000000
Row 8	1,0	00000001 00000000
Row 9	0,128	00000000 10000000
Row 10	0,64	00000000 01000000
Row 11	0,32	00000000 00100000
Row 12	0,0	00000000 00000000
Row 13	0,0	00000000 00000000
Row 14	0,0	00000000 00000000
Row 15	0,0	00000000 00000000
Row 16	0,0	00000000 00000000

## ESC 5 n - DEFINE USER FILL PATTERN

**Action:** Stores the specified fill pattern in the appropriate location in the table.

**Parameters passed:** 33 ASCII numbers. The first number specifies the fill number this pattern will be accessed by; the remaining 32 specify the 16 by 16-dot pattern used as the fill pattern.

**Explanation:** A fill pattern is defined as a 16 x 16 array of dots. Each ASCII number defines 8 dots, the first two numbers define the first row of dots, the second two define the second row, and so on, until the sixteenth row. See the explanation under "Define User Cursor". Fill patterns 0-7 are programmed with increasing density as the number increases. Fill pattern 8 is completely blank. Any of the nine fill patterns can be defined and specified.

#### ESC 5 o - GET CHARACTER SET TYPE

**Action:** Returns an ASCII number specifying the selected character set type as follows:

- 0 - Normal character set booted with the system (10 dot wide, 16 dots high).
- 4 - Alternate set booted with the system.
- 8 - Horizontal, Non-proportional.
- 9 - Horizontal, Proportional.
- 10 - Vertical, Non-proportional.
- 11 - Vertical, Proportional.
- 12 - Special, Horizontal, Non-proportional.
- 13 - Special, Horizontal, Proportional.
- 14 - Special, Vertical, Non-proportional.
- 15 - Special, Vertical, Proportional.

**Parameters passed:** None.

#### ESC 5 p - HI-RES PRINT

**Action:** Prints the characters at the current graphics cursor position and updates the cursor position.

**Parameters passed:** Printable ASCII characters terminated by carriage return/line feed.

#### ESC 5 q - ENABLE CURSOR

**Action:** Displays the currently selected cursor at the cursor position.

**Parameters passed:** None.

#### ESC 5 r - DISABLE CURSOR

**Action:** Does not display the cursor.

**Parameters passed:** None.

**Explanation:** Even though the cursor is not visible, you can position it anywhere within the cursor positioning space.

#### ESC 5 s - ENABLE SHADOW PRINT

**Action:** Shadow prints all characters using Hi-Res print.

**Parameters passed:** None.

**Explanation:** Shadow printing is accomplished by printing a character twice. On the second printing, the character is moved one dot to the right.

#### ESC 5 t - DISABLE SHADOW PRINT

**Action:** Returns to normal print.

**Parameters passed:** None.

#### ESC 5 u - GET GRAFIX CURSOR

**Action:** Returns two ASCII numbers. The first corresponds to the absolute x position of the cursor, the second to the absolute y position.

**Parameters passed:** None.

#### ESC 5 v - ENTER REVERSE VIDEO MODE

**Action:** Causes all further Hi-Res print to be in reverse video.

#### ESC 5 w - RESET REVERSE VIDEO MODE

**Action:** Returns to normal print.

**Parameters passed:** None.

ESC 5 x - SELECT CURSOR TYPE

**Action:** Uses the specified cursor.

**Parameters Passed:** An ASCII number as follows:

- 0 - Block
- 1 - Cross-hair
- 2 - Arrow (the default )
- 3 - User-defined

ESC 5 y - SET UNDERLINE MODE

**Action:** All characters printed from this point are underlined. (The second row of dots from the bottom of the character cell is turned ON.)

**Parameters passed:** None.

ESC 5 z - RESET UNDERLINE MODE

**Action:** Returns to normal print.

**Parameters passed:** None.

ESC 5 0 - SAVE GRAFIX CURSOR POSITION

**Action:** Saves the current GRAFIX cursor position for subsequent return.

**Parameters passed:** None.

ESC 5 1 - RETURN GRAFIX CURSOR TO PREVIOUSLY SAVED POSITION

**Action:** GRAFIX cursor returns to position saved when ESC 5 0 was executed and stays on the screen.

**Parameters passed:** None.

ESC 5 2 - CLEAR SCREEN

**Action:** Sets all dots on the Work Screen to zero.

**Parameters passed:** None.

ESC 5 9 - TOGGLE TEXT WINDOW SCREEN

**Action:** Successively enables and disables printing to the text window.

**Parameters passed:** None.

ESC 5 ? - SCREEN DUMP TO DOT MATRIX PRINTER

**Parameters passed:** None.

**Action:** Prints the Work Screen on the printer.

**Explanation:** The type of printer, if any, is identified at installation. You must install the printer correctly.

# HI-RES PRINTING

## 6

### 6.1 CHARACTER SETS

The dot pattern for any character is contained in a 16-dot-wide by 16-dot-high matrix. Each character set has two dimension attributes: height and width.

Character height (H) specifies the height (in dots) of the field to be covered by the characters. Character height is the same for all 128 characters in the set.

Character width (W) is the same for all characters in the set (non-proportional printing), or depends on the width of each character (proportional printing). In proportional printing, for example, the letter W is wider than the letter I. Each character in a set is defined by a window starting at the lower left of the 16 x 16 dot character array. This window has a width specified for that character and a height specified for the whole character set. Essentially, any given character is a "small window" which is being transferred to the selected screen.

Since all characters within any given set have the same height, the cursor proceeds along the top of the boundary defined for the characters as they are printed (if you're using only one character set).

The Hi-Res print function accepts a string of characters to be printed and automatically allocates (from left to right) the amount of space needed for each character. If the character set is proportionally spaced, the printing on the screen is also proportionally spaced (the cursor automatically advances to the next character position each time). You must manage the line width - automatic end-of-line wrap around is not implemented.

If you execute a Hi-Res carriage return/line feed (ASCII code 160, 161) within a string, the Hi-Res cursor advances down and to the left, and then prints the rest of the string. (A normal return/line feed terminates the Hi-Res print function. The specific number of dots the cursor moves down is determined by the height parameter for the character set. The cursor movement to the left is limited by the current left margin.

In contrast to printing on a terminal, the cursor is not a block equal to the size of a character. Instead, the cursor points to a single dot. This dot is at the upper left corner of the character field.

The Hi-Res print routine prints any character between 32 and 127 when it receives the corresponding ASCII code. Characters 0 through 31 within a character set are accessed by sending ASCII codes of 128 through 159.

### 6.2 THE HI-RES PRINT FUNCTION

The Hi-Res print function lets you print character strings on a graphics screen. Place the cursor at the spot where you want the upper left corner of the first character in the string. Then call the Hi-Res print function with the string of characters you want to print as an argument. Terminate the function with a carriage return/line feed.

Before executing the print function you can specify:

- \* The type font
- \* Shadow printing
- \* Reverse video
- \* Underlining
- \* Subscript and superscript printing
- \* Double character size
- \* The left margin

These features are described in the following examples. (g\$ is equivalent to the escape character plus "5".)

1. To print with a different type font (Character Set), use the Select Character Set function (ESC 5 i). For example, if you want to load a set called "script" located on drive B:, execute this command:

```
print g$ + "ib:script"
```

2. With shadow printing, a character is printed a second time by ORing in the same character one dot to the right. To shadow print a string of characters, use the Enable Shadow Print Function (ESC 5 s). For example, if you want to shadow print the phrase

"this is important", execute ESC 5 s, print the string using the print function, then turn off shadow print with the Disable Shadow Print function (ESC 5 t):

```
print g$ + "s"
print g$ + "p" + "this is important"
print g$ + "t"
```

3. To print a word in reverse video, use the Set Reverse Video function (ESC 5 v). For example, to print the word "Warning" in reverse video, execute the reverse video function, print the word, and then exit reverse video:

```
print g$ + "v"
print g$ + "pWarning"
print g$ + "w"
```

4. To underline a character string, use the Set Underline Mode function (ESC 5 y). For example:

```
print g$ + "y"
print g$ + "p" + the_string$
print g$ + "z" 'to reset underline mode
```

5. Superscript and Subscript modes move the cursor up or down a distance equal to one-half the character height. To print a super-script, use the Set Superscript Shift Mode function (ESC 5 C). To print a subscript, use the Set Subscript Shift Mode function (ESC 5 E).

For example;

Superscript:

```
print g$ + "C"
print g$ + "p2"
print g$ + "D"
```

```
print g$ + "p + c"
print g$ + "C"
print g$ + "p2"
print g$ + "D"
print g$ + "p = 0"
```

Subscript:

```
print g$ + "pH"
print g$ + "E"
print g$ + "p2"
print g$ + "F"
print g$ + "pO"
```

If you change the character font size, you must position the cursor exactly if you print superscript or subscripts.

6. When you call the function Set Double Character Size Mode (ESC 5 G) all subsequent graphics characters are printed double their normal width and height. You must set and reset (ESC 5 H) the mode as described in the preceding explanations.
7. ESC 5 J, Set Invert Character and Print Direction, causes all subsequent graphics characters to be printed upside down and from right to left. (If you're using a vertical character set, then characters are flipped over and printed from top to bottom.) Set and reset (ESC 5 K) the mode as described in the preceding explanations.

### 6.3 PROPORTIONAL PRINTING

If you use a proportionally spaced character set, GRAFIX automatically provides the correct amount of space between characters. (You don't need a "print proportional" function.) However, you must keep track of the location of the right margin.

Use the Get Character Width function (ESC 5 a) to keep a running total of the width of the characters, and compare it to the coordinate of the right margin.

You can do microspacing by using the Relative Position Cursor function (ESC 5 R).

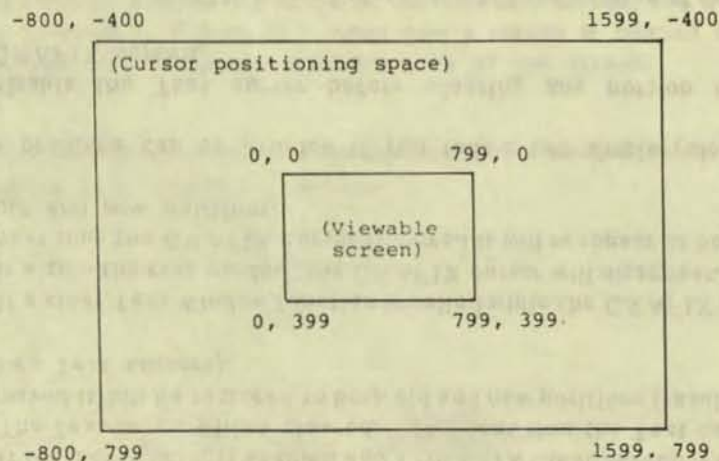
# THE CURSOR

## 7

### 7.1 POSITIONING THE CURSOR

You can position the cursor anywhere within the actual screen - that is, in the space between the limits of -800 and +1599 for x and -400 and +799 for y. The total cursor space is shown in Figure 7-1. All cursor positioning refers to the currently selected screen.

Figure 7-1: Total Cursor Space



The cursor may be selected for printing (cursor ON) or non-printing (cursor OFF) within the actual screen, but it defaults to non-printing if you position it outside the actual screen. To move the cursor from one screen to another, simply change the Work Screen.

### 7.2 POINTING THE CURSOR

The GRAFIX cursor always points at a single dot on the screen. Table 7-1 indicates the target dots of the different cursors.

Table 7-1: Targets of Grafix Cursors

Type of Cursor	Target Dot
Block Cursor	The upper left corner of the block is the absolute x and y position of the cursor.
Arrow Cursor	The dot corresponding to the tip of the arrow is the absolute x and y location of the cursor.
Cross-Hair Cursor	The dot in the middle of the cross-hair specifies the absolute x and y position of the cursor.
User-defined Cursor	The absolute location of the dot pointed to is the upper left corner of a 16 X 16 dot array specified by the x and y cursor offsets.

### 7.3 MULTIPLE CURSOR PROBLEMS

GRAFIX supports two cursors: the text cursor and the GRAFIX cursor. You can position the GRAFIX cursor anywhere within the cursor positioning space. You can position the text cursor anywhere within the text window.

When a GRAFIX command to print or draw is issued, the GRAFIX cursor is removed from the screen, the command is executed, and the GRAFIX cursor is then XORed back into the screen. When the cursor is moved, it is XORed out of the current position and XORed into the new position. The GRAFIX cursor never disturbs any of the graphics on the screen. The Text cursor, a block, is also XORed into the screen and XORed out when it is moved.

Two kinds of screen erase are possible. The GRAFIX Clear Screen function (ESC 5 2) erases the entire Work Screen which might not be the current Display Screen). The Standard Clear function (ESC E) clears the text window only.



If the GRAFIX cursor is enabled when a GRAFIX Clear Screen is executed, the cursor is removed, the screen is cleared, and the cursor is restored. Similarly, if the Text cursor had been enabled when the Standard Clear function was called, the Text cursor is removed. The text window is cleared, and then the Text cursor is restored.

The GRAFIX Clear Screen manages only the GRAFIX cursor; it does not manage the Text cursor. The Standard Clear function manages only the Text cursor, not the GRAFIX cursor. Multiple cursors will result if either of the screen clear functions affects a portion of the screen on which the other type of cursor resides. For example:

- If the Text cursor is enabled and a GRAFIX Clear Screen is called. The Text cursor will be cleared. The next time the Text cursor is moved it will be restored to both old and new positions (resulting in two Text cursors).
- If a clear Text Window function is called while the GRAFIX cursor is within the text window, the GRAFIX cursor will disappear. The next time the GRAFIX cursor is moved it will reappear at both the old and new positions.

These problems can be avoided if you follow two simple rules:

1. Disable the Text cursor before clearing any portion of the GRAFIX screen.
2. Disable the GRAFIX cursor before clearing any portion of the text window.

## USING WINDOWS AND SCREENS 8

### 8.1 CREATING A WINDOW

A window is a rectangular portion of the screen of any size (up to the whole screen). To create a window on the Work Screen:

1. Place the cursor at the desired location of the upper left corner of the window, and
2. Enter the extent of the window by using the Define Window function, specifying the width and height of the window in dots.

Windows can be saved on disk, loaded from disk, moved around on a screen, or moved from one screen to another.

### 8.2 MOVING A WINDOW

The Move Window function lets you use one of sixteen possible combination rules to move a window from screen to screen or into different places on the same screen.

For example, if you want to move the upper left quarter of Screen 3 to the lower right quarter of Screen 0, follow this sequence of calls:

1. Enable Screen 3 as the Work Screen.
2. Position cursor to:  $x = 0, y = 0$ .
3. Call the window function, with  $x = 399$  and  $y = 199$ .
4. Move cursor to:  $x = 399, y = 199$ .
5. Specify the combination rule if it needs to be changed.
6. Call the Move Window function, specifying Screen 0.

This moves the window from the upper left corner of Screen 3 to the lower right corner of Screen 0, and changes the Work Screen from Screen 3 to Screen 0.

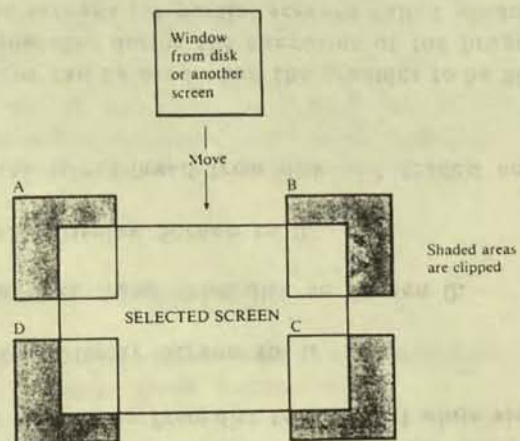
### 8.3 AUTOMATIC CLIPPING

Any point within the cursor positioning space ( $-800 < x < 1599$  and  $-400 < y < 799$ ) can be used for any of the drawing routines (see Figure 7-1).

Graphics are automatically clipped: the viewable screen acts as a clipping window. If you draw a line from the origin ( $x = 0$  and  $y = 0$ ) to  $x = 1599$  and  $y = 799$ , the visible line goes diagonally across the screen ending at  $x = 799$  and  $y = 399$ . The cursor is at position 1599,799. If you attempt to position the cursor outside the cursor space, an error result.

When you move a window that has just been defined or loaded from disk, place the GRAFIX cursor where you want it on the destination screen. (Be sure to invoke the correct combination rule.) If you position the GRAFIX cursor outside the range of the viewable screen, the window will be "clipped." Figure 8-1 shows how a screen is clipped if the window does not fit in the viewable area of the screen.

Figure 8-1: Moving a Window



If the cursor is positioned at the point labeled A (above and to the left of the selected screen), only the portion of the window that coincides with the screen is combined with the screen. The shaded area of the window is discarded.

If the cursor is within the selected screen and the window falls entirely within that screen, all of the window is transferred. When loading a window from disk, be sure the cursor is within the viewable screen.

The Move Window function need not actually perform a move. For example, assume that you have placed the cursor at the upper left corner of a window and that you've specified the  $x$  and  $y$  extents of that window. If you invoke a Move and specify the screen on which the window has been defined, without moving the cursor, the window does not move: however, the combination rule is applied, as if there has been a move.

The portion of the screen defined as the window is combined using the set combination rule. If the rule in effect is  $D^1 = \text{NOT } D$ , the portion of the screen defined by the window turns into reverse video. Similarly, the portion of the screen defined by the window can be erased by invoking the appropriate combination rule ( $D^1 = 0$ ).

### 8.4 SAVING TIME

You can save time in two ways when redrawing graphics figures. First, any number of graphics features can be generated just once and saved on disk. After placing the cursor at the upper left corner of your chosen region, call the Load Window From Disk function.

Second, you can build a set of graphics figures on a background (non-Display) screen, and then recall the set (move it to the Display Screen) when you need it.

Moving a window or a screen does not destroy the original image. If you move a window from Screen 3 to Screen 0, the contents of the window are still available on Screen 3.

## 8.5 USING SCREENS

You can save a screen, or a portion of a screen, on disk. In applications where numerous frames are to be displayed, you should use Screens 0 and 1. Follow these steps:

1. Load the first frame from disk to Screen 1 while viewing Screen 0.
2. Change the Display Screen to 1.
3. Load the next frame from disk to Screen 0.
4. Change the Display Screen to 0.

Then, the data is retrieved from disk and loaded onto the screen rapidly.

Multiple screens can be used when the graphics to be displayed on the screen are generated during the execution of the program. You can build complete screens (or partial screens called windows) on a non-Display Screen and then bring those windows onto the Display Screen.

Multiple screens also save time in applications where you need several graphics screens concurrently. You can build or bring successive frames from disk once - and then select or move them to the viewable screen as desired.

## USING THE COMBINATION RULES 9

You can use the sixteen combination rules to logically combine windows with screens or other windows, and screens with other screens. The first window or screen is called the "destination" (D). The second window or screen, called the "source" (S), is the window or screen you are defining. The rules specify how the destination is transformed by the source. The transformed destination is labeled "D'".

Figure 9-1 shows a possible destination and a possible source combined under each of the sixteen rules. Using the combination rules, you can:

1. Turn ON all the dots in the destination. ( $D' = 1$ )
2. Turn OFF all the dots in the destination. ( $D' = 0$ )
3. Invert all the dots in the destination. ( $D = \text{not } D$ )
4. Move the source into the destination. ( $D = S$ )
5. Move the inverted source into the destination. ( $D = \text{NOT } S$ )
6. Combine the source and destination or invert either one through the logical ANDing, ORing, or XORing of the two.

You need only specify the dimensions of the source. The destination automatically assumes the same dimensions. The source and the destination can be:

- On separate screens
- On different portions of the same screen
- Identical (on the same portion of the same screen)

The combination rules furnish multiple means for achieving the same end. They apply to all window and screen moves, high resolution printing, the set dot function, draw line, draw circle and draw arc functions. All sixteen rules apply to window and screen moves and the set dot function; only three apply to the HI-Res print function.

## 9.1 EXCHANGE OF SCREENS

Assume that Screen 0 is the Display Screen and that you want to view Screen 3 while preserving the contents of Screen 0. If you invoke the Move Screen function, Screen 3 will be transferred to Screen 0, but the contents of Screen 0 will be lost (the contents of both screens are now the same).

If you want to exchange the contents of the two screens without having to write one to a buffer, (a 40,000 byte buffer is needed), follow these instructions (the contents of Screen 0 are labeled S0, the contents of Screen 3 are labeled S3).

1. Set the combination rule to:  $D' = S \text{ XOR } D$
2. Set the Work Screen to 0. This places the cursor on Screen 0.
3. Move Screen to Screen 3. This XORs the contents of Screen 0 with Screen 3 and leaves the results in Screen 3. The Work Screen is now Screen 3.
4. Move Screen to Screen 0. This XORs the contents of Screen 3 with Screen 0 and leaves the results in Screen 0. The original contents of Screen 3 are now in Screen 0. The Work Screen is now Screen 0.
5. Move Screen to Screen 3. This XORs the contents of Screen 0 (S3) with Screen 3 and leaves the result in Screen 3. The original contents of Screen 0 are now in Screen 3.

## 9.2 BLEND OR FADE EFFECT

To fade the contents of another window or screen (called PIC3) into the viewed screen, follow these instructions:

1. Specify the screen containing PIC3 as the Work Screen.
2. Set the combination rule to: OR ( $D' = S \text{ OR } D$ ).
3. Move Screen (or Move Window) to the Display Screen. This ORs-in the desired graphics onto the Display Screen.

4. Set the combination rule to:  $D' = D$ .
5. Specify the screen containing PIC3 as the Work Screen.
6. Move Screen (or Move Window) to the Display Screen. This places PIC3 on the viewed screen.

## 9.3 HI-RES PRINTING

The combination rules that apply to Hi-Res printing are:

- \* Hard print the character. ( $D' = S$ )
- \* OR-in the character. ( $D' = D \text{ OR } S$ )
- \* XOR-in the character. ( $D' = D \text{ XOR } S$ )

If a combination rule other than one of these three is in effect when Hi-Res print is activated, the OR rule is automatically implemented as a default.

## 9.4 LINE DRAW FUNCTIONS

The sixteen combination rules can be summarized in the following four rules for all the Line Draw function. In this case, the line is the source and all the dots in the source are turned ON.

- \* Hard write (or OR-in) the line. ( $D' = S$ )
- \* Erase a line. ( $D' = \text{NOT } S$ )
- \* XOR-in the line. ( $D' = D \text{ XOR } S$ )
- \* No operation. ( $D' = D$ )

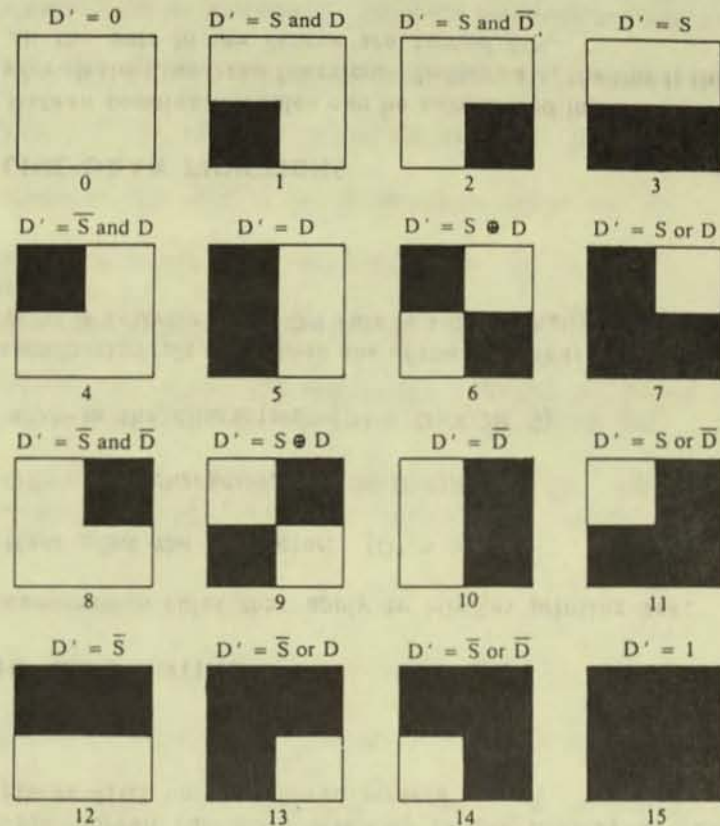
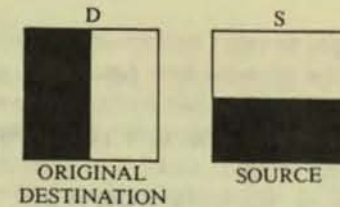
## 9.5 THE COMBINATION RULES

The portion of a screen (or a whole screen) to be moved is called the source and is labeled "S". The portion of the screen (or a whole screen) targeted as the location of the move is called the destination. Prior to your move, it is labeled "D"; after the move, the destination is labeled "D'".

The following rules specify the possible transformations of the destination by the source:

0.  $D' = 0$ : The destination RAM contains all zeros.
1.  $D' = S \text{ AND } D$ : The destination is formed by ANDing the corresponding bits in the original destination and the source.
2.  $D' = S \text{ AND NOT } D$ : The destination is formed by ANDing the complemented original destination with the source.
3.  $D' = S$ : The source is moved to the destination.
4.  $D' = \text{NOT } S \text{ AND } D$ : The destination is formed by ANDing the inverted source with the destination.
5.  $D' = D$ : No operation: no change in the destination results regardless of source.
6.  $D' = S \text{ XOR } D$ : The destination is formed by XORing the corresponding bits in the original destination and the source.
7.  $D' = S \text{ OR } D$ : The destination is formed by ORing the corresponding bits in the original destination and the source. This is the starting combination rule.
8.  $D' = \text{NOT } S \text{ AND NOT } D$ : The destination is formed by ANDing the complemented destination and the complemented source.
9.  $D' = \text{NOT } S \text{ XOR } D$ : The destination is formed by XORing the corresponding bits in the original destination and the complemented source.
10.  $D' = \text{NOT } D$ : The destination is formed by complementing all the bits in the original destination.

Figure 9-1: Combination Rules



11. **D<sup>1</sup> = S OR NOT D:** The destination is formed by complementing the original destination and ORing the corresponding bits with the source.
12. **D<sup>1</sup> = NOT S:** The complemented source is moved to the destination.
13. **D<sup>1</sup> = NOT S OR D:** The destination is formed by ORing the corresponding bits in the original destination and the complemented source.
14. **D<sup>1</sup> = NOT S OR NOT D:** The destination is formed by ORing the complemented original destination and the complemented source.
15. **D<sup>1</sup> = 1 :** The destination contains all ones.

The combination rules provide flexibility in graphics presentation, including simulated motion on the screen. Figure 9-1 shows the results of applying the combination rules to a simple destination and source.

For your convenience, here are some simple examples of using combination rules.

**Inverting a Screen:** If you select combination rule **D<sup>1</sup> = NOT D** and specify a move from Screen N back to Screen N, each dot on Screen N is inverted. Since only Screen 0 or Screen 1 can be selected as the Display Screen, the function allows Screen N (if N>1) to be moved to the screen selected for display. This lets you display Screen N.

**Erasing a Screen:** When used with a combination rule. The Move function allows any screen to be erased by setting that screen to all ones or all zeros. If you set a screen to all ones and then XOR characters into that screen, a black-on-green output results. Invoking **D<sup>1</sup> = NOT D** creates the normal green-on-black display.

**Scratchpad:** You can select a given screen as a scratchpad where drawings, graphs and characters are formed and then brought onto the Display Screen. Simply invoke a move using any of the combination rules.

## ESCAPE AND CONTROL SEQUENCES

# A

### A.1 TERMINAL FUNCTIONS

- ESC H** Moves the cursor to the first character position on the defined top line.
- ESC C** Moves the cursor one character position to the right. If the cursor is at the right end of the line, it remains there.
- ESC D** Moves the cursor one character position to the left. If the cursor is at the start (left end) of a line, it remains there.
- ESC B** Moves the cursor down one line without changing columns. If the cursor reaches the bottom line, it remains there and no scrolling occurs. No action is taken on bottom line + 1.
- ESC A** Moves the cursor up one line without changing columns. If the cursor reaches the top line, it remains there and no scrolling occurs. No action is taken on the bottom line +1.
- ESC I** Moves the cursor up one line, remaining in the same column. If the cursor is on the top line, a scroll down is performed. No action is taken on bottom line + 1.
- ESC n** The operating system's Console In function reports the cursor position in the form of ESC Y line # column #.
- ESC h** Moves the cursor left to next mod 8 position. Stops at left side of screen.
- ESC j** The present cursor position is saved so the cursor can be returned to it after a Set Cursor To Saved Position command (ESC K).
- ESC k** Returns the cursor to the Position where the Save Cursor Position command was last executed. (See ESC Y).

**ESC Y I# c#** Moves the cursor to the position you indicate on the screen by entering the escape code, the character representing the line number, and the character representing the column number.

Line and column numbers should be offset by 32. Thus, to move to the (1,1) position of the screen, follow ESC Y by two occurrences of the character representing the line number, "1".

If the line number entered is smaller than the defined top line, the cursor moves to the top line. If the line number entered is greater than the defined bottom line + 1, the cursor does not move from its present line. If the column number is too high, the cursor moves to the end of the line.

**ESC E** Erases the screen from the defined top line to the defined bottom line. If the cursor is on the bottom line + 1, the function erases bottom line + 1 only. The cursor remains in the home position. Places the cursor in the home position.

**ESC b** Erases from the start of the screen to the cursor, including the cursor position.

**ESC J** Erases the screen from the cursor (including the cursor position) to the end of the defined bottom line. If on bottom line + 1, then erases to the end of the line only.

**ESC o** Erases from the beginning of line to the cursor, including the cursor position.

**ESC K** Erases from the cursor (including the cursor position) to the end of the defined bottom line. If on bottom line + 1, erases to end of line only.

**ESC L** Inserts a new blank line by moving the line that the cursor is on (and all the following lines) down one line, to the defined bottom line. Then the cursor is moved to the beginning of the new blank line. No action is taken on bottom line + 1.

**ESC M** Deletes the contents of the line that the cursor is on. Then places the cursor at the beginning of the line, moves all the following lines up one line, and adds a blank line at the defined bottom line. No action is taken on bottom line + 1.

**ESC N** Deletes the character at the cursor position and shifts any text located to the right of the cursor one position to the left.

**ESC Z** Responds with "ESC/K" to indicate that it can perform as VT52.

**ESC @** Enters Insert Character mode, allowing insert into text on the screen. As new characters are typed in, text to the right of the cursor shifts to the right and the character at the end of the line is lost.

**ESC O** Exits from the Insert Character mode.

## A.2 CONTROL CODES

**CTRL G** Bell (07H).  
Generates a bell sound.

**CTRL H** Backspace (08H).  
Moves the cursor back one column. If wrap-around mode is enabled and the cursor was at column 1, then the cursor is positioned at last column of the previous row (unless at column 1, row 1, in which case the cursor is positioned at last column in row 1). If in discard mode, the cursor does not move from column 1.

**CTRL I** Horizontal Tab (09H).  
Moves the cursor forward to the next tab stop. Tab stops are fixed at Columns 9, 17, 25, 33, 41, 49, 56, 65, 73, and 81. If the cursor is at the last column, it remains there.

**CTRL J** Line Feed (0AH).  
Moves the cursor down one line at same horizontal

position, scrolls the screen up if a line feed occurs on the bottom line. If the cursor is on the bottom line + 1, then no action is taken.

- CTRL M** Return (0DH).  
Moves cursor to the leftmost column of the same line.
- CTRL X** Cancel (18H).  
Aborts any escape sequence in progress. Starts displaying characters as normal ASCII.
- CTRL [** Escape (1BH).  
Starts an escape sequence.
- CTRL O** Shift In (0EH). (Not implemented.)  
Switches the character cell size to 10 by 16, resulting in a display of 80 columns by 25 lines. The top of the screen is set to line 1 and the bottom of the screen is set to line 24. The cursor homes.
- CTRL N** Shift Out (0FH). (Not implemented.)  
Switches the character cell size to 6 by 10, resulting in a display of 133 columns by 40 lines. The top of the screen is set to line 1 and the bottom of the screen is set to line 39. The bottom + 1 is line 40, similar to VT52's 25th line. The cursor homes.

### A.3 CONFIGURATION FUNCTIONS

**ESC x Ps** - Sets the following modes, where Ps equals:

- 1 = Enable bottom line
- 4 = Block cursor
- 5 = Cursor off
- 8 = Automatic line feed on receipt of CR
- 9 = Automatic CR on receipt of line feed

A = Send to VT52

B = Send to VT52

C = Send to VT52

**ESC y Ps** - Resets the following modes, where Ps equals:

- 1 = Disables bottom line
- 4 = Underscore cursor (OK if character height > 9)
- 5 = Cursor on
- 8 = No automatic line feed
- 9 = No automatic CR (carriage return)
- A = Send to VT52
- B = Send to VT52
- C = Send to VT52

### A.4 OPERATION MODE FUNCTIONS

- ESC [** Sets hold mode. (Not implemented.)
- ESC /** Clears hold mode. (Not implemented.)
- ESC p** Enters reverse video mode. Characters are displayed as black characters on a green background.
- ESC q** Exits reverse video mode.
- ESC F** VT52 graphics character in character numbers 94 to 127 of the ASCII character set.
- ESC G** Exits GRAFIX mode. Normal lowercase characters appear in characters numbers 94 to 127.
- ESC t** Enter keypad shift mode. (Not implemented.)



ESC u Exit keypad shift mode. (Not implemented.)

ESC = ALT. keypad on. (Not implemented.)

ESC > ALT. keypad off. (Not implemented.)

ESC { Inhibits the output of the keyboard. Pass through to VT52.

ESC } Enables the keyboard after a keyboard disable command. Pass through to VT52.

ESC v A print to the last column of the line positions to the first column of the next line. The page scrolls up if necessary.

ESC w A print to the last column of the line does not change the cursor position and overprinting occurs. Thus, only the last character received is displayed in the last column position.

ESC z Reset back to 80 column mode. (Not implemented.)

ESC ( Simulates high intensity mode by shadow printing. (Not implemented.)

ESC ) Prints normal characters. (Not implemented.)

ESC # Will transmit only a RETURN (0DH) LINE FEED (0AH).

ESC \$ Will transmit only a RETURN (0DH) LINE FEED (0AH).

ESC ] Will transmit only a RETURN (0DH) LINE FEED (0AH).

ESC \_ Turns the debug mode on of off. In debug mode, the bottom line + 1 displays the hex codes for the print stream.

ESC 0 Sets the underline mode.

ESC 1 Resets the underline mode.

ESC 4 Sets key value. Five characters are passed through to VT52 to set new key values.

ESC 8 Displays next character literally.

ESC m2 p1p2 Redefines the text the text window: p1 defines the first line of the test window; p2 defines the last line of the text window. p1 and p2 are ASCII representations of the row numbers, as described in Figure A-1.

---



---

Figure A-1: Window R definition Parameters

row 0 (space)	row 9 )	row 17 1
row 1 !	row 10 *	row 18 2
row 2 "	row 11 +	row 19 3
row 3 #	row 12 ,	row 20 4
row 4 \$	row 13 -	row 21 5
row 5 %	row 14 .	row 22 6
row 6 &	row 15 /	row 23 7
row 7 ' !	row 16 0	row 24 8
row 8 (		

---



---

---

## ERROR MESSAGES

# B

If you give an invalid command, the Grafix Package generates one of the following error messages:

- **Invalid Command:** This error occurs when you follow ESC 5 by a symbol which is not interpreted by the Grafix Kernel.
- **Parameter(s) Missing:** This error occurs when you invoke an escape sequence which expects parameters to be passed, and parameters have been omitted or an insufficient number has been passed.
- **Invalid Parameter:** This error occurs when you execute any invalid command. For example, the error message appears if you try to move a window to Screen 4 when Grafix has been installed with only three screens. This error also occurs when you pass numeric parameters and alpha parameters are expected, or vice versa.
- **Cursor Out of Range:** This error occurs when you attempt to move the cursor out of the cursor positioning range (between -800 and 1599 in the x direction, and -400 and 799 in the y direction). This message also occurs if you call a function which expects the cursor to be in the screen range and it is not.
- **File Not Found:** This error occurs if you try to enable a character set that does not exist on the specified or default drive. It is also generated when a Load Window From Disk function is accessed and the file name passed does not exist.
- **Disk Full:** This error occurs if you execute the Save Window On Disk command and there is not enough space to save the file on the specified (or default) drive.
- **Invalid File:** This error occurs if the header sector does not match the format required by a Load Window or Select Character Set command. This occurs if the file has length "zero" or if the file was not generated correctly.

- **Table Overflow:** This error occurs if you try to use the Fill function in a polygon whose shape is too complex.
- **Out of Character Sets:** This error occurs if there are no character sets available to load when you do a Select Character Set. This can occur if you invoke Grafix with a maximum of two character sets, or if all the loadable sets have headers designating them as system character sets.

# GRAFIX FUNCTIONS REFERENCE LIST

## C

### C.1 PARAMETER INPUT COMMANDS

Select the Work Screen. . . . .	ESC 5 A
Select the Display Screen . . . . .	ESC 5 B
Select the fill pattern . . . . .	ESC 5 L
Define user fill pattern. . . . .	ESC 5 N
Define a screen window. . . . .	ESC 5 I
Select a character set. . . . .	ESC 5 i
Select a cursor type. . . . .	ESC 5 x
Define user cursor. . . . .	ESC 5 m
Enable cursor . . . . .	ESC 5 q
Disable cursor. . . . .	ESC 5 r
Select a combination rule . . . . .	ESC 5 X
Set relative cursor position. . . . .	ESC 5 R
Set absolute cursor position. . . . .	ESC 5 Q
Set line width. . . . .	ESC 5 Y
Set line type . . . . .	ESC 5 Z
Set left margin . . . . .	ESC 5 O
Set a dot . . . . .	ESC 5 c
Set superscript mode. . . . .	ESC 5 C
Reset superscript mode. . . . .	ESC 5 D
Set subscript mode. . . . .	ESC 5 E
Reset subscript mode. . . . .	ESC 5 F
Set invert character & print direction. . . . .	ESC 5 J
Reset invert character & print direction. . . . .	ESC 5 K
Set double character size mode. . . . .	ESC 5 G
Reset double character size mode. . . . .	ESC 5 H
Enable shadow print . . . . .	ESC 5 s
Disable shadow print. . . . .	ESC 5 t
Set reverse video mode. . . . .	ESC 5 v
Reset reverse video mode. . . . .	ESC 5 w
Set underline mode. . . . .	ESC 5 y
Reset underline mode. . . . .	ESC 5 z
Save GRAFIX cursor position . . . . .	ESC 5 O

### C.2 PARAMETER RETURN COMMANDS

Get enabled screen number . . . . .	ESC 5 j
Get displayed screen number . . . . .	ESC 5 k
Get window parameters . . . . .	ESC 5 e
Get dot . . . . .	ESC 5 b
Get character width . . . . .	ESC 5 a
Get character height. . . . .	ESC 5 l
Get character type. . . . .	ESC 5 o
Get GRAFIX cursor . . . . .	ESC 5 u

### C.3 ACTION COMMANDS

Fill a region . . . . .	ESC 5 M
Fill at bar . . . . .	ESC 5 N
Draw a circle . . . . .	ESC 5 P
Draw an arc . . . . .	ESC 5 h
Absolute line draw. . . . .	ESC 5 U
Relative line draw. . . . .	ESC 5 f
Move a window or a screen . . . . .	ESC 5 V
Move a screen . . . . .	ESC 5 W
Initialize. . . . .	ESC 5 d
Hi-Res print. . . . .	ESC 5 p
Clear screen. . . . .	ESC 5 2
Return GRAFIX cursor to previous saved position	ESC 5 1
Print a screen. . . . .	ESC 5 ?
Toggle text window screen . . . . .	ESC 5 9

### C.4 FILE MAINTENANCE COMMANDS

Save a window on disk . . . . .	ESC 5 S
Load a window from disk . . . . .	ESC 5 T
Select character set (if that character set is not already in memory. . . . .	ESC 5 i

PUBLIC DOMAIN SOFTWARE  
MAIL ORDER FORM

VICTOR\*GROUP  
1493 Beach Park Boulevard, #180  
Foster City, CA 94404

Date _____	Office use only:
Name _____	Order rec'd _____
Company _____	Member check _____
Address _____	Comments _____
City _____ State _____ & Zip _____	_____
Country _____ Phone _____	Order shipped _____

Circle disk numbers ordered (all are single-sided). Un-archived versions are available where "A", "AB", or "ABC" appears below number. If originally released (archived) version is desired, cross out corresponding "A", "AB", or "ABC". (Original releases of all other disks contain no archived files.) Note that pricing below is per "disk number" rather than per "disk" (re-released unarchived "disk numbers" may be comprised of one disk (A), two disks (AB), or three disks (ABC). Write for a catalog that has a brief description of disks 1-60 and (Part II) of disks 61-84. See VictorTalk for brief descriptions, as follows (Index, Vol. 5, No. 5):

Vol. No. Disks	Vol. No. Disks	Vol. No. Disks	Vol. No. Disks
4 2 61-64	5 1 78-81	5 5 87-90	6 3 96-97
4 3 65-71	5 2 82-84	5 6 91-92	6 4 98
4 6 72-77	5 4 85-86	6 2 93-95	6 5 99

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34*	35*	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54 A	55	56*	57*	--	59	60 AB
61	62	63	64	65	66	67	68	69	70	71	72	73 A	74	75	76	77	78 AB	79 AB	80 AB
81 AB	82 AB	83 AB	84 AB	85 AB	86 AB	87 AB	88 ABC	89 ABC	90 AB	91 AB	92 ABC	93 ABC	94 AB	95 ABC	96 AB	97 AB	98	99	101

102-104#

\* Two-disk sets    # Three-disk sets

1988 Victor\*Group member: No \_\_\_\_\_ Yes \_\_\_\_\_ Date joined/renewed \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$5.00 (members only)..... \$ \_\_\_\_\_

Number of disk numbers ordered \_\_\_\_\_ x \$10.00 (non-members)..... \$ \_\_\_\_\_

Postage/handling (applies to disk orders only)..... \$ 4.00

Overseas orders, add an additional \$1.00 per disk ordered..... \$ \_\_\_\_\_

VictorTalk back issues: 1986, 1987, 1988, sets of 6, \$12.00 /set.. \$ \_\_\_\_\_

1985, set of 9, \$1.00 each ..... \$ \_\_\_\_\_

1989 membership fee \$35 (receive all 1989 issues of VictorTalk) .. \$ \_\_\_\_\_  
(Foreign membership, \$40 in U.S. funds.)

TOTAL ENCLOSED (Make check, U.S. funds only, payable to Victor\*Group; we cannot accept credit cards or purchase orders.).. \$ \_\_\_\_\_