
VICTOR 9000

Supplementary Information

Home of Victor Computers

Rev. A

COPYRIGHT

©1986 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.
IBM is a registered trademark of International Business Machines Corporation.
MS- and GW-BASIC are trademarks of Microsoft Corporation.
Microsoft is a registered trademark of Microsoft Corporation.

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.

First VICTOR printing April, 1986.

ISBN 0-88182-187-X

Printed in U.S.A.

Contents

Preface	V
1. Operator's Reference Guide Changes	
1.1 AUTOSET	1-1
1.2 FORMAT	1-2
2. MS-DOS 3.1 Reference Changes	
2.1 ANSI.EXE	2-1
2.2 FORMAT	2-1
2.3 MODE, PORTSET, and CLST	2-2
2.3.1 PORTSET	2-2
2.3.2 CLST	2-5
2.4 SELECT, KEYBxx, and MODCON	2-5
2.5 TREE, SEARCH, and LS	2-8
3. New VICTOR MS-DOS Operating System Functions	
3.1 Enhancements of MS-DOS	3-1
3.2 MS-DOS 3.1 Features Not Documented Elsewhere	3-2
3.2.1 CAUX and CLST	3-2
3.2.2 HDPARK	3-3
3.2.3 SCRNMx80 and SCRNM84	3-3
3.2.4 CP/M-86 Emulation	3-4
4. VBASICA Changes	
4.1 Enhanced Features	4-1
4.1.1 LCOPY Statement	4-1
4.1.2 The VBCONF Program	4-2
4.1.3 Bit Map Allocation	4-3
4.1.4 Graphics Screens and Modes	4-4

4.2	Features Not Supported in This Version of VBASICA ...	4-8
4.3	Compatibility Problems and Resolutions.....	4-9
4.3.1	Screen Compatibility	4-9
4.3.2	Hardware Compatibility	4-10
4.3.3	Function Keys.....	4-10

Home of Victor Computers

Preface

This guide provides supplementary information for the *Operator's Reference Guide II*, the *MS-DOS 3.1 Reference*, and the *VBASICA Interpreter Reference Manual*.

Section 1, "Operator's Reference Guide Changes," describes changes and enhancements to MS-DOS functions documented in the *Operator's Reference Guide II*.

Section 2, "MS-DOS 3.1 Reference Changes," describes changes and enhancements to MS-DOS functions discussed in the *MS-DOS 3.1 Reference*.

Section 3, "New VICTOR MS-DOS Operating System Functions," describes MS-DOS operating system functions that are included in the current implementation but are not described in either the *Operator's Reference Guide II* or the *MS-DOS 3.1 Reference*.

Section 4, "VBASICA Changes," describes changes to VBASICA features documented in the *VBASICA Interpreter Reference Manual*, as well as differences between the current implementation of VBASICA and IBM's BASICA.

Home of Victor Computers

Operator's Reference Guide Changes

This section describes changes and enhancements to the *Operator's Reference Guide II*.

1.1 AUTOSET

The AUTOSET hard disk configuration utility described in Chapters 2 and 3 of Section III has been enhanced to support configuration of 30-megabyte (MB) internal hard disk drives. If your system has a 30-MB internal hard disk drive, do not use any earlier version of AUTOSET.

All hard disk configuration files (those with the extension .CFG) whose filenames begin with the number 3 are for use with 30-MB internal hard disks only.

1.2 FORMAT

The description of the FORMAT utility in Chapter 3 of Section II does not apply to this implementation of the MS-DOS operating system. Refer to Chapter 6 of the *MS-DOS 3.1 Reference* for a description of the current FORMAT utility. That description applies to the VICTOR 9000 with the following exceptions.

The /8 and /4 switches are not available on the VICTOR 9000.

The following additional switches are available:

- ▶ The /M (messages) switch tells FORMAT to display the current track and other information while formatting.
- ▶ The /C (clear) switch tells FORMAT to clear the directory and file allocation table (FAT) of the target diskette without formatting the diskette. Use this switch to erase an already formatted diskette. If you use it with an unformatted diskette, a write error occurs.

Home of Victor Computers

MS-DOS 3.1 Reference Changes

This section documents changes and enhancements to the *MS-DOS 3.1 Reference*.

2

2.1 ANSI.EXE

The ANSI-defined standard escape sequences for key reassignment, graphics functions, and cursor movement described in Appendix C are available only when the executable file ANSI.EXE is resident in memory.

To have access to the escape sequences described in Appendix C, you must first either explicitly invoke ANSI.EXE or include ANSI.EXE in a CONFIG.BAT file. The CONFIG.BAT file is described on pages 2-13 and 2-14 of the *MS-DOS 3.1 Reference*. Once loaded, ANSI.EXE remains resident in memory until you reboot or turn off the system.

2.2 FORMAT

The description of the FORMAT utility in Chapter 6 applies to the VICTOR 9000 with the following exceptions.

The /8 and /4 switches are not available on the VICTOR 9000.

The following additional switches are available:

- ▶ The /M (messages) switch tells FORMAT to display the current track and other information while formatting.
- ▶ The /C (clear) switch tells FORMAT to clear the directory and file allocation table (FAT) of the target diskette without formatting the diskette. Use this switch to erase an already formatted diskette. If you use it with an unformatted diskette, a write error occurs.

2.3 MODE, PORTSET, and CLST

The MODE command described in Chapter 6 is not included in this implementation of the operating system. The PORTSET and CLST commands perform many of the same functions as MODE.

2.3.1 PORTSET

PORTSET is an external command that you can use to change the transmission settings for serial output ports A and B. Use PORTSET to adjust serial output to match a serial device such as a printer or modem. Entering PORTSET without parameters lists the current settings of the serial port(s) and displays the on-line help command.

The syntax for entering PORTSET with all its parameters is as follows:

PORTSET [port] [baud rate] [parity] [stop bits] [data bits] [/S]

port is the name of the serial port to be set; you must use A or B.

baud rate is the speed, in bits per second, at which data is to be transmitted from the port. The baud rate you set for the serial port must match the baud rate of the serial device attached to the port. These are the possible baud rates you can use:

50	300	2400
75	600	3600
110	1200	4800
134.5	1800	9600
150	2000	19200
200		

These baud rates are also listed in the PORTSET help screens.

parity refers to the number of ON bits in each byte of data transmitted. Parity checking is a method of detecting single-bit errors in transmitted data. The possible values for parity are ODD, EVEN, and NONE (you can abbreviate these values as O, E, or N, respectively).

- ▶ If parity is ODD, a parity bit is switched on or off as necessary to ensure that the number of bits turned on in each byte is always odd.
- ▶ If parity is EVEN, the parity bit is turned on or off to ensure that an even number of bits are on.
- ▶ NONE specifies that no parity checking takes place. If you specify 8 data bits, you **must** specify NONE for parity; a parity bit and an eighth data bit are mutually exclusive.

stop bits specifies the number of bits used to mark the end of each byte of data transmitted. Values for this parameter are 0, 1, 1.5, and 2.

data bits specifies the number of bits in each byte that will carry the data. Values are 5, 6, 7, and 8.

/S suppresses all screen messages that would ordinarily be displayed during execution of a PORTSET command.

When you load MS-DOS, the values for the serial port parameters are taken from the DEVICE commands in the CONFIG.SYS file. These are the default settings in the CONFIG.SYS file on your system diskette:

Baud rate	1200
Parity	NONE
Stop bits	2
Data bits	8

You can separate the parameters of a PORTSET command with spaces, with commas, or with any of the delimiters listed in the first PORTSET help screen. You do not have to enter the parameters in the order shown here; PORTSET can distinguish among parameters regardless of order. Nor do you have to specify all five parameters. If you want to accept the default settings for some parameters and specify others, enter only the parameters you want to specify.

The following PORTSET command sets serial port B to 300 baud, EVEN parity, two stop bits, and six data bits:

portset b, 300, even, 2, 6,

Since the default value for stop bits is 2, the following command has the same effect:

```
portset b, 300, even, 6↵
```

If you make a typing error when you enter the command, PORTSET displays a message pointing out the first command element that is incorrect and giving the correct command syntax. In the following command, for example, the word EVEN is mistyped as EBEN:

```
portset a, 1200, eben, 1↵
```

In response, PORTSET displays the following message:

```
a, 1200, eben, 1
```

```
↑
```

```
PORTSET Version X.X
```

```
*** Error: Invalid command line.
```

```
For help type: PORTSET HELP <Return>
```

```
This is the legal command format:
```

```
PORTSET <Port> [Baud rate] [Parity] [Stop bits]  
[Bits per character] [/S]
```

```
Legal values:
```

```
Port = "A" or "B"
```

```
Baud rate = 50, 75, 110, 134.5, 150, 200, 300,  
600, 1200, 1800, 2000, 2400, 3600,  
4800, 9600, 19200.
```

```
Parity = "Even", "Odd", "None".
```

```
Stop bits = 0, 1, 1.5, 2
```

```
Bits per character = 5, 6, 7, 8.
```

PORTSET changes the default settings only for the current operating session. When you turn off or reset your system, the temporary settings are lost. To change the serial port settings permanently, you must change the DEVICE commands in the CONFIG.SYS file.

2.3.2 CLST

To assign printer output to a serial port, use the internal command CLST, in one of the following forms:

clst seriala ↵

clst serialb ↵

The first command assigns list device output to serial port A; the second command assigns list device output to serial port B.

Note: Before using CLST to change to a serial port, you must first load the device driver for that port. Do this by using the DEVICE command in your CONFIG.SYS file; then the device drivers are loaded at boot time.

2.4 SELECT, KEYBxx, and MODCON

The SELECT and KEYBxx commands described in Chapter 6 are not part of the current implementation of the operating system. In place of SELECT and KEYBxx, the current implementation of the operating system includes the MODCON command.

MODCON is an external command for modifying the keyboard layout and character set used by the console. You can use MODCON to select a new keyboard file and/or a new character set file. You can also save the current set(s) and restore them later.

The command syntax for MODCON is as follows:

**MODCON [path]sourcefile name[sourcefile extension]
[path][savefile name][.savefile extension]** ↵

path specifies the path from the current directory to the source file. If the source file is in the current directory, you can omit **path**.

sourcefile name specifies the file containing the keyboard layout or character set you want to be made active. If you specify the filename without an extension, the corresponding keyboard (.KB) file and character set (.CHR) file are both activated.

sourcefile extension specifies whether a keyboard file or a character set file is to be made active. To specify a keyboard file only, use the extension .KB. To specify a character set file only, use the extension .CHR. To activate both the keyboard and the character set files, omit **sourcefile extension**.

path specifies the path from the current directory to the save file. If the save file is in the current directory, you can omit **path**.

savefile name specifies the file to which the current keyboard layout or character set is to be saved. If you specify the filename without an extension, the current keyboard layout is saved to that filename with the extension .KB, and the current character set is saved to that filename with the extension .CHR.

savefile extension specifies whether the current keyboard layout or the current character set is to be saved. To specify the current keyboard layout only, enter the extension .KB. To specify the current character set only, enter the extension .CHR. To save both the current keyboard layout and the current character set, omit **savefile extension**.

If you want to save the current keyboard layout or character set to a file without activating a new keyboard layout or character set, you can enter an asterisk (*) in place of the sourcefile specification.

You can also activate a keyboard file and save a character set, or vice versa; the name and extension of the source file are independent of the name and extension of the save file.

Some examples of the MODCON command are shown below.

```
modcon b:\dir3\test.chr\chr\save.chr
```

This command modifies the character set, but does not affect the keyboard layout. The new character set is contained in the file TEST.CHR in the \DIR3 subdirectory on drive B. The current character set is saved to the file SAVE.CHR in the \CHR subdirectory on the current drive.

```
modcon \chr\test.kb
```

This command selects a new keyboard file (TEST.KB in the \CHR subdirectory on the current drive) but does not save the current keyboard layout.

```
modcon germ01 g02save
```

This command activates the keyboard layout specified in the file GERM01.KB and the character set specified in GERM01.CHR. It saves the current keyboard layout to G02SAVE.KB and saves the current character set to G02SAVE.CHR.

```
modcon * brit01.kb
```

This command saves the active keyboard layout to the file BRIT01.KB on the current drive, but leaves it as the active keyboard layout. This command does not have any effect on the character set.

The following batch file uses MODCON to set up a dedicated keyboard layout for a WordStar word processing session. It restores the original keyboard layout when you exit from WordStar.

```
modcon wordamer.kb saved.kb  
ws  
modcon saved.kb  
del saved.kb
```

The first command in this batch file activates the keyboard layout specified by WORDAMER.KB and saves the current keyboard layout to the file SAVED.KB. The second command invokes WordStar. The third command re-activates the original keyboard stored in the file SAVED.KB when you exit from WordStar. The last command deletes SAVED.KB.

To use MODCON, you need keyboard layout and character set files for the keyboard layouts and character sets you want to use.

You can obtain keyboard layout files in either of two ways:

- ▶ Choose a keyboard layout file from those provided on the system selection (SYSGEN) diskette in the *Applications Programmer's Tool Kit*.
- ▶ Modify the current keyboard layout table (or any available keyboard layout table) using the keyboard table editor KEYGEN, included in the *Graphics Tool Kit* and the *Applications Programmer's Tool Kit*.

You can obtain character set files in one of three ways:

- ▶ Choose a character set file from those provided on the system selection (SYSGEN) diskette in the *Applications Programmer's Tool Kit*.
- ▶ Choose a graphics character set provided with CHARGRAF in the *Graphics Tool Kit*.
- ▶ Modify the current character set (or any available character set) using the character font editor EFONT included in the *Graphics Tool Kit* and the *Applications Programmer's Tool Kit*.

2.5 TREE, SEARCH, and LS

The TREE command described in Chapter 6 is not included in this implementation of the operating system. The SEARCH and LS commands provide the same functions as the TREE command, as well as many other capabilities. SEARCH and LS are fully documented in Chapter 6.

New VICTOR MS-DOS Operating System Functions

This section describes additional functions that are included in this implementation of the MS-DOS 3.1 operating system.

3.1 Enhancements of MS-DOS 2.1

With the exception of the SWITCHAR and AVAILDEV configuration commands, which are not supported under MS-DOS 3.1, the VICTOR MS-DOS 3.1 operating system provides all of the functions contained in MS-DOS 2.1, plus a number of enhancements:

- ▶ New interrupt 21 functions and a file sharing program to support file and record locking
- ▶ MS-Networks support
- ▶ UNIX-like utilities: CONCAT, FGREP, HISTORY, LS, MV, SEARCH, TAIL, UNIQ, and WC
- ▶ Support for the 30-megabyte hard disk
- ▶ Enhanced PRINT utility that supports path specifications and can be invoked from within user programs
- ▶ Enhanced COMMAND, DISKCOPY, and FORMAT utilities
- ▶ A large Linker that can link programs of up to 1 megabyte
- ▶ New CONFIG.SYS commands: FCBS and LASTDRIVE
- ▶ Enhanced error-recovery facilities

You can now reset the operating system from the keyboard by simultaneously pressing the ALT and LOCK/CAPS LOCK keys and the decimal (.) key on the numeric keypad.

A reset interrupt has been added at interrupt 69h. If you want to reset the operating system from within an application, you can do so by issuing an interrupt 69h. No parameters are required for this interrupt.

A “hot key” interrupt has been added at interrupt 68h. When the keyboard scan codes are available from the hardware, they are passed in the AX register to interrupt 68h; interrupt 68h is called before any processing of the scan code occurs. An application monitoring this interrupt can do any 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

3.2 MS-DOS 3.1 Features Not Documented Elsewhere

This section describes features of MS-DOS 3.1 that are not documented in the *MS-DOS 3.1 Reference* or the *Operator's Reference Guide II*.

3.2.1 CAUX and CLST

In addition to the CTTY command described on page 6-39, VICTOR MS-DOS 3.1 provides two other commands for changing input/output devices: CAUX and CLST. Use CAUX to change the auxiliary input/output device; use CLST to change the list device (printer).

The default device driver for CAUX is AUX. The default device driver for CLST is PRN.

The command syntax and the devicename definition given for CTTY also apply to CAUX and CLST.

3.2.2 HDPARK

HDPARK is a hard disk utility that positions the read/write heads of the hard disk over the innermost cylinder of the drive. Once the read/write heads are moved to this area of the hard disk, data is not damaged in the event that the heads contact the hard disk. Such contact can scratch the hard disk platters and make that area of the disk unusable.

If you have a hard disk drive, you should run HDPARK before you turn your system off. It is also a good practice to run HDPARK and to turn the system off before you move the system or when power to your system is suspect (for example, during storms or brownouts).

CAUTION: Once the heads are parked, you can return to the operating system if you wish. Doing so, however, may unpark the heads.

To invoke HDPARK, enter the command without parameters:

hdpark

3.2.3 SCRNMX80 and SCRNM84

Two screen print utilities have been added: SCRNMX80 and SCRNM84. These utilities are specific to the Epson MX and FX printers and to the Okidata Micro84 printers, respectively. You can invoke either utility from the command line or from within an AUTOEXEC.BAT or CONFIG.BAT file.

Once loaded, the utility stays resident; whenever you press the Print Screen key, the current screen is sent to the printer in a graphics format specific to the chosen printer.

The Print Screen key is assigned to function key 10. You can change this key assignment by using the /K switch on the command line (or in the batch file) when you invoke the screen dump utility. Enter /K followed by a space and a one- or two-digit hexadecimal number that specifies the assignment of the Print Screen key.

For example, to send the contents of the screen to your MX80 whenever you press ALT-D, enter the following command:

```
scrnmx80 /k 04 ↵
```

04 is the ASCII value of ALT-D.

Once you have loaded SCRNMX80 or SCRNM84, you can also initiate a screen dump from within a program by issuing an INT 5.

The screen dump utilities load high in memory; therefore, they do not use memory needed for the graphic screens.

3

3.2.4 CP/M-86 Emulation

The current implementation of the VICTOR MS-DOS 3.1 operating system provides a CP/M-86 emulator that allows you to run programs written for CP/M-86.

Home of Victor Computers

RDCPM

Before you can use the CP/M-86 emulator program, you must first transfer your CP/M-86 programs to an MS-DOS formatted diskette. To do so, use the RDCPM command. RDCPM is an external command that can:

- ▶ Display the directory of a CP/M-86 diskette while you are using MS-DOS
- ▶ Transfer files from a CP/M-86 diskette to an MS-DOS diskette in the current directory

To display the directory of a CP/M-86 diskette, use this syntax:

```
[drive]RDCPM DIR[drive][filename.ext] ↵
```

The first **drive** parameter specifies the drive that contains the MS-DOS diskette with the RDCPM.EXE file. If the MS-DOS diskette is in the current drive, you can omit **drive**.

The second **drive** parameter specifies the drive that contains the CP/M-86 diskette whose directory you want to display.

You can use **filename.ext** with or without wildcard characters to specify the file(s) you want to be displayed.

The following command displays the directory of the CP/M-86 diskette in drive B:

```
rdcpm dir b:␣
```

The next command displays only those files with the extension .EXE:

```
rdcpm dir b:*.exe␣
```

To transfer files from a CP/M-86 diskette to an MS-DOS diskette, use the following syntax:

```
[drive]RDCPM[drive]filename.ext␣
```

The first **drive** parameter specifies the drive that contains the MS-DOS diskette with the RDCPM.EXE file. If the MS-DOS diskette is in the current drive, you can omit **drive**. If you want to copy the file to a specific subdirectory on the MS-DOS diskette, you must use the CHDIR command to change directories before you enter the RDCPM command.

The second **drive** parameter specifies the drive that contains the CP/M-86 diskette from which you want to copy files.

Use **filename.ext** with or without wildcard characters to specify the file(s) you want to copy.

The following command copies the file TEXT.TXT from the CP/M-86 diskette in drive B to the MS-DOS diskette in the current drive:

```
rdcpm b:text.txt,↓
```

The following command copies **all** the files from the CP/M-86 diskette in drive B to the MS-DOS diskette in the current drive:

```
rdcpm b:*.*,↓
```

3

Emulator Operation

Once you have transferred the CP/M-86 programs to an MS-DOS diskette that contains the emulator program CPM.EXE, you can run the programs under the CP/M-86 emulator. To do so, type CPM, followed by a space and the CP/M-86 command line, exactly as you would enter it under CP/M-86.

Memory Management

Memory management functions are fully emulated. Functions such as “system reset” and “stay resident,” which are made necessary by or require the CP/M-86 console command processor (CCP), are not supported. The following memory functions are emulated:

- ▶ Get maximum available memory
- ▶ Get maximum memory, absolute
- ▶ Get memory region
- ▶ Get absolute memory region
- ▶ Free memory
- ▶ Free all memory

The emulator supports the following file system interface functions:

- ▶ Open/Make (Function 15)
- ▶ Close (Function 16)
- ▶ Read sequential (Function 20)
- ▶ Write sequential (Function 21)
- ▶ Read random (Function 33)
- ▶ Write random (Function 34)
- ▶ Set random record (Function 36)
- ▶ Write random with zero fill (Function 40)
- ▶ Search first/next—with the exception noted (Functions 17 and 18)
- ▶ Chain to program (Function 47)
- ▶ Delete (Function 19)
- ▶ Rename (Function 23)
- ▶ File size (Function 35)

Byte-Oriented Device Interface

Byte-oriented logical devices are fully supported, including redirection. The following functions are supported:

- ▶ Console input (Function 1)
- ▶ Console output (Function 2)
- ▶ Reader input (Function 3)
- ▶ Punch output (Function 4)
- ▶ List output (Function 5)
- ▶ Direct console I/O (Function 6)
- ▶ Print string (Function 9)
- ▶ Read console buffer (Function 10)
- ▶ Get console status (Function 11)

System Functions

These system functions are supported:

- ▶ System reset—terminates emulation; does not allow program to remain resident (Function 0)
- ▶ Return version number (Function 12)
- ▶ Reset disk system—MS-DOS disk reset (Function 13)
- ▶ Select disk—uses MS-DOS function (Function 14)
- ▶ Return login vector—reflects all drives known to MS-DOS (Function 24)
- ▶ Return current disk—CP/M-86 equivalent (Function 25)
- ▶ Get DMA address—CP/M-86 equivalent (Function 47)
- ▶ Write-protect disk—marks diskettes as write-protected within the emulator (Function 28)
- ▶ Get address—R/O vector; returns address of the emulator R/O vector (Function 29)
- ▶ Reset drive—resets emulator R/O vector (Function 37)
- ▶ Get/Set DMA base—CP/M-86 equivalent (Functions 51 and 52)
- ▶ Get SYSDAT address where the equivalent of SYSDAT information is found within the emulator, with the same offset values as in CP/M-86 (Function 49)

These direct BIOS calls are supported:

- ▶ INIT terminates emulation (BIOS subroutines)
- ▶ WBOOT terminates emulation (BIOS subroutines)
- ▶ CONST, CONIN, CONOUT (BIOS subroutines)
- ▶ LIST, LISTST, PUNCH, READER (BIOS subroutines)
- ▶ HOME, SELDSK (BIOS subroutines)
- ▶ GET/SET IO BYTE (Functions 7 and 8)

These system functions are **not** supported:

- ▶ Get address—allocate (Function 27)
- ▶ Get/Set user code (Function 32)

These direct BIOS calls are **not** supported:

- ▶ SETTRK, SETSEC, SETDMA (BIOS subroutines)
- ▶ READ, WRITE, SECTAN (BIOS subroutines)
- ▶ SETDMA, SETDMAB, GETSEGB (BIOS subroutines)

If you attempt to use a system function that is not supported by the CP/M-86 emulator, an error message is displayed and the emulator terminates.

3

Home of Victor Computers

Home of Victor Computers

VBASICA Changes

This section documents features of VICTOR's VBASICA that are enhanced relative to IBM's BASICA, lists features that are not supported in this implementation of VBASICA, and discusses some minor compatibility problems and their resolutions.

4.1 Enhanced Features

Features enhanced over BASICA include the LCOPY statement, the VBCONF program, bit map allocation, and additional graphics screens and modes.

Home of Victor Computers

4.1.1 LCOPY Statement

FORMAT:

LCOPY

PURPOSE:

Dumps the screen display to a graphics printer.

REMARKS:

With the LCOPY statement VBASICA can output both text and graphics to a dot-matrix printer. VBASICA supports seven printers; you can also define additional printers. Before you can use LCOPY, you must install VBASICA for the printer you are using. Do so by running the VBCONF.BAS program provided on your distribution diskette (VBCONF is described in the next section). After you install VBASICA for a particular printer, you can use the LCOPY statement only with that printer.

EXAMPLE:

```
10 SCREEN 2           'set hi-res graphics screen
20 KEY OFF             'turn off function key display
30 CLS 2               'clear the screen
40 FOR RADIUS = 20 TO 200 STEP 10
50 CIRCLE (400,200),RADIUS 'draw some graphics
60 NEXT RADIUS
70 LCOPY               'copy the screen to the printer
80 END
```

4.1.2 The VBCONF Program

4 You can run the VBCONF configuration program either from the operating system level or from VBASICA. To run VBCONF from the operating system level, enter the following command in response to the system prompt:

VBASICA VBCONF

To run VBCONF from the VBASICA command level, enter this command in response to the "Ok" prompt:

RUN "VBCONF"

The program displays the following menu choices:

ABORT - Abort this program without changing VBASICA.
VICTOR Printer 6010/6020
VICTOR Printer 6015/6025
VICTOR Printer 6070/6075
Tally Printer (160s/180s)
Tally Printer (140)
C. Itoh Printer (8510A, 8510S/1550A, 1550S)
Epson MX-80/MX-100 (with GRAFTRAX)
Epson FX-80/FX-100
Okidata Printer (U84 Only)
No Printer - disable LCOPY

Use the cursor arrow keys to move the reverse-video bar up and down the menu. When the choice you want is highlighted, press Return. The V9BASICA.XEC file on the default drive is modified and you return to the operating system. If you choose to ABORT, you return to VBASICA. If you choose "No Printer," LCOPY returns an "Illegal function call" error message if you try to use it.

The VBASICA sign-on message displays the name of the installed printer. Rerun VBCONF any time you change printers.

You can install a printer not listed in the menu by modifying the VBCONF.BAS source code. After reading your printer's manual and the comments in the VBCONF.BAS source code, you should be able to configure VBASICA for most printers that support bit-mapped dot-matrix printing.

VICTOR dot-matrix printers automatically correct the aspect ratio between the printer and the screen so that circles are printed correctly, rather than slightly elongated, as is the case with most printers.

Home of Victor Computers

4.1.3 Bit Map Allocation

The graphics functions of VBASICA require a bit map. A **bit map** is memory that defines the graphics screen; this memory is loaded in the lower 64K of RAM.

The bit map for VBASICA is usually allocated at run time by the VBASICA loader. Many load-and-stay-resident utilities, however, load in low memory, thus preventing the VBASICA loader from allocating the bit map. To overcome this conflict, two utilities—GETSCRN and KILLSCRN—have been provided to allocate and deallocate the memory for the bit map. You can run these utilities at any time.

To avoid conflicts with resident utilities, first run GETSCRN to reserve memory for the bit map, and then load your other utilities. You can then run VBASICA at any time, and it will be able to use the bit map allocated by GETSCRN. If you run GETSCRN before VBASICA, VBASICA knows that memory for the bit map has already been allo-

cated and does not attempt to allocate more memory. You can return the reserved memory to the system by using KILLSCRN.

Communication between GETSCRN and VBASICA is accomplished as follows. GETSCRN allocates the bit map (if it can), 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 VBASICA check interrupt D9 with each invocation; if the bit map has already been allocated, they do not allocate another one.

4.1.4 Graphics Screens and Modes

4 VBASICA can create medium- or high-resolution graphics on a color screen or on a standard screen. If your computer has a color card, you can put color and black-and-white images on an attached color screen. VBASICA supports three screen types:

- ▶ The standard VICTOR screen
- ▶ A software-simulated IBM color screen displayed on the standard VICTOR screen
- ▶ The color screen, which requires a color screen and the VICTOR IBM-compatible color card

In addition to three screen types, VBASICA supports three screen resolution modes:

- ▶ Mode 0: text-only mode
- ▶ Mode 1: medium-resolution graphics and text mode
- ▶ Mode 2: high-resolution graphics and text mode

You can set both the screen type and the screen mode from within a VBASICA program by using the SCREEN command. Nine screen/mode combinations are possible, although only eight are implemented.

The possible combinations are shown in Table 1. Each screen number in the table is a two-digit number. The first digit specifies the screen type; the second digit specifies the resolution mode. Use these screen numbers as arguments to the SCREEN command to switch from one type/mode combination to another.

Table 1: Screen Type and Mode Combinations

Screen	Rows	Graphics Columns	Pixels	Comments
Color Screens				
20	25	40/80	No graphics	Full 16 colors, 8 or 4 pages, depending on width
21	25	40	320 × 200	4 colors, medium resolution
22	25	80	640 × 200	2 colors: black and white
Standard Screens				
40	25	40/80	No graphics	Standard text mode
41	(Not available; use screen 61)			
42	25	80	400 × 800	Standard high-resolution screen
IBM Monochrome Screens				
60	25	40/80	No graphics	IBM text mode
61	25	40	320 × 200	4 shades, medium resolution
62	25	40/80	640 × 200	Simulated IBM high-resolution monochrome mode

When the operating system loads VBASICA, it checks to see whether or not a color card is installed and whether or not the bit map is allocated.

If a color card is installed, the VBASICA default screen is screen 20. If no color card is installed, the VBASICA default screen is screen 40.

If the bit map is not allocated, your programs cannot access graphics screens 42, 61, and 62.

After you set a screen type/mode combination using one of the two-digit screen numbers shown in Table 1, you can keep the same screen type but change the resolution mode by issuing a **SCREEN** command followed by the one-digit mode number (this procedure corresponds to the **SCREEN** command description in your VBASICA manual).

The following command, for example, switches to the standard screen in text mode:

SCREEN 40

You can then use the following command to switch to the standard screen in high-resolution mode:

SCREEN 2

If you want to run color graphics programs but do not have a color screen, issue the command **SCREEN 60** to simulate the IBM color screen on the standard screen.

Color Attributes

You can specify a color attribute with the graphics statements **PSET**, **PRESET**, **LINE**, **CIRCLE**, **PAINT**, and **DRAW**. The range is 0 to 3. These color attribute numbers are distinct from the numbers that refer to actual colors; the latter are used only as parameters in the **COLOR** statement.

On screen 61, 0 selects black; 1, 2, and 3 select varying shades of white.

In Mode 1 on the color screen (screen 21), 0 selects the background color; 1, 2, and 3 select foreground colors.

In Mode 2 (screen 22 or screen 62), 0 and 2 select black; 1 and 3 select white.

Note: The **COLOR** statement does not affect any graphics screen except screen 21.

Coordinates

The drawing statements PSET, PRESET, LINE, CIRCLE, GET, PUT, and PAINT require you to specify points on the screen as pairs of x-y coordinates. Specify coordinates in the format (x, y), where x and y are numeric expressions. The ranges of x and y values for each screen type/mode combination are shown in Table 2.

Table 2: Screen Coordinates

<u>Screen Type/Mode</u>	<u>X-Range (Horizontal)</u>	<u>Y-Range (Vertical)</u>
21, 61	0-319	0-199
42 (standard screen)	0-799	0-399
22, 62 (color screen)	0-639	0-199

The upper left corner of the screen is always point (0, 0).

When you clear the screen with either the SCREEN statement or the CLS statement, the graphics cursor is set to the middle of the screen. Table 3 gives the midscreen coordinates for each screen type/mode combination.

Table 3: Midscreen Coordinates

<u>Screen</u>	<u>Coordinates</u>
21, 61	(160, 100)
42	(400, 200)
22, 62	(320, 100)

4.2 Features Not Supported in This Version of VBASICA

The following features are not supported:

- ▶ The PAINT statement does not support tiling.
- ▶ The LINE statement does not support the style attribute.
- ▶ The PLAY statement does not support the incrementing and decrementing octaves option.
- ▶ User-defined trappable keys are not supported. Only keys 0 through 11 can be trapped.
- ▶ The MS-DOS PATH command and the use of pathnames in file specifiers are not supported.
- ▶ I/O redirection is not supported.

These statements, commands, and functions are not supported:

CHDIR command	PLAY STOP statement
ENVIRON statement	PLAY (n) function
ENVIRON\$ function	PMAP function
ERDEV function	RANDOMIZE statement
ERDEV\$ function	RMDIR command
IOCTL statement	SHELL statement
IOCTL\$ function	TIMER OFF statement
MKDIR command	TIMER ON statement
ON PLAY statement	TIMER STOP statement
ON TIMER statement	VIEW statement
PLAY OFF statement	VIEW PRINT statement
PLAY ON statement	WINDOW statement

4.3 Compatibility Problems and Resolutions

VBASICA is compatible with IBM BASICA, and programs written to run under IBM BASICA should run under VBASICA. There are, however, three differences between VBASICA and BASICA that you should be aware of.

4.3.1 Screen Compatibility

VBASICA supports three screen types, while BASICA supports one type. If you have a color screen, the difference in the screen types should not affect the operation of the BASICA program because the color screen is IBM compatible. VBASICA defaults to screen 20 (color screen, text mode) if you have a color card installed. Thus, as long as the program switches resolution modes when needed (SCREEN 0, 1, or 2), the program should run normally.

If you do not have a color screen, VBASICA defaults to screen 40 (VICTOR screen, text mode). Screen 40 is not an 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 do 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 resolution modes when needed (SCREEN 0, 1, or 2), it should run normally.

4.3.2 Hardware Compatibility

The second area of difference is in the hardware. If a program attempts to access directly an IBM hardware feature that is unavailable on the VICTOR 9000, the program will not run correctly under VBASICA. Known areas of hardware incompatibility are as follows:

- ▶ BLOADs of a binary file from a hard disk to the color card do not work. Only 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. The color card memory begins at &hB800:0.

BLOADs from diskette to the color card work correctly because they do not use DMA.

- ▶ 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 LOCATE statement

LOCATE LIN%, COL%, CRSR.ATTR%

the variable CRSR.ATTR% is a boolean value. 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

$1 \leq \text{CRSR.ATTR\%} \leq 255$

CRSR.ATTR% cannot be negative. To avoid problems, use +1 for true and 0 for false.

4.3.3 Function Keys

VBASICA supports a maximum of seven function keys. Therefore, if you have a BASICA program that uses function keys F8 through F10, you will have to rewrite the program to use function keys F1 through F7 only.