Contents

Installing and Using the Windows Version

Chapter 1	Installing the DOS/Windows Versions
Chapter 2	Using the Keyboard
Chapter 3	CRT and Graphic Drivers
Chapter 4	Printer and Pixel Image Device Drivers
Chapter 5	Graphic Input Drivers
Chapter 6	I/O Device Drivers
Chapter 7	Transferring Programs and Data from HP BASIC
Chapter 8	Customizing the Environment
Chapter 9	If You Have Problems
Chapter 10	Changes From Earlier Releases
Other Manuals	User's Guide Reference Manual

Chapter 1

Manual Organization

This Manual

Before Installing HTBasic

The README.TXT file

System Requirements

Installation

Installing Win32s

Installing on Multiple Computers

Running HTBasic

Drivers

Starting HTBasic

Automatically Running HTBasic

Leaving HTBasic

Command Line Switches

Alternate AUTOST Switch

Extended Output Area Buffer Switch

Colors Switch

Color Map Usage Switch

Font Switch

Window Geometry Switch

Graphics Buffering

Window Title Switch

Workspace Memory Switch

The AUTOST Program

Usage Notes

GESCAPE Window Manipulation

Long File Names

Limitations

Chapter 2

Keyboard Functions

Second Character

Ctrl Key

Softkeys

KBD CMODE

Alphabetical Keyboard Functions List

Softkeys

The Keypad

Functions invoked by the Alt Key

Other Function Keys

Additional Keyboard Features

Detailed Descriptions

ALPHA

ANY CHAR

BACKSPACE

BEGIN

BOL

CAPS LOCK

CLR->END

CLR I/O

CLR LN

CLR SCR

CLR TAB

CONTINUE

DEL CHR

DEL LEFT

DEL LN

DISPLAY FCTNS

DOWN

DUMP ALPHA

DUMP GRAPHICS

EDIT

END

ENTER

EOL

EXECUTE
GRAPHICS
HOME
s-HOME
INCR LABELS
INS CHR
INS LN
K0 to K23
KATAKANA
LEFT
MENU
NEXT
NEXT WORD
PAUSE
PREV
PREV WORD
PRT ALL
RECALL
RECALL NEW
RESET
RESULT
RIGHT
ROMAN
RUN
SELECT
SET TAB
STEP
STOP
SUB MODE
SYSTEM
TAB
TAB BACK
UP
USER

Chapter 3

The PLOTTER IS Statement

Loading Drivers

CRT Drivers

Default CRT Driver

CRTA and CRTB Modes

Graphic Drivers

Devices

Files

Driver Documentation

WIN Driver

Window Resize

Limitations

HPGL Driver

Plotting Area

Polygons

Options

Pen Colors

Drawing Mode

Line Thickness

Line Caps and Joins

Crosshatching

Pages

Ending Plots

PostScript Driver

Options

Pen Colors

Line Thickness

Line Caps and Joins

Crosshatching

Pages

Ending Plots

Chapter 4

The CONFIGURE DUMP Statement

DUMP Device Statement

Loading Drivers

Dumping Rows

Number of Colors

Options

WIN-DUMP Driver

Print Manager

DUMP Size

INVERT Option

Graphics Buffering

PCL Driver

Options

APPEND

ALPHA Dumps

Option Tables

OPTIONS for 640x480 Screen

OPTIONS for 1024x768 Screen

PS-DUMP Driver

Options

The APPEND Keyword

ALPHA Dumps

GIF Driver

Options

The APPEND Keyword

Chapter 5

The GRAPHICS INPUT IS Statement

Loading Drivers

KBD or ARROW KEYS Driver

KBDA Driver

HPGL Driver

TABLET Driver

Communication

ASCII Data Format

Binary Data Formats

Resolution

Option Configuration

HP Mode

Summagraphics MM1103 Emulation

Hitachi HDG-1111B Emulation

Chapter 6

WIN-PRINT Driver

Printing to the Printer

Printer Control

Selecting a Printer

WIN-PRINT CONTROL Registers

WIN-PRINT STATUS Registers

Serial (RS-232) Driver

Loading

Options

Modes of operation

Interrupts

READIO/WRITEIO

IEEE-488 Drivers

Compatibility Table

Loading

Multiple GPIB Cards

Options

BOARD Option

BASE Option

INTERRUPT Option

ISC Option

DMA Option

SYSTEM and NOTSYS Options

Micro Channel Slot Number

LPT Option

FAST Option

TransEra's HM900 Board

I/O Address

TransEra IEEE-488 Card I/O Address Table

Interrupt and DMA Jumpers

DMA Channel

System Controller

HP SICL Driver -HPIBS

Choosing a GPIB Interface Driver

Features of the HPIBS Driver

Features of the GPIBH Driver

HPIBS Driver Options

Interface Registers

Unsupported Keywords

NI's API Driver - GPIBTNT

PCMCIA Cards

National Instruments Cards

INES cards

Chapter 7

File Types

Typed Files

Ordinary Files

Remote Access

Mounted Network Volumes

Network File Transfer

Other File Transfer Utilities

Files Created by Other Computers

LIF ASCII Files

BDAT Files

PROG Files

Ordinary Files

Problems

HPCAT

HPCOPY

HPPURGE

Chapter 8

Softkey Macros

Assigning Softkeys

Editing Softkeys

Reading Softkeys

Deleting Softkey Macros

Listing Softkey Macros

Storing Softkey Macros

Loading Softkey Macros

CONFIGURE Statement

HP Compatible BDAT Files

Controlling File Header Type

Controlling Implicit Dimensioning

DUMP GRAPHICS Printer Type

Non-Latin-1 Character Set Keyboard Remapping

Redefining Function Keys

Table for Shift-mask and Shift-value Calculation

Defining New LABEL Characters

Enabling Use of Long File Names

HP MSUS to Windows disk substitutions

Changing the value of PRT

Windows Compatible ASCII Files

Chapter 9

Keyboard Lockup

Graphics Disappear

Error 163, Interface not present

Softkeys Not Visible

Colors Wrong or Dithered

Cannot be Run in DOS Mode

Bad Command Error

Error larger than 10000

Error 2009, Wrong Revision

Chapter 10

Changes_From_5.0_to_6.0
Changes From 4.0 to 5.0
LIF Utilities
Numeric Compiler
HP BASIC Files
Changes From 3.0 to 4.0
New HP BASIC 6.x features
New Features from HP BASIC/UX
New features Unique to HTBasic
Other Enhancements
Changes From 2.0 to 3.0
PLOTTER IS Change
Power-PROG Format
CONFIGURE
Additions and Enhancements
Diferences with RMB
I/O Differences
Mass Storage Differences
MSUS Format
RMB CSUB
Performance Tuning:

Chapter 1
Installing the Windows Version

This manual explains how to install and run the Windows Version of HTBasic and how the documentation is organized. There is a file named README.TXT on the distribution media. The README.TXT file contains up-to-date information about HTBasic, manual corrections, and compatibility problems. Read it now, then read this chapter, and then install HTBasic.

Manual Organization

The HTBasic manual set consists of the *User's Guide*, the *Reference Manual,* and the *Installing and Using the Windows Version* manual.

Conventions. The following terms are used in the following way in the HTBasic manuals: Rocky Mountain BASIC (RMB) is a dialect of BASIC. HTBasic and HP BASIC are particular implementations of RMB.

Several conventions are popular for denoting a hexadecimal number: &H2A0, 2A0h and 0x2A0. In the HTBasic manual set we use the first, &H2A0.

Several printing conventions are used in this manual. Keywords, keyboard key names, IEEE-488 bus commands and operating system commands are shown in all uppercase, for example: ENTER. Examples that show exactly what you should type or what the computer displays are shown in a fixed width font, such as

```
10 DIM A$[50]
```

Italics are used to denote words where you substitute your own input in place of what is printed:

`LOAD "`*`prog_name`*`"`

User's Guide. The *User's Guide* contains in-depth information about using the HTBasic language. It is arranged topically. Chapter 2 of the *User's Guide*, "Porting HP BASIC Programs to HTBasic" is a porting guide for users with HP BASIC programs. If you are porting existing Rocky Mountain BASIC programs, this chapter is very important. Other chapters define the elements of the HTBasic language, program flow statements, mathematics, graphics and I/O. Chapters are devoted to use of the CRT, keyboard, files, IEEE-488 (HP-IB) and other interfaces. The *User's Guide* also contains chapters about international language support and mixed language programming for the DOS Version of HTBasic.

Reference Manual. The *Reference Manual* consists mainly of a dictionary style presentation of HTBasic keywords. The *Reference Manual* also is available on-line through the HELP statement. The manual includes a chapter of definitions, a statement summary table, a list of error messages and an ASCII code chart. The ASCII code chart contains ASCII, decimal and hexadecimal values and IEEE-488 commands and addresses.

This Manual

This manual is the *Installing and Using the Windows Version* manual. It is specific to the Windows Version of HTBasic. It explains how to install and configure HTBasic. HTBasic is highly configurable, so it is extremely important that you read the appropriate chapters.

Chapter 1, "Installation," contains the information needed to get HTBasic running. All users should read this chapter.

Chapter 2, "Using the Keyboard," describes the dozens of special keyboard function keys that are defined. All users should read this chapter.

Chapter 3, "CRT and Graphic Drivers," explains how to load CRT drivers, plotter drivers and vector file drivers. Users should read the sections in this chapter that refer to drivers they will be using.

Chapter 4, "Printer and Image File Drivers," explains how to load printer drivers and pixel image file drivers. Users should read the sections in this chapter that refer to drivers they will be using.

Chapter 5, "Graphic Input Drivers," explains how to load Graphic Input drivers for digitizing. If you are using digitizers, read this chapter.

Chapter 6, "I/O Device Drivers," explains how to load drivers for IEEE-488 (GPIB or HP-IB), GPIO and RS-232 (Serial) interfaces. If you are using IEEE-488, GPIO, serial or other I/O devices, check this chapter.

Chapter 7, "Transferring Programs and Data From HP BASIC," explains how to move over and use existing Rocky Mountain BASIC programs and data. If you need to transfer existing programs or data over, read this chapter.

Chapter 8, "Customizing the Environment," talks about softkey macros and creating HP workstation compatible files. It explains controlling implicit variable and string dimensioning and customizing keyboard key assignments. Defining additional LABEL characters is explained as well as specifying a Windows path specifier to use in place of an HP mass storage unit specifier. The chapter describes changing the value of PRT and creating Windows ASCII program files.

Chapter 9, "If You Have Problems," describes most of the common problems that users have while getting up and running. Often, you can save yourself a phone call by checking here.

Chapter 10, "Changes From Earlier Releases," details the significant changes from Release 2.0 to 3.0 to 4.0 to 5.0 to 6.0. Also included here are the significant differences between HTBasic and HP BASIC series 200/300/700 (RMB). This is optional reading for most users. If you are upgrading and something seems to have changed, this is the place to check for more information.

Before Installing HTBasic

Note: In the HTBasic manual set, the term "disk" may apply to either floppy diskettes or hard disks. The term HTBWin is sometimes used in place of HTBasic for Windows. Also, in this manual the term Windows sometimes applies only to Microsoft® Windows™ and at other times is used to mean both Microsoft® Windows™ and Microsoft® Windows NT™.

The README.TXT file

If you have not yet read the README.TXT file, you should do so now. It can be displayed on the computer screen by double-clicking on the icon next to the README.TXT or Readme filename in a folder or in file manager.

System Requirements

A 386/486/Pentium class processor or better is required. A processor with built-in numeric processor (486, Pentium) or an external numeric coprocessor (387SX, 387DX, 487) is strongly recommended. Four megabytes of memory are required, but 8 megabytes are recommended. HTBWin will run with a monochrome, 16-color, 256-color or 24-bit display driver, but a 256-color display driver is recommended. COLOR MAP operations function only with a 256-color display driver. HTBasic can send screen dumps to any graphic printer supported by Windows. Plotters must be HPGL or PostScript compatible. HTBWin is designed to run under Windows 95. While it will run under Windows NT or Windows 3.x (with Win32s installed), some features are not supported.

Installation

To install HTBasic, first place the HTBasic distribution CD-ROM into a CD-ROM drive. Double-click on "My Computer." (Users of Program Manager can open the Main program group and double-click on File Manager, instead.) Click on the icon for the CD drive, double-click on the program named "Setup" or "setup.exe".

Follow the instructions on the screen to install HTBasic. You will be asked for your HTBasic Serial Number which can be found on your CD jewel case. Correct input of this number is required to complete installation This number will be stored on your system and will be returned by the command:

```
SYSTEM$("SERIAL NUMBER")
```

You will also be given an opportunity to select which components to install.

Installing Win32s

For full functionality, HTBWin is designed for use with Windows 95 or later. It can be used under Windows 3.1 or 3.11 if Win32s is installed. This is done automatically for you during installation. If you should need to re-install this component you may follow the instructions below.

Note: If installing Win32s on Windows 3.1, it is necessary to run the MS-DOS Share utility before starting Windows. Add SHARE.EXE to your AUTOEXEC.BAT file. SHARE.EXE is not required for Windows for Workgroups. Also, it is recommended that your CONFIG.SYS file set number of **files** to at least 30. Set **files=30** in CONFIG.SYS if there is no **files** command line or if it specifies less than 30 files.

Disabling Win32s. It should not be necessary to disable Win32s. The Win32s DLLs will only be loaded when a Win32 application is executed. The Win32s VxD is loaded when Windows starts, but has little memory overhead. If you must disable Win32s or wish to do a clean reinstallation of Win32s, take the following actions:

1. Remove the Win32s VxD line from Windows SYSTEM.INI file in the [Enh386] section:

```
device=c:\windows\system\win32s\w32s.386
```

2. Delete the WIN32S.INI, W32SYS.DLL and the WIN32S16.DLL files from the *windows*\SYSTEM directory and all files in the *windows*\SYSTEM\WIN32S subdirectory (*windows* is the Windows installation directory such as C:\WINDOWS.)

3. Restart Windows.

Installing on Multiple Computers

HTBasic is not copy protected. You may make backups of HTBasic and install HTBasic on more than one computer. However, HTBasic is copyrighted, protected by international treaty and provided to you only under license. You have license to use HTBasic on only one computer at a time. The serial number on your jewl case encourages compliance with the license.

To install HTBasic on more than one computer, repeat the installation instructions given above.

Running HTBasic

The following sections explain several topics. One section explains loading device drivers. Others explain how to start HTBasic manually or automatically, how to terminate HTBasic and command line switches that can be used to modify the behavior of HTBasic.

Drivers

HTBasic supports drivers for different interfaces, I/O ports, displays, tablets, plotters, printers and graphic output files. HTBasic automatically attempts to load the correct display driver. Other drivers should be specified in the AUTOST file or on the command line, as explained later in this chapter.

For example, if you plan on using an IEEE-488 card or a serial (RS-232) interface you should modify your AUTOST program to load the needed drivers. Chapters 3 to 6 contain instructions for loading and customizing drivers. If no driver is loaded, a statement such as

```
OUTPUT 719;A$
```

will produce error 163, Interface not present.

Also, if you plan on using plotter or printer drivers, the AUTOST file is a good place to put the PLOTTER IS or CONFIGURE DUMP statements needed to load the drivers.

Starting HTBasic

HTBasic is normally started by double-clicking on the HTBasic icon. The location of the icon varies depending on the version of Windows and the tool used to find the icon. Sections are given below for users of Windows 95, users of Program Manager and users of File Manager.

Windows 95 In Windows 95, click on Start, Programs, TransEra HTBasic, and then on the HTBasic icon:

The HTBasic icon can be found in other places and shortcuts can be made. Double-click on "My Computer", then the drive on which HTBasic is installed, then the Windows folder, Programs folder, TranEra HTBasic folder, and finally the HTBasic icon. Alternately, double-click on "My Computer", then the drive on which HTBasic is installed, then the HTBWin folder, and finally the HTBasic icon (named HTBWin or HTBWin.exe).

Shortcuts can be made from the HTBasic icon located using either method of the previous paragraph. Using the right mouse button, drag a copy of the HTBasic icon to either the desktop or the Start button. Release the mouse button and select "Create Shortcut(s) Here." If you created the shortcut on the desktop, start HTBasic by double-clicking on the HTBasic icon on the desktop. If you created the shortcut on the start button, start HTBasic by clicking on Start and then on the HTBasic menu item.

MS-DOS Shell. HTBWin can also be started (except in Windows 3.x) from an MS-DOS window by typing:

```
C> \htbwin\htbwin
```

Program Manager Program Manager can be used to start HTBasic. Open the "TransEra HTBasic" program group and double-click on the HTBasic icon.

File Manager In File Manager, click on the drive icon for drive c. (If you installed on another drive, click the drive letter you used.) Click on the htbwin folder, then double-click on htbwin.exe.

If you have problems trying to run HTBasic, read the information in Chapter 9, "If You Have Problems."

Automatically Running HTBasic

Windows can be set up to run HTBasic automatically when Windows starts. When Windows starts, it automatically runs any programs found in the StartUp folder (or Windows 3.x program group). For Windows 95, hold the right mouse button down while dragging a copy of the HTBasic icon from the "TransEra HTBasic" folder to the StartUp folder. Release the mouse button and select "Create Shortcut(s) Here."

For Windows 3.x, to place a copy of the HTBasic or HTBWIN.EXE icon in the StartUp program group, hold the Ctrl key and left mouse button down while dragging a copy of the icon from the "TransEra HTBasic" program group to the StartUp program group. Release the mouse button to copy the icon.

The next time you start Windows, it launches starts HTBasic with the command line switches specified, in the directory specified. HTBasic then executes the AUTOST file, if one exists in the start-up directory. No attempt is made to explicitly restore a program or data that was in memory when you shut down Windows. You can customize the AUTOST file to perform any desired initialization and program loading.

Leaving HTBasic

Use the QUIT statement to leave HTBasic. It can be entered for immediate execution from the keyboard if a program is not running or it can be executed from within a running program. This command closes any open files, shuts down HTBasic and returns to Windows. Alternately, any normal method for closing a Window will also cause the QUIT statement to execute. If you wish to use Alt-F4 to close the window, the standard HTBasic definition for Alt-F4 (PAUSE) must first be removed. See "Redefining Function Keys," in Chapter 8, "Customizing the Environment."

Command Line Switches

Optionally, one or more command line switches can be specified when starting HTBasic. Command line switches affect the behavior of HTBasic. Usually, no switches are necessary. Read through the headings to identify any you may need to use.

To set the command line switches and the inital MSI (directory) used by HTBasic, change the Properties associated with the icon used to start HTBasic.

For Windows 95 and later, click on the HTBasic icon with the right mouse button and select Properties....

In Program Manager, double-click on the "TransEra HTBasic" program group, single-click on the HTBasic icon, select the File menu, and then the Properties... menu item. After setting the desired options, select OK.

To specify switches when starting HTBasic in an MS-DOS shell, use this syntax:

htbwin [-switch...]

HTBasic command line switches can be abbreviated to the shortest unique abbreviation. The following paragraphs document the individual switches. Examples are given as if the switches were used in an MS-DOS shell. To use the switch in a Properties dialog box, leave off "htbwin" at the beginning. Separate multiple switches with spaces.

Alternate AUTOST Switch

The **-ALT** switch can be used to run an AUTOST program other than the one named "AUTOST" in the current directory. For example, the following statement will start HTBasic running and will use a file named STARTUP.BAS in the root directory of drive C, regardless of what the current directory is:

```
C> HTBWin -ALT C:\STARTUP.BAS
```

To start HTBasic without running any files, direct the -ALT switch to a non-existant file name for example:

```
C>  HTBWin -ALT junk
```

Extended Output Area Buffer Switch

The **-buf** switch specifies the number of lines to reserve for the Extended Output Area. The visible portion of the screen is called the Output Area. HTBasic saves lines that scroll off the top of the screen, effectively increasing the number of screen lines. The buffer containing the on- and off-screen lines of text is called the Extended Output Area buffer. The -buf switch determines the number of lines in the buffer. The default is 160. For example, if you have 25 lines on-screen but wish to scroll back and look at 100 previous lines, use this command:

```
C> HTBWIN -buf 125
```

Colors Switch

The **-colors** switch specifies the number of colors you wish to work with. The number specifies the size of the color map to use. If the color map usage method is Share (as explained below), the number must be less than the number of system palette entries available. Usually, a 256-color windows driver has 236 entries available. A 16-color driver has no entries available. If the color map usage method is ReadOnly, the number can be 1 to 256.

On "True Color" displays, HTBasic functions slightly differently. True color displays are 32768 (15-bit), 65536 (16-bit) and 16777216 color (24-bit) displays. No "physical" color map is present. On True Color displays, HTBasic creates a "logical" color map with no more than 256 entries (PENs). The limit of 256 pens does not limit the number of colors displayable since all the colors are available by assigning different subsets to the PENs at different times. Also, the AREA COLOR/INTENSITY statement can be used in place of AREA PEN statements to access all the colors, regardless of the current PEN colors.

On a display with a color map, when a color map entry is changed through the SET PEN command, all pixels already on the screen written using that color map entry change color. This is sometimes used to give animation to a picture on the display. On true color displays, the colors of pixels already on the screen do not change when an entry in the logical color map is changed. This allows an image to contain more than 256 colors but does not permit the type of animation mentioned.

The -colors switch is often used in conjunction with the -cu switch, described below. If not specified, sixteen colors or less are used depending on the number of colors possible on the display. An example of the switch usage is:

```
C> HTBWIN -colors 16 -cu Share
```

Color Map Usage Switch

The **-cu** switch specifies the Color Map Usage Method. When the -cu switch is not specified, HTBasic defaults to Shared if the Windows display driver supports a writable color map and to ReadOnly if not. Sixteen color drivers such as the standard VGA driver generally do *not* support a writable color map. Usually, a 256-color driver must be used to obtain a writable color map. Note that in Windows, the term "palette" is often used instead of "color map." The syntax is:

-cu { Share | ReadOnly }

Share. When the -cu switch is not specified and the display driver supports a writable color map, the Share color map usage method is used. Share means that HTBasic will share the default color map with other running applications. HTBasic attempts to allocate the number of requested colors from unused entries in the color map. Since the entries exist at random in the color map, color combinations resulting from Normal and Alternate drawing modes are undefined. SET PEN can be used to change color map entries. During a full-screen GLOAD, any colors in the image not present in the current COLOR MAP are changed to the nearest colors in the COLOR MAP.

This example explicitly specifies the Share color map usage method:

```
C> HTBWIN -cu Share
```

ReadOnly. When the -cu switch is not specified and the display driver does not support a writable color map, the ReadOnly color map usage method is used. This typically occurs on 16-color and 24-bit color display drivers.

Sixteen and 24-bit/pixel color modes (also called High or true Color) are supported through the ReadOnly color use method ("-cu ReadOnly" command line switch). However, HTBasic for Windows can only use a maximum of 256 colors at any one time. When using this mode, no color aliasing or flashing will occur as other windows are selected. DUMP GRAPHICS is not supported in these modes.

Color combinations resulting from Normal and Alternate drawing modes are undefined. While color map statements can be used, since no physical color map is present the color produced will be the closest available color (sometimes by dithering) and the color of pixels already on the screen do not change. During a full-screen GLOAD, colors are changed to the nearest colors in the Windows static colors.

To obtain the maximum number of solid colors, assign Window's static colors to the HTBasic pens. The following program does this by changing pens 8 to 15, since the default colors of HTBasic pens 0 to 7 are already Windows static colors. The WIN-DUMP driver requires this color map to prevent color loss during a dump.

```
1!RE-SAVE"WINCLR.BAS"
10    PLOTTER IS CRT,"INTERNAL";COLOR MAP
20    DATA .5,.5,.5    ! 8  = dark grey
30    DATA .75,.75,.75 ! 9  = light grey
40    DATA .5, 0, 0    ! 10 = dark red
50    DATA .5,.5, 0    ! 11 = dark yellow
60    DATA 0,.5, 0     ! 12 = dark green
70    DATA 0,.5,.5     ! 13 = dark cyan
80    DATA 0, 0,.5     ! 14 = dark blue
90    DATA .5, 0,.5    ! 15 = dark magenta
180   !
190   DIM Palette(7,2)
200   READ Palette(*)
210   SET PEN 8 INTENSITY Palette(*)
220   END
```

Private. The Private color map usage option provides a method to reserve colors exclusively for HTBasic for Windows use. The benefit of the Private mode is that when changing a pen color (SET PEN ...), all graphics currently displayed that use this pen will change to the new color. When using the Shared or

ReadOnly modes, the new pen color will only apply to subsequent graphics statements. The Private mode most closely emulates the mode used in HP BASIC 300/700.

Font Switch

The **-fn** switch specifies a Windows font to use in place of the default. To specify a point size, give the font name, a comma and the point size. If the font name contains a space, place quotes around it. If no font switch is used, the Windows Version uses the FixedSys font. The syntax is:

```
HTBWIN -fn "font_name",size
```

Only fixed width fonts should be used. Although most Windows fonts represent the ISO 8859 (Latin 1) character set, not all do. If the keyboard is set to one character set and the screen font uses a different character set, it is possible for non-ASCII characters to display incorrectly or not at all.

If size is positive, the spacing is in pixels; if negative, the spacing is in points (1/72 inch increments). Windows 95 can remap character sets and can use TrueType fonts so any fixed-width font with the required characters may be used as a code page 437 font. For example:

-fn "Courier New",-12,255

selects a code page 437 mapping of the built-in Courier New font at 12-point size. Windows 3.x cannot remap existing fonts. However, you can install a non-Latin-1 font using the Windows 3.1 Control Panel, Fonts item. Then specify the -fn command line switch.

There are some fonts included with HTBasic, they may be found in the Lexical directory. These fonts are not hinted. A fully hinted font with code page 437 character mapping from Microsoft and Monotype can be found at the web site www.winsite.com in a file: gc0651.exe.

SYSTEM$("FONT") returns the name of the Windows font in use.

Window Geometry Switch

The **-geometry** switch specifies the size and optionally the position of the HTBasic window. The syntax is:

-geometry *WIDTH*x*HEIGHT*[+*XOFF*+*YOFF*]

where *WIDTH*, *HEIGHT*, *XOFF* and *YOFF* are numbers and only *WIDTH*x*HEIGHT* must be specified. *WIDTH*x*HEIGHT* specifies the size of the HTBasic window and can be specified in either pixels or characters, although the same units must be used for both. If the height is less than 100, the units are interpreted as characters (columns by rows), otherwise the units are interpreted as pixels. *XOFF* and *YOFF* specify the position of the HTBasic window on the display relative to the upper-left corner of the display. *XOFF* and *YOFF* are always specified in pixels. By default, HTBasic creates a window that fills the screen.

The first example below creates a window that is 80 columns by 25 lines and the position is set by the window manager. The second example creates a window that is 1024 by 768 pixels, positioned 100 pixels from the top of the screen and 200 from the left edge.

```
C> HTBWIN -geometry 80x25
C> HTBWIN -geometry 1024x768+200+100
```

Graphics Buffering

The **-graphics_buffer** switch specifies whether graphics drawn into the window are buffered are not. The syntax is:

-gr[aphics_buffer] { on | off }

By default, graphics buffering is on. When buffering is on, the graphics are restored, however, the speed of screen output is severely impacted. When buffering is off, all graphics are lost if the HTBasic window is minimized or placed behind another window.

To disable buffering, use the switch:

```
C> HTBWIN -gr off
```

Window Title Switch

The **-title** switch specifies the title displayed in the title bar of the HTBasic window and the minimized icon. If the title contains spaces, place quotes around it. The syntax is:

-title *NAME*

where *NAME* is the title string.

```
C> HTBWIN -title "Furnace Test System"
C> HTBWIN -title Controller15
```

Workspace Memory Switch

The **-w** (workspace) switch specifies how much memory to set aside for your programs and data. The syntax is:

-w *amount*[k|m]

where *amount* should be replaced with a number specifying the amount of memory. *Amount* can optionally be followed by a "k" or an "m". If no "k" or "m" is given, the number specifies bytes. If "k" is given, the number specifies kilobytes and if "m" is given, the number specifies megabytes. *Amount* cannot include a period (i.e. 2.4m).

The default workspace size is 16 megabytes. Note that the amount of free memory reported can be somewhat less than that requested because device drivers or other memory users may allocate some of the memory during startup. The following example allocates thirty-two megabytes:

```
C> HTBWIN -w 32m
```

The AUTOST Program

Each time HTBasic starts, it checks the current directory for an AUTOST program file. If one is found, it is automatically loaded and executed. This allows you to load any necessary device drivers, customize HTBasic using any of the programmable statements and start any default application or menu program you desire.

To customize the AUTOST program, you should:

```
LOAD "AUTOST"
EDIT
    (Make your changes)
RE-STORE "AUTOST"
```

Usage Notes

The following short topics contain important information about using HTBasic for Windows.

GESCAPE Window Manipulation

Several GESCAPE codes allow manipulation of the HTBasic window.

Code	**Operation**
30	Maximize the window
31	Hide the window
32	Restore the window
33	Set window position and size
34	Get window position and size
35	Bring the window to the top
36	Get the screen size
38	Hide / Restore the title bar
37	Returns the Title Bar enable flag
39	Set the DUMP size (% of paper width)
41	Minimize the window

The following example shows the syntax for each of the new GESCAPES. Note that codes that set information have a comma before the array name while codes that get information have a semicolon.

```
10   INTEGER Get4(1:4),Set4(1:4),Get2(1:2),Set1(1:1)
20   DATA 90,100,500,300    ! Position of upper left corner:
30   !                        (90,100), Width = 500, Height = 300
40   READ Set4(*)
50   GESCAPE CRT,30         ! Maximize the window
60   GESCAPE CRT,31         ! Hide the window
70   GESCAPE CRT,32         ! Restore the window
80   GESCAPE CRT,33,Set4(*) ! Set position and size: X,Y,W,H
90   GESCAPE CRT,34;Get4(*) ! Get position and size: X,Y,W,H
100  GESCAPE CRT,35         ! Bring the window to the top
110  GESCAPE CRT,36;Get2(*) ! Get the screen size: W,H
120  GESCAPE CRT,37;Get3(*) ! Get the title bar enable flag
130  PRINT Get(2)           ! Print the Screen Size
140  PRINT Get(3)           ! Print the title bar enable flag
150  Set1(0)=50             ! Set the DUMP size to 50%
160  GESCAPE CRT,38         ! Hide window Title Bar
170  GESCAPE CRT,38         ! Restore window Title Bar
180  Set (1)=50             ! Set the DUMP size to 50%
190  GESCAPE CRT,39,Set1(*) ! Set the DUMP size (default is 100%)
200  GESCAPE CRT,41         ! Minimize the window
210  GESCAPE CRT,32         ! Restore the window
220  END
```

Long File Names

In newer versions of DOS and Windows, long filenames are allowed in addition to the standard 8.3 names. The filenames can be about 256 characters long and can have embedded spaces. However, by default HTBasic removes spaces from file specifiers and CAT listings don't have enough room for long filenames. To enable display and use of long filenames, use the statement

```
CONFIGURE LONGFILENAMES ON
```

With LONGFILENAMES ON, spaces are not deleted from directory and file specifiers since they may be significant. Also, the listing format for CAT is changed to accommodate varying length filenames. It is roughly modeled after the Windows NT DIR command listing format. Also, UNC network names can be used.

Limitations

Device Drivers. Operating system device drivers whose functionality is presented through standard file system commands are usable from HTBasic. Drivers that must be accessed through other API's are usable by writing DLL's and producing CSUB's with the separately available HTBasic for Windows CSUB Toolkit.

General. Peek and Poke (through READIO and WRITEIO) are not allowed. The state of the Caps Lock and the keyboard delay and repeat rate cannot be changed from HTBasic. The WAIT statement can wait much longer than requested. The extra amount seems to be in the order of one second.

Due to the resolution of the PC clock (about 55 milliseconds), all timing functions (WAIT, ON CYCLE, ON DELAY, ON TIME, ON TIMEOUT, etc.) will be rounded to the nearest clock interval.

The LINE TYPEs supported on the display by this release are limited. Line type 3 is drawn the same as 4, and 6 is drawn the same as 7. The optional repeat length is also not supported on the display.

The SEPARATE ALPHA FROM GRAPHICS statement is not supported. The DUMP GRAPHICS and some GSTORE commands do not work if the display is configured for more than 256 colors (e.g. "High Color" or 24-bit color).

HTBasic for Windows cannot directly access HP LIF format diskettes. Use the HPCAT, HPCOPY, and HPPURGE utilities to access HP LIF format diskettes.

Running more than one instance of HTBWIN at a time is not recommended under Windows 3.X. (This is supported on Windows 95 and NT.)

Because of a Microsoft math library error, values between "xx495" and "xx500" are rounded up instead of down. For example: DROUND(12.3499,3) returns 12.4.

The SET and RETURN clauses of the ASSIGN, DIALOG, CONTROL, and STATUS statements are limited to 36 attribute pairs.

In Windows 3.1x, recursive function calls (DEF FNxxx) are not allowed.

When plotted labels are rotated with LDIR, isotropic scaling of the label may not be maintained on some plotting devices. If you attempt to CAT a networked directory containing a file named "CON", HTBWIN will hang.

TRANSFER and related statements are not supported. Multiple window statements are not supported: CLEAR WINDOW, CONTROL/STATUS window, CREATE WINDOW, DESTROY WINDOW, DUMP window, LIST WINDOW, MOVE WINDOW, PLOTTER IS window, SCRATCH WINDOW/W, SYSTEM$ ("WINDOW SYSTEM").

Windows Limitations. The Windows environment has several limitations that affect HTBasic. Windows does not support graphic planes and thus SEPARATE ALPHA and MERGE ALPHA is not supported.

Windows 3.x. Running under Windows 3.1 requires Win32s, as explained earlier. Some features are not supported in this environment. Any drawing operation that draws a single point does not work. This includes LABEL characters.

Long filenames are not supported. If a statement that attempts to load a driver or binary (PLOTTER IS, CONFIGURE DUMP, GRAPHICS INPUT IS, LOAD BIN) specifies a driver that is not present, Windows 3.1 displays a "File Error" dialog box. The dialog box can be dismissed by clicking on OK or pressing the Enter key. Only the default font will print correctly when using ISC 10 (WIN-PRINT). A stack overflow error can not be handled by an ON ERROR handler; the program is always stopped.

Windows 95. Windows 95 is the target environment for HTBWin; virtually all features are supported in this environment.

Windows NT. Windows NT is a protected environment that prevents some features of HTBasic from working. The LIF utilities LIFCAT, LIFCOPY, and LIFPURGE are not supported on Windows NT. A 486/66Mhz is the minimum supported NT system.

Due to Windows NT's enhanced hardware protection, minimal device drivers are curently supported on Windows NT. The "GPIB/GPIBN/GPIBH" drivers are NOT supported on Windows NT, use GPIBTNT and HPIBS instead. Check each section for specific information regarding NT and the device driver you wish to use.

The "performance register" (CONTROL KBD,202) can influence the synchronization between I/O operations and display updates. If KBD,202 is set to a high value (such as the default of 32), then the CRT may not display any of the output statements until the current statement is completed. At that point, it will "flush" all of the PRINT text at one time. However, low values of KBD,202 have a negative impact on HTBWIN performance, even on Windows NT. Do not use a value of 1 unless this particular synchronization is much more important than the over-all performance of your application.

Chapter 2
Using the Keyboard

This chapter explains how the different keyboard functions available in HTBasic are invoked. The last half of the chapter gives detailed descriptions of each function.

Keyboard Functions

Each keyboard function has a generic name by which it is referred to in all the manuals. To access the function, you must know which key to press on your keyboard. The generic names may or may not match the physical labels printed on the keyboard keys. Use this chapter, or any of the following, to look up key assignments:

- The system softkey menu
- The "HELP #" command where # is a BASIC keyword

Each of these methods allows you to find quickly which keys correspond to which functions. The assignments are logical and easy to remember.

Second Character

Internally, each keyboard function is represented by two characters. The first, CHR$(255), tells HTBasic that a function key is being pressed. The second character identifies the function. A program can execute any of the function keys by outputting the two characters to the keyboard (See the *User's Guide*, Chapter 8, "CRT, Keyboard and Printer."). The second character for each function is listed in the last column of the alphabetical keyboard functions table.

Ctrl Key

If the Ctrl key is pressed while a function key is pressed, the function key is not executed, but is entered into the keyboard buffer. This is useful when defining keyboard macros that expand into several key presses and when composing OUTPUT KBD statements to execute function keys from a program.

Softkeys

A softkey is a function key whose function can be changed under software control. HP BASIC defines 24 softkeys. Softkeys are programmed with the ON KEY statement to provide convenient user/program interaction. When not used by an ON KEY statement, a softkey can be assigned a keyboard macro (or, as HP calls it, a "Typing Aid"). A macro is a key that is assigned one or more keystrokes; thus by pressing one key you can mimic pressing several keys. Softkey Macros are explained in Chapter 8, "Customizing the Environment."

A softkey menu is displayed at the bottom of the screen. The labels in the menu are numbered to correspond to the numbers printed on the function keys. The label marked "1" corresponds to the F1 key. The number is not meant to be the softkey number, but is printed to help the user locate the correct key to press. For example, the first label is marked "1" and corresponds to the F1 key, though depending on KBD CMODE (explained below), it might be softkey K0 or softkey K1.

If your HP computer does not display the softkey menu before a program defines any softkeys, then it does not have any keyboard macros defined or does not have the KBD binary loaded. To set HTBasic to this condition, execute a SCRATCH KEY statement in your AUTOST file.

KBD CMODE

Over the years, HP BASIC has supported two major softkey layouts: ITF and Nimitz. HTBasic can run programs written for either keyboard layout, regardless of the physical keyboard in use. The KBD CMODE statement selects the layout to use. When developing new programs, either style may be used at your preference.

Use KBD CMODE OFF for programs written for the ITF keyboard (46021A). With KBD CMODE OFF, eight function keys act as softkeys. The softkey labels are displayed at the bottom of the screen in two groups, four on the left and four on the right. Each label is eight characters wide and two lines high. With KBD CMODE OFF, the softkeys do quadruple duty. Each softkey has four meanings, depending on which softkey menu is active when the key is pressed. The four menus are System, User 1, User 2 and User 3. An indicator is displayed immediately above the softkey menu to show which menu is active. Another function key is used to cycle through the menus. KBD CMODE OFF is the default softkey mode.

Use KBD CMODE ON for programs written for the Nimitz keyboard (98203). With KBD CMODE ON, ten function keys act as softkeys. The softkey labels are displayed at the bottom of the screen in two rows. Each row contains five labels and each label is 14 characters wide.

Alphabetical Keyboard Functions List

The following table lists alphabetically all the keyboard functions available in HTBasic, the generic names and the key pressed on a PC keyboard to invoke that function.

Keyboard Function	Generic Name	PC Key	2nd Char
Alpha screen	ALPHA	Alt-F6	M
Any character input	ANY CHAR	Alt-= or Alt-K	$
Backspace	BACKSPACE	not assigned	B
Begin of line	BOL	Shift-←	H
Begin of output area or program	BEGIN	Shift-↓	T
CAPS state toggle	CAPS LOCK	Caps Lock	U
Clear alpha screen	CLR SCR	Home	K
Clear I/O	CLR I/O	Alt-F5	I
Clear line	CLR LN	End	#
Clear tab under cursor	CLR TAB	Alt-C	[
Clear to end of line	CLR->END	Shift-End	%
Continue program	CONTINUE	Alt-F2	C
Delete left of cursor	DEL LEFT	Backspace	.
Delete program line	DEL LN	Shift-Del	/
Delete under cursor	DEL CHR	Del	-
Display functions	DISPLAY FCTNS	Alt-F	F
Dump alpha screen	DUMP ALPHA	Alt-A	O
Dump graphics screen	DUMP GRAPHICS	Alt-G	N
"EDIT" key macro	EDIT	Alt-E	D
End of line	EOL	Shift-→	G
End of output area	s-HOME	Shift-Home	_
End of output/program	END	Shift-	W
Enter	ENTER	Enter	E
Execute	EXECUTE	not assigned	X
Graph screen	GRAPHICS	Alt-F7	L
Home position	HOME	not assigned	\
Increment menu labels	INCR LABELS	Shift-F9	~
Insert program line	INS LN	Shift-Ins	*
Insert/replace	INS CHR	Ins	+
Katakana mode	KATAKANA	not assigned	J
Left	LEFT	←	<
Menu labels on/off	MENU	F9	|
Next line	UP		^
Next word	NEXT WORD	Alt-→	Z
Pause program	PAUSE	Pause or Alt-F4	P
Previous line	DOWN	↓	V
Previous word	PREV WORD	Alt-←	Q
Print all output	PRT ALL	Alt-P	A
Recall more recent line	RECALL NEW	Shift-PgDn	@
Recall older line	RECALL	Shift-PgUp	?
Reset BASIC	RESET	Ctrl-Break	none
Result of last calculation	RESULT	Alt-R	=
Right	RIGHT	→	>

Roman	ROMAN	not assigned	Y
Run program	RUN	Alt-F3	R
Scroll output down	PREV	PgDn	'
Scroll output up	NEXT	PgUp	,
Select (bell)	SELECT	not assigned	&
Set tab under cursor	SET TAB	Alt-S	]
Softkeys 1 to 8	K1 to K8	User 1 Menu	1-8
Softkeys 9 to 16	K9 to K16	User 2 Menu	9,a-g
Softkeys 17 to 23,0	K17 to K23, K0	User 3 Menu	h-n,0
Step program	STEP	Alt-F1	S
Stop program	STOP	Shift-Alt-F4	!
System softkeys	SYSTEM	F10	}
Tab backwards	TAB BACK	Shift-Tab	(
Tab forward	TAB	Tab	)
Toggle	SUB mode	SUB MODE	Alt-T
User softkeys	USER	Shift-F10	{

Note: The Alt-F1...F8 keys are present on the System Menu and can be pressed without the Alt key when the System Menu is visible.

The following paragraphs present the keyboard function assignments by physical grouping. Detailed descriptions of each function are given at the end of this chapter.

Softkeys

KBD CMODE OFF. With KBD CMODE OFF, Function keys F1 to F8 are used as softkeys. As stated before, each softkey has four meanings, depending on which softkey menu is active when the key is pressed. The key assignments for each menu are given in the following table.

Keyboard Function	Generic Name	PC Keyboard
Step program	STEP	System-F1
Continue program	CONTINUE	System-F2
Run program	RUN	System-F3
Pause program	PAUSE	System-F4
Clear I/O	CLR I/O	System-F5
Alpha screen	ALPHA	System-F6
Graph screen	GRAPHICS	System-F7
Recall older line	RECALL	System-F8

Keyboard Function	Generic Name	PC Keyboard
Stop program	STOP	Shift-System-F4
Dump alpha screen	DUMP ALPHA	Shift-System-F6
Dump graphics screen	DUMP GRAPHICS	Shift-System-F7
Recall more recent line	RECALL NEW	Shift-System-F8
Softkeys 1 to 8	K1 to K8	User 1- F1 to F8
Softkeys 9 to 16	K9 to K16	User 2- F1 to F8
Softkeys 17 to 23	K17 to K23	User 3- F1 to F7
Softkey 0	K0	User 3-F8

Pushing INCR LABELS (Shift-F9) will cycle through the four menus. Pushing SYSTEM (F10) will immediately display the System Menu and USER (Shift-F10) will immediately display the User 1 Menu. Pushing the MENU (F9) key will toggle the menu on and off. These same operations can be done using the BASIC statements SYSTEM KEYS, USER *n* KEYS and KEY LABELS ON/OFF. A short-cut exists for the System Menu keys. They can be activated, even if a User Menu is displayed, by holding the Alt key down while the function key is pressed.

Keyboard Function	Generic Name	PC Keyboard
Menu labels on/off	MENU	F9
Increment menu labels	INCR LABELS	Shift-F9
System softkeys	SYSTEM	F10
User softkeys	USER	Shift-F10

KBD CMODE ON. With KBD CMODE ON, ten function keys are used as softkeys. PC function keys F1 to F10 correspond to keys k0 to k9 of the Nimitz keyboard. The labels on the screen are numbered to correspond to the number printed on the function keys. The number is not meant to be the softkey number, but is printed to help the user locate the correct key to press. For example, the first label is marked "1" and corresponds to the F1 key, though it is softkey K0.

Pushing the Shift key with a function key activates K10 to K19, though no labels are displayed for these keys. Pushing the Alt key with a function key activates System functions listed in the following table.

Keyboard Function	Generic Name	PC Keyboard
Softkeys 0 to 9	K0 to K9	F1 to F10
Softkeys 10 to 19	K10 to K19	Shift- F1 to F10
Softkeys 20 to 23	K20 to K23	not assigned

Keyboard Function	Generic Name	PC Keyboard
Step program	STEP	Alt-F1
Continue program	CONTINUE	Alt-F2
Run program	RUN	Alt-F3
Pause program	PAUSE	Alt-F4
Clear I/O	CLR I/O	Alt-F5
Alpha screen	ALPHA	Alt-F6
Graph screen	GRAPHICS	Alt-F7
Recall older line	RECALL	Alt-F8
Recall more recent line	RECALL NEW	Alt-F9
Stop program	STOP	Alt-F10

The Keypad

The PC keyboard has a keypad that has both numbers (for numeric keypad use) and edit functions: arrow keys, Home, PgUp, End, PgDn, Ins and Del. Some PC keyboards have separate numeric keypads and edit-function keys. The NumLock key is pressed to switch the keypad between numeric use and edit use. With NumLock off, the keypad is set for edit use. The keys produce a different function when used with the Shift key. The following table shows what keys are assigned to the keypad.

Keyboard Function	Generic Name	PC Keyboard
Clear Alpha Screen	CLR SCR	Home
Clear line	CLR LN	End
Scroll output up	NEXT	PgUp
Scroll output down	PREV	PgDn
Insert/replace	INS CHR	Ins
Delete under cursor	DEL CHR	Del
Pause program	PAUSE	Pause
Left	LEFT	←
Right	RIGHT	→
Next line	UP	
Previous line	DOWN	↓

Keyboard Function	Generic Name	PC Keyboard
End of output area	s-HOME	Shift-Home
Clear to end of line	CLR->END	Shift-End
Recall older line	RECALL	Shift-PgUp
Recall more recent line	RECALL NEW	Shift-PgDn
Insert program line	INS LN	Shift-Ins
Delete program line	DEL LN	Shift-Del
Begin of line	BOL	Shift-←
End of line	EOL	Shift-→
End of output/program	END	Shift-
Begin of output/program	BEGIN	Shift-↓
Next word	NEXT WORD	Alt-→
Previous word	PREV WORD	Alt-←

Note: The arrow keys and PgUp and PgDn do the opposite of what you might first expect. These keys are assigned to provide compatibility with HP BASIC's behavior. The arrow keys and PgUp and PgDn keys operate as follows: the computer screen (the output area) is a window onto a page of information (the extended output area) that is much larger than the computer screen. The position of your computer screen is fixed (on top of your computer), but the position of the page is movable. To move the page up, push the key or PgUp. To move the page down, push the ↓ key or PgDn. In other words, UP scrolls the screen up and DOWN scrolls the screen down. If you prefer to switch the action of the and ↓ keys to match other programs, use CONFIGURE KEY statements, explained in Chapter 8, "Customizing the Environment."

Functions invoked by the Alt Key

Besides the System Softkeys, which can be invoked with the Alt key, several other functions are invoked by holding down the Alt key while pressing one of the regular alphabetic keys on the keyboard. These are listed below. As you can see, they are mnemonic in nature:

Keyboard Function	Generic Name	PC Keyboard
Any character input	ANY CHAR	Alt-= or Alt-K
Dump alpha screen	DUMP ALPHA	Alt-A
Clear tab under cursor	CLR TAB	Alt-C
"EDIT" key macro	EDIT	Alt-E
Display functions	DISPLAY FCTNS	Alt-F
Dump graphics screen	DUMP GRAPHICS	Alt-G
Print all output	PRT ALL	Alt-P
Set tab under cursor	SET TAB	Alt-S
Result of last calculation	RESULT	Alt-R
Toggle SUB mode	SUB MODE	Alt-T

Other Function Keys

Keyboard Function	Generic Name	PC Keyboard
CAPS state toggle	CAPS LOCK	Caps Lock
Delete left of cursor	DEL LEFT	Backspace
Enter	ENTER	Enter
Reset BASIC	RESET	Ctrl-Break
Tab forward	TAB	Tab
Tab backwards	TAB BACK	Shift-Tab

Keyboard Function	Generic Name	PC Keyboard
Backspace	BACKSPACE	not assigned
Execute	EXECUTE	not assigned
Home position	HOME	not assigned
Katakana mode	KATAKANA	not assigned
Roman mode	ROMAN	not assigned
Select(bell)	SELECT	not assigned

Additional Keyboard Features

Windows gives the Keyboard some built-in functionality that is available to almost all programs that run under Windows. The Windows documentation is the best source of information in this area. The following two features, however, are worth noting here.

The first feature is the Print Screen key. While HTBasic is the active window, pressing Alt-Print Screen places a "snapshot" of the HTBasic window onto the clipboard. The image can then be pasted into any application that accepts the bitmap format. Pressing Print Screen (without Alt) places a snapshot of the entire screen onto the clipboard.

The second feature is the generation of any ASCII value using the numeric keypad. If Num Lock is on and you hold the Alt key down while typing a number on the keypad, the keyboard will automatically generate one keystroke corresponding to that value. For example, typing 1 3 on the keypad while holding down the Alt key will enter a CHR$(13) character. This feature, when combined with the HTBasic ANY CHAR (Alt-= or Alt-K) function allows you to generate almost any character to be included in a string literal, even if the character is not available on the keyboard.

The ASCII value of a character can be entered in either the Windows (ISO 8859, Latin 1) character set or the OEM (code page 437) character set. To use the Windows character set, type a leading 0. For example, Alt-(0163) enters the character "£" while Alt-(163) enters "ú". Character set tables are found in the *User's Guide.*

Note: A null character cannot be entered from the keyboard into a string literal. Use CHR$(0) instead.

Detailed Descriptions

The previous sections described the key assignments for each editor function. As stated, each function has a generic name. The following sections describe, by generic name, what each editor function does.

ALPHA

ALPHA makes the Alpha Screen visible. Pushing the key twice makes the Graphics Screen invisible. If ALPHA and GRAPHICS are merged, this key has no effect.

ANY CHAR

ANY CHAR allows any character to be entered. If the next character pressed is a Function key, then two characters are entered. The first character has the value CHR$(255). The second character is a code that identifies which editor function was pressed. You also can enter regular characters or control characters using this function.

BACKSPACE

BACKSPACE moves the cursor one place to the left. This is identical to the LEFT function. While this is the action of the Backspace key in HP BASIC, the Backspace key in HTBasic does a DEL LEFT.

BEGIN

BEGIN moves to the beginning of the program (if in edit mode) or the top most line in the Extended Output Area.

BOL

BOL moves the cursor to the beginning of the line.

CAPS LOCK

CAPS LOCK toggles the capital letter state of the keyboard. With CAPS LOCK on, uppercase letters will be produced when typing on the keyboard. When off, lowercase letters will be produced. On some computers the shift key temporarily inverts the state of the CAPS LOCK. Also, on some computers the state of the CAPS LOCK is not displayed on the screen.

CLR->END

CLR->END clears to the end of the line.

CLR I/O

CLR I/O aborts an I/O operation that is in progress. Unless timeouts have been enabled, the system will wait forever for an I/O operation to complete. Pressing this key forces the program to return to a paused condition. Pressing CONTINUE will cause the I/O statement to be reexecuted.

CLR LN

CLR LN clears the input line.

CLR SCR

CLR SCR clears the Alpha Screen. On systems where Alpha and Graph have been merged, this also will clear any graphics that are present. This key also is used to leave EDIT mode.

CLR TAB

CLR TAB clears a tab stop if one exists at the present cursor location.

CONTINUE

CONTINUE resumes program execution if the program is in a paused state.

DEL CHR

DEL CHR deletes the character where the cursor is.

DEL LEFT

DEL LEFT deletes the character to the left of the cursor. This is the default action of the Backspace key in HTBasic. This function is not available in HP BASIC.

DEL LN

DEL LN deletes the program line that the cursor is on, if in Edit Mode.

DISPLAY FCTNS

DISPLAY FCTNS displays characters that would normally be interpreted as control codes, including control characters that might normally be thrown away. This mode is useful for debugging I/O operations.

DOWN

DOWN scrolls the output area down one line. In edit mode, scrolling the program down one line leaves the cursor on the previous line.

DUMP ALPHA

DUMP ALPHA sends the contents of the Alpha Screen to the printer or file specified by the DUMP DEVICE IS statement.

DUMP GRAPHICS

DUMP GRAPHICS sends the contents of the Graphic Screen to the printer or file specified by the DUMP DEVICE IS statement. On bit mapped systems where Alpha and Graph have been merged, this will also DUMP the contents of the ALPHA screen.

EDIT

EDIT clears the input line and enters "EDIT" for you. You may then hit ENTER to start EDIT mode or you may enter a program line and hit ENTER to start EDIT mode at a particular line number. To exit EDIT mode, press the CLR SCR key. Likewise, you may press EDIT and then press a softkey to begin editing that softkey. To finish editing the key, press the ENTER key.

END

END moves to the end of the program (if in Edit mode) or the last line in the Extended Output Area.

ENTER

ENTER executes a command or enters a line into a program.

EOL

EOL moves the cursor to the end of the line.

EXECUTE

EXECUTE is the same as ENTER. This function exists for compatibility with old HP keyboards that had separate keys for executing lines and entering information.

GRAPHICS

GRAPHICS makes the Graphics Screen visible. Pressing the key twice makes the Alpha Screen invisible. If ALPHA and GRAPHICS are merged, this key has no effect.

HOME

HOME moves the print position to the home position on the screen. This is the upper-left corner of the screen.

s-HOME

s-HOME sets the print position to the first blank line at the end of the Extended Output Area. If this position is not visible, the screen is scrolled to move it onto the screen.

INCR LABELS

INCR LABELS cycles through the softkey menu labels at the bottom of the screen.

INS CHR

INS CHR toggles Insert/Replace mode. The default is replace mode, where new characters overwrite any existing characters at the position of the cursor. In insert mode, old characters are shifted to the right to open up a place for each new character typed.

INS LN

INS LN inserts a program line before the current line while in Edit mode. A new line number is generated automatically.

K0 to K23

K0 to K23 are user Softkeys 0 to 23. User softkeys can be used as either keyboard macros (sometimes called "typing aids") or for generating events in a running program. By default, a user softkey acts as a keyboard macro. When you push a user softkey, one or more keystrokes that have been assigned to that key will be entered just as if you had typed them. See "Softkey Macros" in Chapter 8, "Customizing the Environment."

If the key is used in an ON KEY statement, then it is no longer active for softkey macros, but instead generates an event that causes a GOTO, GOSUB, CALL or RECOVER.

KATAKANA

KATAKANA turns on Katakana Mode (as opposed to Roman Mode).

LEFT

LEFT moves the cursor one position to the left.

MENU

MENU alternately turns the softkey menu on or off.

NEXT

NEXT allows viewing of the next part of the Extended Output Area by scrolling the output area up.

NEXT WORD

NEXT WORD moves the cursor right to the first of the next word.

PAUSE

PAUSE pauses a running program. The program can be continued by pressing CONTINUE.

PREV

PREV allows viewing of the previous part of the Extended Output Area by scrolling the output area down.

PREV WORD

PREV WORD moves the cursor left to the first of the previous word.

PRT ALL

PRT ALL causes subsequent screen output to the message line, display line, input line and output area also to be sent to the PRINTALL device.

RECALL

RECALL recalls the last line entered. Several lines are saved and repeated pressing of this key recalls progressively older lines.

RECALL NEW

RECALL NEW recalls a more recent line. This is the opposite of the RECALL key. You can use RECALL and RECALL NEW to search back and forth through the saved lines.

RESET

RESET resets HTBasic. If a program is running, it is stopped. If a program is in memory, it is not deleted.

RESULT

RESULT recalls the result of the last numeric calculation into the input line.

RIGHT

RIGHT moves the cursor one position to the right.

ROMAN

ROMAN turns on Roman Mode (as opposed to Katakana Mode).

RUN

RUN starts a program running at the first line in the program.

SELECT

SELECT rings the bell.

SET TAB

SET TAB sets a tab stop at the present cursor position.

STEP

STEP executes one line of the program and then pauses. If a program is not currently running when you press this key, the first press of the key causes the program to be prerun in preparation for the next press of the STEP key. With each press of the STEP key, the statement that will execute next is displayed.

STOP

STOP stops a running program. If the program is executing in an I/O operation, you may have to press CLR I/O first.

SUB MODE

SUB MODE toggles between regular edit mode and SUB mode. In SUB mode, only the first line of each context is shown. This allows rapid movement to any of the contexts in the program. Pressing the key again returns to the regular edit mode where all lines of the program are shown.

SYSTEM

SYSTEM displays the System Softkey Menu.

TAB

TAB moves the cursor forward to the next tab stop.

TAB BACK

TAB BACK moves the cursor backward to the previous tab stop.

UP

UP scrolls the output area up one line. In Edit mode, scrolling the program up one line leaves the cursor on the next line.

USER

USER displays the User 1 Softkey menu.

Chapter 3
CRT and Graphic Drivers

HTBasic supports several graphic drivers. Graphic drivers are classified as CRT or Graphic Output drivers.

The CRT driver provides all the routines necessary for controlling output to the screen. This includes displaying text, writing screen labels, controlling the cursor, scrolling lines and clearing the screen. All CRT drivers also include a graphic output driver. The graphic output driver includes routines to move and draw, load and store portions of the screen, fill and edge an area, dither an area and control the color palette.

The following table lists the drivers available at the time of this manual printing.

Name	**Type**	**Graphic Driver**
INTERNAL	CRT	Reuse last display driver specified
WIN	CRT	Microsoft Windows Display Driver
HPGL	Graphic	Hewlett-Packard Graphic Language
PS	Graphic	PostScript printers, plotters and files

The PLOTTER IS Statement

The PLOTTER IS statement is used both to load drivers and switch among them. The PLOTTER IS statement directs vector graphics to a device or file. (Use the DUMP DEVICE IS statement to print bit-mapped graphics from the screen to a device or file.) The default PLOTTER IS device is the CRT. Executing a PLOTTER IS statement directs all subsequent graphics output to the specified target.

Loading Drivers

Under Windows and NT, driver files can be loaded at any point. It is recommended that PLOTTER IS statements be included in your AUTOST file to load any necessary drivers. Up to ten graphic and dump drivers can be loaded at a time.

To find the driver file, HTBasic takes the language specified in the PLOTTER IS statement and performs several operations upon it to find the correct driver file. ".DW6" is appended to the name. Then the following locations are searched, in the specified order:

1. The directory containing the HTBasic executable.
2. The current directory.
3. The Windows system directory (such as \WINDOWS\SYSTEM).
4. The Windows directory.
5. The directories listed in the PATH environment variable.

CRT Drivers

The syntax of the PLOTTER IS statement used to load CRT drivers is:

PLOTTER IS *destination*, *language* [; COLOR MAP]

where *destination* specifies the special interface select codes 1, 3 or 6. Also, the predefined constant CRT can be used for the value 1. *Language* is the driver name or the constant "INTERNAL". "INTERNAL" is a special language string synonymous with the last CRT driver specified. Optionally, the *language* string can include driver options as explained later. For example:

```
PLOTTER IS CRT,"INTERNAL";COLOR MAP
```

If the display has a writable color map, the COLOR MAP option enables color map manipulation. See the COLOR and SET PEN entries in the *Reference Manual* for an explanation of how to change the color map. If COLOR MAP is specified and the display type selected does not have a color map, an error is returned.

Default CRT Driver

HTBasic automatically loads the WIN driver when it starts. It is not necessary to use a PLOTTER IS "WIN" statement, nor a -CRT WIN command line switch.

CRTA and CRTB Modes

The WIN driver only supports CRTB mode. If you use CRTA mode on the DOS version or a series 200 BASIC workstation, this limitation may be confusing. Please read the material in Chapter 2, "Porting HP BASIC Programs to HTBasic," of the *HTBasic User's Guide.* Briefly, CRTA mode uses a true text mode to display the ALPHA screen. The CRTB mode uses bits written into a graphics screen to display the ALPHA screen.

The mode is specified using the interface select code in the PLOTTER IS statement. Specifying interface select code 3 selects CRTA mode and 6 selects CRTB mode. If the specified mode is not supported, the value is ignored. Specifying CRT or 1 in the PLOTTER IS statement reselects the last mode used.

Graphic Drivers

The syntax of the PLOTTER IS statement for graphic drivers is

PLOTTER IS *destination*, *language* [,*hard-clip*] [; APPEND]

Language is the driver name which optionally can be followed by driver options. Options are included by appending a semicolon to the driver name, followed by the options. Each driver has its own options. Consult the driver documentation, later in this chapter, for the legal options of each driver.

Hard-clip is composed of four values separated by commas that specify the size of the drawing surface. The four values are *xmin*, *xmax*, *ymin* and *ymax*, respectively. For example:

```
PLOTTER IS 10,"HPGL",2,268,0,190
PLOTTER IS "Pictfile","HPGL",5.75,250.50,7.25,136.875
```

The *destination* of a graphic driver can be a device or file, although not every driver can send output to all targets. For example, it doesn't make sense to send DXF output to anything but a file.

Devices

To send graphic output to a device such as a plotter or a printer capable of vector graphics, use the interface select code of the interface connecting the device. Use the device-selector if the device is on the IEEE-488 bus. If hard-clip limits are specified, they are given in the order *xmin*, *xmax*, *ymin*, *ymax* and are specified in millimeters. If the hard-clip limits are not specified, they are read from the device when this statement is executed. The specified device must respond to this query or the computer will wait indefinitely for the response. Use the CLR-I/O key to clear the computer if it gets stuck in this state.

The following example sends HPGL commands to a LaserJet III printer. The first line resets the printer, starts landscape printing and switches into HPGL mode. The second line directs plotter output to the LaserJet III and sets the hard-clip units for an 8-1/2 x 11 sheet of paper. Both lines assume that the LaserJet III is connected to interface select code 10.

```
OUTPUT 10;CHR$(27)&"E"&CHR$(27)&"&l1O"&CHR$(27)&"%1B";
PLOTTER IS 10,"HPGL",2,268,0,190
```

Files

To send graphics output to a file, *destination* should be replaced with the file name. The file must be an existing, ordinary or BDAT file. The hard-clip limits may be specified or defaulted to ⌠392.75 mm on the *x* axis and ⌠251.5 mm on the *y* axis. If APPEND is not specified, the file is positioned to the beginning and truncated. The file is closed when another PLOTTER IS, GINIT or SCRATCH A statement is executed. For example:

```
CREATE "DRAW.PLT",0
PLOTTER IS "DRAW.PLT","HPGL"
```

Driver Documentation

The following paragraphs document information specific to each particular driver.

WIN Driver

The WIN driver is a CRT driver that uses the Microsoft Windows display drivers.

For compatibility with HP BASIC/UX, options for the WIN driver are specified on the command line. Command line switches were explained in Chapter 1. These command line switches are passed to the WIN driver:

Switch	**Effect**
-colors	Number of Colors to Use
-cu	Specify how to use color maps
-fn	Use named font
-geometry	Specify initial size of HTBasic window
-gr	Graphics buffering
-title	Specify the window title

Window Resize

Resizing the HTBasic window using the mouse is supported, but has the following effects. If the number of text columns changes, any text present is discarded. If in edit mode, the screen is redrawn using the new size. If the HTBasic Window is made larger than the size of the extended output area buffer, the size change is ignored and the text screen will be justified in the upper-left corner of the HTBasic window. To prevent this from occurring, start HTBasic with a -geometry switch that specifies the largest window size desired or use the -buf switch to provide enough space in the extended output area buffer for the largest window size desired.

Any graphics present in the window are discarded. The current pen position is left undefined. The VIEWPORT, WINDOW and hard clip limits are unchanged, although GESCAPE CRT,3 returns the new window size. Use the GINIT statement to set the VIEWPORT, WINDOW and hard clip limits to the new window size. Or use the

```
PLOTTER IS CRT,"INTERNAL"
```

statement to activate use of the new hard clip limits without the side effects of GINIT.

Limitations

The DUMP statement is affected by graphics buffering (see the -buf switch, described in Chapter 1). When graphics buffering is off, parts of a window that are obscured or off the edge of the screen are not dumped correctly. If the window is minimized, a dump of the icon is returned. When graphics buffering is on, the window is correctly dumped in all cases.

Because Windows does not allow applications to use the entire color map, color combinations resulting from Normal and Alternate drawing modes are undefined. This same limitation, combined with Windows lack of support for plane oriented graphics, means SEPARATE ALPHA can not be implemented.

The standard windows VGA display driver does not allow applications to change the color map. If your computer has SVGA capabilities, change your Windows driver to a 256 color driver; this will allow color map operations.

HPGL Driver

The HPGL graphic output driver provides support for any output device that accepts Hewlett Packard's HPGL language. The driver also can store the HPGL information into a file that can be imported into a number of graphics packages and word processors.

The minimum and maximum hard clip limits can be specified for either a device or file. This allows you to output HPGL information to a printer that can't return P points. If no hard clip units are specified for a device, P points are requested from the device. If no hard clip units are specified for a file, the default hard clip limits are -392.75, 392.75, -251.5, 251.5 (millimeters).

The HPGL plotter driver is loaded with a line like

PLOTTER IS *device*,"HPGL[;*options*]",[*p1x,p2x,p1y,p2y*]

or

PLOTTER IS "*file*","HPGL[;*options*]",[*p1x,p2x,p1y,p2y*]

In the above, *device* refers to an HTBasic device number. *File* refers to a file in the computer's file system or a pipe. The file must already exist when the PLOTTER IS statement is executed.

Plotting Area

The points *(p1x,p1y)* and *(p2x,p2y)* determine the lower left and the upper right corners of a rectangular area the driver will plot to. These points are specified in mm from the lower left corner of the paper. *P2x* and *p2y* must be larger than *p1x* and *p1y*, respectively. All of these coordinates must be positive or zero if the PCL5 option is used (see Options, below). If the plotting area is omitted, the driver reads the plot area from the plotter, if it is connected to a serial or IEEE-488 port. If output is directed to a file or pipe, the driver uses the default values from the table below.

PCL5 Option	**Orientation**	**(P1x,P1y) (P2x,P2y)**
No	Landscape	(-393, -252) (393, 252)
No	Portrait	(-252, -393) (252, 393)
Yes	Landscape	(0,0) (254, 184)
Yes	Portrait	(0,0) (184, 254)

Polygons

The HPGL driver, for compatibility with HP BASIC, outputs polygon fills as separate lines. However, the driver can be instructed to output HPGL/2 polygon fill commands. This is useful if the plotter supports the polygon fill command or if an HPGL file is produced for import into another program that supports polygons. To enable polygon mode, use GESCAPE code 104, operation number 1:

```
10  INTEGER Param(1)
20  Param(0)=1    ! HPGL Operation Number 1 is HPGL/2 Flag
30  Param(1)=1    ! Set HPGL/2 Flag to 1=enable, 0=disable
40  GESCAPE Isc,104,Param(*)
```

If output is to a device, substitute the device ISC for Isc in line 40. If output is to a file, substitute 1 for Isc.

Options

The options are listed after the semicolon in the driver name, within the quotes. If more than one option is specified, the option names are separated by commas. When no options are specified, the HPGL driver produces the same output as the HPGL driver. The options are as follows:

COLOR. This option tells the driver that the device used for plotting is a color printer with plotter functions, such as the Hewlett-Packard PaintJet XL-300. This option is ignored unless the PCL5 option is also specified.

FILL. This option tells the driver that the plotter being used can do area filling. Area filling produced by the plotter is generally much faster than that produced by the driver.

GRAY. This option causes the driver to produce grayscale plots when used with a printer. Each color that normally would be plotted is changed to a brightness using the method explained in the Pen Colors section, below, before plotting. Note that the brightness level is inverted unless the INVERT option is also used. The GRAY option need not be specified; it is the default. This option is ignored unless the PCL5 option is also used.

HPGL2. By default, the driver produces plots for an HP-GL plotter. This option allows the driver to produce plots for an HP-GL/2 plotter, such as the Hewlett-Packard DraftMaster. Since HP-GL/2 plotters can all do area filling, the HPGL2 option turns on the FILL option.

INVERT. By default, the driver reverses black and white on color plots and reverses all gray levels on grayscale plots when the plots are made on a printer. This is suitable for printers that use dark inks on white paper, but is the opposite of the colors normally shown on the computer screen. The INVERT option causes colors or gray levels to be represented as they are on the computer screen. This option is ignored unless the PCL5 option is also used.

PCL5. This option tells the driver that the plotter is a laser or electrostatic printer with built-in plotter emulation using the PCL-5 language. This causes the driver to send escape sequences at the beginning and end of plots to enable and disable the plotter emulation. When this option is used, a PLOTTER IS CRT,"INTERNAL" statement should be executed at the end of plotting to make the printer eject the page containing the plot. Since all PCL-5 devices use the HP-GL/2 plotter language, this option turns on the HPGL2 and FILL options.

PORTRAIT. The PORTRAIT option causes the driver to produce plots in portrait orientation, that is, with the long edge of the paper vertical. Without this option, the driver produces plots in landscape orientation, with the long edge of the paper horizontal.

1600 The 1600 option provides compatibility for most newer HP DeskJet printers. This option is ignored unless the PCL5 and COLOR options are included.

Choosing Options. The following table may help in choosing from the FILL, HPGL2, PCL5 and COLOR options for many plotter models. Model numbers refer to Hewlett-Packard products except where noted.

Plotter Model	Options
7470A, 7475A, 7580A, 7585A, older 7580B and 7585B, 7090A, LaserJet II printer with Pacific Data Plotter-in-a-Cartridge, Sweet-Pea, Houston Instruments ImageMaker, other plotters that emulate the 7470A or 7475A, plot files for use in WordPerfect, Ventura Publisher, Microsoft Word, Pagemaker, Ami Pro, etc.	none
17440A, ColorPro, 7510A, 7550A, 7570A, newer 7580B, 7585B, 7586B, 7595A, 7596B, Houston Instruments DMP-60, JetPro	FILL
7550 Plus, DraftMaster series, CalComp Classic, JDL Model 4000E, Houston Instruments DMP-160, DMP-162R	HPGL2
LaserJet III printers, LaserJet IV printers, LaserJet IIP printer with LaserJet III emulation	PCL5

cartridge 7600 series electrostatic plotter, DesignJet and DesignJet 600 plotters	
PaintJet XL-300 printer, PaintJet XL printer with HP-GL/2 cartridge	PCL5,COLOR

Pen Colors

When the HPGL driver is used with a pen plotter, the HTBasic PEN command selects the indicated pen on the plotter. However, when the driver is used with a printer (as indicated by the PCL5 option), the effect of the PEN command is that described in the following text.

The colors or grayscales produced by each pen depend on the states of the COLOR and INVERT options used in loading the driver, as well as the state of the COLOR MAP option of the HTBasic CRT driver. If the COLOR MAP option is off, the following gray levels or colors are used:

PEN	**GRAY**	**COLOR**	**GRAY INVERT**	**COLOR INVERT**
0	white	white	black	black
1	black	black	white	white
2	30% black	red	70% black	red
3	89% black	yellow	21% black	yellow
4	59% black	green	41% black	green
5	70% black	cyan	30% black	cyan
6	11% black	blue	89% black	blue
7	40% black	violet	60% black	violet
8	black	black	white	white
9	30% black	red	70% black	red
10	89% black	yellow	21% black	yellow
11	59% black	green	41% black	green
12	70% black	cyan	30% black	cyan
13	11% black	blue	89% black	blue
14	40% black	violet	60% black	violet
15	black	black	white	white

If the COLOR MAP option of the CRT driver is on, the plot is made using the colors in the HTBasic color map if the COLOR option is used. If the INVERT option is not used, black and white are reversed. If the COLOR option is not used, the colors in the HTBasic color map are converted to shades of gray using the NTSC equation:

brightness = 11% blue + 59% green + 30% red

If the INVERT option is not used, the brightness is inverted before plotting is done. With both pen plotters and printers, the sign of the pen is ignored; the absolute value determines the pen used.

Drawing Mode

When the PCL5 option is specified, the HTBasic statement GESCAPE CRT,5 sets alternate drawing mode for the driver. Normally, the driver replaces anything previously at a location with what is currently drawn. In the alternate drawing mode, the previous black or colored areas show through the white areas of the new plot. The HTBasic statement GESCAPE CRT,4 returns the driver to normal drawing mode.

Line Thickness

If the PCL5 option is specified, line thicknesses can be set in the driver. Lines default to 0.35 mm thick. The line thickness for all pens can be changed by the GESCAPE CRT,104 statement as in either of the examples below:

```
INTEGER Param(1:2)        ! an array for the command
Param(1) = 10             ! line thickness code
Param(2) = thickness      ! desired thickness (in 1/100 GDU's)
GESCAPE CRT,104,Param(*) ! send thickness
```

```
INTEGER Param(1:2)        ! an array for the command
Param(1) = 11             ! line thickness code
Param(2) = thickness      ! desired thickness (in 1/100 mm)
GESCAPE CRT,104,Param(*) ! send thickness
```

Line Caps and Joins

When the PCL5 option is specified, line cap and join styles can be specified. By default, the device driver uses round caps to end lines and round joins to connect lines, which simulates the round pens used on pen plotters. This can be changed with the following statements.

```
INTEGER Param(1:3)        ! an array for the command
Param(1) = 12             ! line thickness code
Param(2) = cap            ! desired line cap
Param(3) = join           ! desired line join
GESCAPE CRT,104,Param(*) ! set cap and join
```

The values for *cap* and *join* can be selected from the following tables.

cap	**meaning**	**join**	**meaning**
1	butt cap	1	mitered join
2	square cap	2	mitered, beveled
if too long			
3	triangular cap	3	triangular join
4	round cap	4	round join
		5	beveled join
		6	no join

Note that many low-resolution PCL-5 devices use a butt cap and no join with lines less than 0.35 mm thick, regardless of the cap and join settings.

Crosshatching

The HPGL driver can crosshatch areas meant to be filled. This is its default behavior unless the FILL or PCL5 option is specified, in which case the default is to use solid fills.

If the FILL or PCL5 options are specified, the driver can be made to crosshatch filled areas with the following statements:

```
INTEGER Param(1:2)        ! an array for the command
Param(1) = 1              ! set fill type
Param(2) = state          ! turn solid filling on or off
GESCAPE CRT,104,Param(*) ! send command
```

State is 0 to use crosshatching and any other value to use solid filling. For compatibility with older drivers, if *state* is nonzero, this command turns on the FILL option if neither the FILL nor the PCL5 option was specified when the driver was loaded.

When crosshatching is turned on, the following sets of statements can be used to control the crosshatch parameters. If these statements are not executed, crosshatching is done with solid horizontal lines spaced 0.01 in. (0.25 mm) apart, which is useful on most devices for producing a solid fill.

```
INTEGER Param(1:2)        ! an array for the command
Param(1) = 2              ! set crosshatch type
Param(2) = type
GESCAPE CRT,104,Param(*) ! send command
```

Type is 1 for single hatching, 2 for crosshatching.

```
INTEGER Param(1:2)        ! an array for the command
Param(1) = 3              ! set hatch angle
Param(2) = angle          ! desired angle, degrees
GESCAPE CRT,104,Param(*) ! send command
```

Angle is the angle in degrees (regardless of the HTBasic RAD or DEG setting) for hatching. *Angle* is rounded to the nearest multiple of 45 degrees.

```
INTEGER Param(1:2)        ! an array for the command
Param(1) = 4              ! set line spacing
Param(2) = spacing        ! desired spacing (in 1/100 GDU's)
GESCAPE CRT,104,Param(*) ! send command

INTEGER Param(1:2)        ! an array for the command
Param(1) = 5              ! set line spacing
Param(2) = spacing        ! desired spacing (in 1/100 mm)
GESCAPE CRT,104,Param(*) ! send command
```

The above commands are equivalent except that in the first command, *spacing* is expressed in 1/100 GDU and in the second in 1/100 mm.

```
INTEGER Param(1:3)        ! an array for the command
Param(1) = 6              ! set line type for hatching
Param(2) = type           ! desired line type for crosshatching
Param(3) = size           ! desired pattern repetition size
GESCAPE CRT,104,Param(*) ! send command
```

Type is the type of line, as listed in the LINE TYPE section of the *HTBasic Reference Manual*. *Size* is the pattern repetition length in 1/100 GDU's. This would be 100 times the pattern repetition length specified in a LINE TYPE statement.

Pages

The GCLEAR statement causes subsequent plotting to be done on a new page. If the PCL5 option is specified, the GCLEAR statement causes the printer to eject the old plot. Also, opening a file with

```
PLOTTER IS "file","HPGL";APPEND
```

causes the driver to append new pages of plot information to the current file if it exists already. Note that most word processor programs and other programs that can import files will probably superimpose the plots imported from a file containing more than one plot.

Ending Plots

If the PCL5 option is used, the HPGL driver will not eject a plot until a GCLEAR statement is executed, HTBasic is ended or when the PLOTTER IS device is set to a different device. It is recommended that a statement like

```
PLOTTER IS CRT,"INTERNAL"
```

be placed at the end of each program section that produces a plot using the PCL5 option driver.

PostScript Driver

The PostScript graphics output driver generates PostScript-language files from HTBasic plotting commands. These files are suitable for printing on PostScript-language printers and photographic equipment and for importing into documents using the PostScript file format. The PostScript graphics output driver is loaded with the following statement:

PLOTTER IS *destination*,"PS[;*options*]",[*p1x,p2x,p1y,p2y*]

Destination refers to a device or file. If it is a file, the file must already exist when the PLOTTER IS statement is executed and it should be an *ordinary file*. Otherwise the HTBasic file header will appear as bad data at the start of the file.

The points *(p1x,p1y)* and *(p2x,p2y)* determine the lower left and the upper right corners of a rectangular area the driver will plot to. These points are specified in mm from the lower left corner of the paper. All of these coordinates must be positive or zero and *p2x* and *p2y* must be larger than *p1x* and *p1y*, respectively. If omitted, the driver uses *(p1x,p1y)* = (25.4 mm, 25.4 mm) and *(p2x,p2y)* = (262.7 mm, 190.5 mm) in landscape mode and *(p2x,p2y)* = (190.5 mm, 262.7 mm) in portrait mode, which produces a plot with adequate margins on US "A" or European A4 size paper. Note that most PostScript printers cannot print to the edges of the paper. Because of this, the points specified should include a small (about 1 cm) margin on each side when the driver is used with a printer.

Options

The options are listed after the semicolon in the driver name, within the quotes. If more than one option is specified, the option names are separated by commas. The options are as follows:

COLOR. This option causes the driver to produce color plots. Note that black and white are inverted from their values on the screen unless the INVERT option is also used. Color plots require a PostScript level 2 output device or a PostScript level 1 device with color language extensions.

GRAY. This option causes the driver to produce grayscale plots. Each color that normally would be plotted is changed to a brightness using the method explained in the Pen Colors section, below, before plotting. Note that the brightness level is inverted unless the INVERT option is also used. The GRAY option need not be specified; it is the default.

INVERT. By default, the driver reverses black and white on color plots and reverses all gray levels on grayscale plots. This is suitable for printers that use dark inks on white paper, but is the opposite of the colors normally shown on the computer screen. The INVERT option causes colors or gray levels to be represented as they are on the computer screen.

PORTRAIT. The PORTRAIT option causes the driver to produce plots in portrait orientation, that is, with the long edge of the paper vertical. Without this option, the driver produces plots in landscape orientation, with the long edge of the paper horizontal.

Pen Colors

The colors or grayscales produced by each pen depend on the states of the COLOR and INVERT options used in loading the driver, as well as the state of the COLOR MAP option of the HTBasic CRT driver. If the COLOR MAP option is off, the following gray levels or colors are used:

PEN	GRAY	COLOR	GRAY INVERT	COLOR INVERT
0	white	white	black	black
1	black	black	white	white
2	30% black	red	70% black	red
3	89% black	yellow	21% black	yellow
4	59% black	green	41% black	green
5	70% black	cyan	30% black	cyan
6	11% black	blue	89% black	blue
7	40% black	violet	60% black	violet
8	black	black	white	white
9	30% black	red	70% black	red
10	89% black	yellow	21% black	yellow
11	59% black	green	41% black	green
12	70% black	cyan	30% black	cyan
13	11% black	blue	89% black	blue
14	40% black	violet	60% black	violet
15	black	black	white	white

If the COLOR MAP option of the CRT driver is on, the plot is made using the colors in the HTBasic color map if the COLOR option is used. If the INVERT option is not used, black and white are reversed. If the COLOR option is not used, the colors in the HTBasic color map are converted to shades of gray using the HTSC equation:

brightness = 11% blue + 59% green + 30% red

If the INVERT option is not used, the brightness is inverted before plotting is done. GESCAPE codes 4 and 5 are ignored as is the sign of the PEN. Graphics always overwrite existing graphics.

Line Thickness

Lines default to 0.35 mm thick. The line thickness can be changed by the GESCAPE CRT,104 statement as in either of the examples below:

```
INTEGER Param(1:2)       ! an array for the command
Param(1) = 10            ! line thickness code
Param(2) = thickness     ! desired thickness (in 1/100 GDU's)
GESCAPE CRT,104,Param(*) ! send thickness
```

```
INTEGER Param(1:2)       ! an array for the command
Param(1) = 11            ! line thickness code
Param(2) = thickness     ! desired thickness (in 1/100 mm)
GESCAPE CRT,104,Param(*) ! send thickness
```

Line Caps and Joins

By default, the device driver uses round caps to end lines and round joins to end lines, which simulates the round pens used on pen plotters. This can be changed with the following statements.

```
INTEGER Param(1:3)        ! an array for the command
Param(1) = 12             ! set line cap and join
Param(2) = cap            ! desired line cap
Param(3) = join           ! desired line join
GESCAPE CRT,104,Param(*)  ! set cap and join
```

The values for *cap* and *join* can be selected from the following tables.

cap	**meaning**	**join**	**meaning**
1	butt cap	1,2	mitered join, beveled if too long
2	square cap	3,4	round join
3,4	round cap	5,6	beveled join

Crosshatching

By default, the PostScript plotter driver fills areas with shades of gray or color (if the COLOR option has been specified). The driver can be made to crosshatch filled areas with the following statements.

```
INTEGER Param(1:2)        ! an array for the command
Param(1) = 1              ! set fill type
Param(2) = state          ! turn solid filling on or off
GESCAPE CRT,104,Param(*) ! send command
```

State is 0 to use crosshatching and any other value to use solid filling.

When crosshatching is turned on, the following sets of statements can be used to control the crosshatch parameters. If these statements are not executed, crosshatching is done with solid horizontal lines spaced 0.01 in. (0.4 mm) apart.

```
INTEGER Param(1:2)        ! an array for the command
Param(1) = 2              ! set crosshatch type
Param(2) = type
GESCAPE CRT,104,Param(*) ! send command
```

Type is 1 for single hatching, 2 for crosshatching.

```
INTEGER Param(1:2)        ! an array for the command
Param(1) = 3              ! set crosshatch angle
Param(2) = angle          ! desired angle, degrees
GESCAPE CRT,104,Param(*) ! send command
```

Angle is the angle in degrees (regardless of the HTBasic RAD or DEG setting) for hatching. *Angle* is rounded to the nearest integer.

```
INTEGER Param(1:2)        ! an array for the command
Param(1) = 4              ! set line spacing
Param(2) = spacing        ! desired spacing (in 1/100 GDU's)
GESCAPE CRT,104,Param(*) ! send command

INTEGER Param(1:2)        ! an array for the command
Param(1) = 5              ! set line spacing
Param(2) = spacing        ! desired spacing (in 1/100 mm)
GESCAPE CRT,104,Param(*) ! send command
```

The above commands are equivalent except that in the first command, spacing is expressed in 1/100 GDU and in the second in 1/100 mm.

```
INTEGER Param(1:3)        ! an array for the command
Param(1) = 6              ! set line type for hatching
Param(2) = type           ! desired line type
Param(3) = size           ! desired pattern repetition size
GESCAPE CRT,104,Param(*) ! send command
```

Type is the type of line, as listed under the LINE TYPE topic in the *HTBasic Reference Manual*. *Size* is the pattern repetition length in 1/100 GDU's. This would be 100 times the pattern repetition length specified in a LINE TYPE command.

Pages

The GCLEAR statement causes subsequent plotting to be done on a new page. The driver inserts a PostScript "%%Page" comment at the beginning of each page. The comments are used by some print spooling software. Also, opening a file with

```
PLOTTER IS "file","PS";APPEND
```

causes the driver to append new pages of plot information to the current file if it exists already. Since the driver doesn't know how many pages are already in the file, it begins its "%%Page" comments with page 1. This may cause problems with some print spooling software.

Ending Plots

The PostScript language requires information at the end of a plot to cause the plot to be printed. This information is output when the GCLEAR statement is executed, HTBasic is exited or when the PLOTTER IS device is set to a different device. It is recommended that a statement like

```
PLOTTER IS CRT,"INTERNAL"
```

be placed at the end of each program section that produces a plot using the PostScript driver.

Chapter 4
Printer and Pixel Image Device Drivers

HTBasic supports several dump drivers. The CONFIGURE DUMP statement is used both to load drivers and switch among them. The following table lists the drivers available at the time of this manual printing.

Name	For these printers
WIN-DUMP	Send the dump to the default Windows printer
PCL	Advanced HP-PCL driver
PS-DUMP	PostScript printers, devices and files
GIF	Graphic Interchange Format files

The CONFIGURE DUMP Statement

The CONFIGURE DUMP statement specifies what graphic printer language or bitmapped file format to use for the DUMP statement. When the DUMP GRAPHICS statement is executed or the DUMP GRAPHICS Key is pressed, the contents of the screen are copied to a printer, device or file. If a file is specified, insure that it is an *ordinary file*, otherwise the HTBasic file header will appear as bad data at the start of the file. The syntax is

CONFIGURE DUMP TO *language*

where *language* is a string expression naming the printer language and driver options. For example,

```
CONFIGURE DUMP TO "PCL"
```

DUMP Device Statement

The DUMP DEVICE statement controls where the output of the DUMP is directed. There are several choices for this. When using the WIN-DUMP driver, is is necessary to direct the CONFIGURE DUMP to ISC 10, which is the default Windows printer. When using any other option, is it necessary to use the ISC of the device where the output is to be directed. For the parallel port, this is ISC 26. For a device on the IEEE-488 bus, this would be the device's address. For a file, this would be the file name.

Loading Drivers

It is recommended that CONFIGURE DUMP statements be included in your AUTOST file to load any necessary drivers. Up to ten graphic and dump drivers can be loaded.

The first time a driver is specified in a CONFIGURE DUMP statement, the driver is loaded and graphics are directed to it. When the driver is subsequently specified, it is not loaded again, but graphics are again directed to it.

As an example, if you wish to use an HP LaserJet II for screen dumps, use the following command to change to the HP printer control language:

```
CONFIGURE DUMP TO "PCL"
```

If a DUMP is made before doing a CONFIGURE DUMP, HTBasic automatically loads and uses the WIN-DUMP driver. (Under DOS, the EPSON driver is the default.)

Dumping Rows

To specify a portion of the screen to dump when DUMP GRAPHICS is executed, the GESCAPE PRT,106,param(*) statement is used. The param array must be a one dimensional INTEGER array of five elements. The first element is the operation number. The remaining elements specify the boundry for the DUMP. The boundry is specified in screen units:

param(1) - 1
param(2) - Beginning Row
param(3) - Ending Row
param(4) - Must be 0
param(5) - Must be 0

The full screen will be dumped if any of the following conditions occur:

1. The beginning row is greater than ending row.
2. A new Plotter, Graphics or Dump driver is loaded.
3. GINIT, SCRATCH A, PLOTTER IS, GRAPHICS INPUT,
CONFIGURE DUMP command or a Basic Reset is executed.

Number of Colors

The number of colors in the DUMP depends on both the dump driver and the CRT driver. All *dump* drivers support black and white dumps. Some dump drivers also can handle 16, 256 or 16777216 colors. The same is true of *display* drivers. The number of colors in the dump will be the largest number both drivers support.

Presently, on 24-bit displays, the DUMP GRAPHICS command approximates the colors on the screen using 2, 16, or 256 colors, depending on the resolution of the printer and the DUMP GRAPHICS driver. The colors in the printed image may be different from those on the screen.

Options

It is sometimes necessary to specify options to the drivers. Options are included by appending a semicolon to the driver name, followed by the options. Since options vary from driver to driver, the following driver sections contain more details on specific options.

WIN-DUMP Driver

The WIN-DUMP dump driver provides support for any printer supported by Windows that accepts bitmaps. The command to load the WIN-DUMP dump driver is:

CONFIGURE DUMP TO "WIN-DUMP[;*options*]"

If a DUMP is made before doing a CONFIGURE DUMP, HTBasic automatically loads and uses the WIN-DUMP driver.

Print Manager

The default interface select code (ISC) for DUMP DEVICE IS is 10, the Windows Print Manager interface. The WIN-DUMP driver can not send dumps to any other ISC. To bypass the Print Manager when DUMPing, use ISC26 in the DUMP DEVICE IS statement.

Because Windows is a multitasking environment in which several programs may try to print at once, Print Manager collects printer output into "jobs." Only when a job is done is it printed. Normally, the WIN-DUMP driver prints a single dump per print job. To mix text and screen dumps or multiple screen dumps on a single page, output some text to the page before doing the dump. For example,

```
10    ASSIGN @I TO 10
20    OUTPUT @I;"This is a  screen dump:"
30    OUTPUT @I
40    DUMP GRAPHICS
50    ASSIGN @I TO *
60    END
```

The various settings, such as margins and line height, made in the WIN-PRINT driver are honored by the WIN-DUMP driver. See the WIN-PRINT driver documentation in Chapter 6 for more information.

The EXPANDED keyword in the DUMP statement is ignored. The DUMP is made in landscape or portrait mode depending on the printer settings, as explained in Chapter 6.

DUMP Size

By default, the screen image is scaled until it fills 100% of the width between the left and right margins. The size can be changed using GESCAPE code 39. This example sets the scaling to 20% of the width between the margins:

```
10    INTEGER S(1:1)
20    S(1)=20
30    GESCAPE CRT,39,S(*)
40    END
```

INVERT Option

By default, the driver inverts all colors in the image. Black and white are exchanged as well as other colors. This is often suitable for output to a black and white printer, where printing is done with black ink on white paper, but may not be suitable for color output devices, where an exact image of the screen is wanted. The INVERT option causes the colors or gray levels to be dumped exactly as they are on the scren.

Graphics Buffering

The DUMP statement is affected by graphics buffering (see the -buf switch, described in Chapter 1). When graphics buffering is off, parts of a window that are obscured or off the edge of the screen are not dumped correctly. If the window is minimized, a dump of the icon is returned. When graphics buffering is on, the window is correctly dumped in all cases.

PCL Driver

The PCL dump driver provides support for devices and software that accept the Hewlett-Packard PCL printer language or HP Raster Interface Standard graphic commands. The driver supports both DUMP ALPHA and DUMP GRAPHICS from bitmapped displays. In the Windows Version of HTBasic, the DUMP ALPHA and DUMP GRAPHICS commands both dump any text or graphics visible in the window.

The PCL driver is loaded with a line like

CONFIGURE DUMP TO "PCL[;*options*]"

Options

The options are listed after the semicolon in the driver name, within the quotes. If more than one option is specified, the option names are separated by commas. When no options are given, output from the PCL driver is the same as the HP-PCL driver. The options are as follows:

ADJUST. The ADJUST option is ignored in the Windows version of HTBasic. All pixels are considered to be square and the dump is made using the aspect ratio of the window running HTBasic.

BW. This option causes the driver to dump using the paper color for the areas on the screen that were drawn using pen 0 and the ink color (usually black) for the areas on the screen drawn with any other pen. This is reversed if the INVERT option is also used. The BW option need not be specified explicitly; it is the default.

COLOR, CCMY, C16 and C256. These options cause the dump to be done in color to a color printer. The COLOR option uses the printer's default 8-color solid-color palette (black, white, red, green, blue, cyan, magenta and yellow), mapping each color on the screen to the closest one from the palette. COLOR uses the default RGB palette to dump the screen; CCMY uses the default CMY palette. The C16 and C256 options use a 16- or 256-color palette on the printer and only work with printers that have settable color palettes, such as the PaintJet series and the DeskJet 1200C. With printers that use dithering to print mixed colors, you may have to specify a coarser resolution than the printer is capable of in order to enable the dithering; for example, on the original PaintJet printer, C16 and DPI90 together are needed to produce dithering; C16 and DPI180 cause the printer to use only the 8 default colors when printing.

Printing using the COLOR and CCMY options swaps black and white colors when printing, unless the INVERT option is also used.

When using the solid-color palette with older PaintJet printers, the COLOR option should be used, as these printers do not support the CMY color model. The DeskJet 500C and 550C models can only generate color screen dumps with the CCMY option.

COMPRESS. The COMPRESS option specifies that the printer being used can do "packbits"-style data compression. If this option is specified, the screen dump is transmitted to the printer using fewer data bytes. The COMPRESS option can be used with all the LaserJet IIP and IIP+ printers, all LaserJet III and IV series printers, all DeskJet series printers, the PaintJet XL300 printer (but not the older PaintJets), and the DesignJet printers, as well as other brands of printers that emulate these. Note, however, that the printers with slower CPU's will print 2-4 times slower when printing compressed data, so COMPRESS may not be a good option to use with these printers.

EXPANDED. When the EXPANDED keyword is used on the DUMP DEVICE IS statement, graphics screen dumps are placed in landscape. When grayscale dumps are done, the PCL dump driver scales the dump to 10 x 7 inches, if EXPANDED, or 4 x 7 inches, if not EXPANDED. With color or black/white dumps, each screen pixel is dumped to a single printer pixel; use DPI150, DPI100, or DPI75 to increase the dump size in these cases.

DPI*nnn*. This option tells the driver to use *nnn* dots per inch when dumping graphics. Without this option, the printer's default resolution is used. This option is required for the GRAY option, explained below. The resolution specified must be one supported by the printer. For most newer devices, DPI75, DPI100, DPI150 and DPI300 are the legal values for this option. Some older printers, like the Hewlett-Packard ThinkJet, don't support this option.

With the COLOR and BW options, this option controls the size of the dump, by mapping each pixel on the screen to one of the specified-sized dots on the printer; with the GRAY option, this options controls the size of the sub-pixels used to create the printed image, as explained in the GRAY option section.

GRAY. The GRAY option causes the driver to consult the screen's color map and calculate a gray shade for each color using the NTSC grayscale equation. Screen dumps are produced using the resulting shades of gray. If the INVERT option is not also specified, white and black are reversed after the gray shade is calculated, so that lighter colors on the screen become darker colors on the printer.

When dumps are made using this option, the driver calculates the number of printer pixels, as specified in the DPI*nnn* option, required to print a single screen pixel to make a 9 x 6 3/4 inch (23 x 17 mm) plot, up to 4 x 4 printer pixels per screen pixel. The driver sets the appropriate number of printer pixels to black to represent the gray shade of the corresponding screen pixel.The GRAY option is ignored unless the DPI*nnn* option is also specified.

The NTSC grayscale equation is

brightness = 11% blue + 59% green + 30% red.

INVERT. By default, the driver makes images with black and white exchanged from the values used on the screen. If the GRAY option is used, the driver by default reverses the gray level of all pixels dumped from that seen on the display. This is often suitable for output to a printer, where printing is done with colored inks on white paper, but may not be suitable for film output devices, where an exact image of the screen is wanted. The INVERT option causes the colors or gray levels to be dumped exactly as they are on the screen.

RELATIVE. Normally, the driver begins each dump at the left margin. The RELATIVE option causes the driver to begin each dump at the printers current print position.

EJECT. Normally, the driver does not eject the page after a dump is finished. The EJECT option causes the driver to send a Form Feed character to the printer or file at the end of each dump.

APPEND

If the APPEND keyword is used with the DUMP DEVICE IS command and if the dump device is a file, the driver appends dumps to the file, separated by form feeds.

ALPHA Dumps

The DUMP ALPHA command has the exact same function as DUMP GRAPHICS because the ALPHA and GRAPHICS planes are combined in HTBasic for Windows.

DUMP ALPHA produces a text dump at the top of a US "A" or European A4- sized sheet of paper if done from a text screen (as selected by PLOTTER IS 3).

Option Tables

The following tables will help in choosing options to use with specific printer models. Two screen sizes are shown, a 640 x 480 pixel screen common on PC's and a 1024 x 768 pixel window size common on UNIX workstations and Windows. The proper DPI*nnn* option for other screen sizes can be determined by experimentation.

OPTIONS for 640x480 Screen

Printer Type	Dump Type	Portrait	Landscape
ThinkJet		too large	none
LaserJet	B/W	DPI100	none
LaserJet	grayscale	GRAY,DPI300	GRAY,DPI300
LaserJet II-IV	B/W	DPI100, COMPRESS	DPI75, COMPRESS
LaserJet II-IV	grayscale	GRAY,DPI300, COMPRESS	GRAY,DPI300, COMPRESS
DeskJet	B/W	DPI100 none	
DeskJet	grayscale	GRAY,DPI300	GRAY,DPI300
DeskJet 500/550C	color	CCMY,DPI100	CCMY,DPI75
DeskJet 560C	color	CCMY,DPI100	CCMY,DPI75
DeskJet 1200C	color	C256,DPI100	C256,DPI75
PaintJet & XL	B/W	DPI90	DPI90
PaintJet & XL	grayscale	GRAY,DPI180	GRAY,DPI180
PaintJet & XL	color	C16,DPI90 or COLOR,DPI90	C16,DPI90 or COLOR,DPI90
PaintJet XL300	B/W	DPI100	none
PaintJet XL300	grayscale	GRAY,DPI300	GRAY,DPI300
PaintJet XL300	color	C256,DPI100	C256,DPI75
DesignJet 600	B/W	DPI100	none
DesignJet 600	grayscale	GRAY,DPI300	GRAY,DPI300
DesignJet 600	color	C256,DPI100	C256,DPI75

OPTIONS for 1024x768 Screen

Printer Type	Dump Type	Portrait	Landscape
ThinkJet		too	large none
LaserJet	B/W	DPI150	DPI100
LaserJet	grayscale	GRAY,DPI300	GRAY,DPI300
LaserJet II-IV	B/W	DPI150, COMPRESS	DPI100, COMPRESS
LaserJet II-IV	grayscale	GRAY,DPI300, COMPRESS	GRAY,DPI300, COMPRESS
DeskJet	B/W	DPI150	DPI100
DeskJet	grayscale	GRAY,DPI300	GRAY,DPI300
DeskJet 500/550C	color	CCMY,DPI150	CCMY,DPI100
DeskJet 560C	color	CCMY,DPI150	CCMY,DPI100
DeskJet 1200C	color	C256,DPI150	C256,DPI100
PaintJet & XL	B/W	DPI180	DPI90
PaintJet & XL	grayscale	GRAY,DPI180	GRAY,DPI180
PaintJet & XL	color	COLOR,DPI180	COLOR,DPI180
PaintJet XL300	B/W	DPI150	DPI100
PaintJet XL300	grayscale	GRAY,DPI300	GRAY,DPI300
PaintJet XL300	color	C256,DPI150 or COLOR,DPI150	C256,DPI100
DesignJet 600	B/W	DPI150	DPI100
DesignJet 600	grayscale	GRAY,DPI300	GRAY,DPI300
DesignJet 600	color	C256,DPI150	C256,DPI100

PS-DUMP Driver

The PostScript dump driver provides support for devices and software that accept the PostScript graphics language. It provides support for both the DUMP ALPHA and DUMP GRAPHICS commands. In the Windows Version of HTBasic, the DUMP ALPHA and DUMP GRAPHICS commands both dump any text and graphics visible in the window.

The PostScript dump driver produces a screen image intended to be rendered on a US "A" size or European A4 size page. It scales the image so that its longest dimension fits in the shortest dimension of the paper with an adequate margin. When the EXPANDED keyword is used on the DUMP DEVICE IS statement, screen dumps change from their normal portrait orientation to landscape orientation.

The PostScript dump driver is loaded with the following statement:

CONFIGURE DUMP TO "PS-DUMP[;*options*]"

Options

The options are listed after the semicolon in the driver name, within the quotes. If more than one option is specified, the option names are separated by commas. The GRAY and COLOR options are ignored in ALPHA dumps. The options are as follows:

BW. This option causes the driver to dump using the paper color for the areas on the screen that were drawn using pen 0 and the ink color (usually black) for the areas on the screen drawn with any other pen. This is reversed if the INVERT option is also used. The BW option need not be specified explicitly; it is the default.

GRAY. This option causes the driver to render colors on the computer screen as shades of gray on the printer. Each shade of gray is calculated using the NTSC grayscale equation:

brightness = 11% blue + 59% green + 30% red.

Unless the INVERT option is used, the resulting brightness is inverted before printing, so that dark colors on the computer screen print as light colors and vice-versa.

COLOR. The COLOR option causes the driver to output a color image of the screen. The resulting PostScript screen image can only be rendered on a device that supports Level 2 PostScript or the color extensions of Level 1.

INVERT. By default, the driver makes images with black and white exchanged from the values used on the screen. If the GRAY option is used, the driver by default reverses the gray level of all pixels dumped from that seen on the display. This is often suitable for output to a printer, where printing is done with colored inks on white paper, but may not be suitable for film output devices, where an exact image of the screen is wanted. The INVERT option causes the colors or gray levels to be dumped exactly as they are on the screen.

ADJUST. The ADJUST option is ignored in the windowed versions of HTBasic. All pixels are considered to be square and the dump is made using the aspect ratio of the window running HTBasic.

The APPEND Keyword

If the APPEND keyword is used in the DUMP DEVICE IS statement, the dump driver appends all dump images after the first one to the existing file as new pages. The driver inserts "%%Page" comments, used by some print spooling software, into the file at the beginning of each page. If the dumps are done in separate HTBasic sessions, the driver doesn't know which page it is on, so it starts over with page 1. This may be a problem with some spooling software. Also note that only one page can be present in a file that will be imported into a word processor document.

ALPHA Dumps

The DUMP ALPHA command produces a black-and-white graphics image of the entire screen if done from a bitmapped screen (as selected by PLOTTER IS 6, the default on most types on displays) and MERGE ALPHA is in effect. DUMP ALPHA produces a black-and-white image of the text on the screen if done from a bitmapped screen and SEPARATE ALPHA is in effect. The image is in black and white even if there are three ALPHA pens (as with 256-color displays).

DUMP ALPHA produces a text dump at the top of a US "A" or European A4- sized sheet of paper if done from a text screen (as selected by PLOTTER IS 3).

GIF Driver

The GIF dump driver provides support for software that accepts CompuServe Graphics Interchange Format (GIF) files.

With this driver, the DUMP ALPHA command produces a black-and-white graphics image of the entire screen if done from a bitmapped screen (as selected by PLOTTER IS 6, the default on most types on displays) and MERGE ALPHA is in effect. DUMP ALPHA produces a black-and-white image of the text on the screen if done from a bitmapped screen and SEPARATE ALPHA is in effect. The image is in black and white even if there are three ALPHA pens (as with 256-color displays).

If done from a text screen (as selected by PLOTTER IS 3) DUMP ALPHA does an implicit GRAPHICS ON, changing to a bitmapped screen and then proceeds as outlined above.

When the EXPANDED keyword is used on the DUMP DEVICE IS statement, graphics screen dumps are rotated 90 degrees clockwise from their normal orientation.

The GIF dump driver is loaded with the following statement:

CONFIGURE DUMP TO "GIF[;*options*]"

Options

The options are listed after the semicolon in the driver name, within the quotes. If more than one option is specified, the option names are separated by commas. The options are as follows:

BW. The driver normally produces a 16- or 256-color screen dump when used with a color screen. The BW option causes the driver to produce a black-and-white screen dump with color screens. In this dump, pixels of color zero are dumped as black and pixels of any other color are dumped as white. (This is reversed if the INVERT option is also specified.)

COLOR. COLOR. This option causes the driver to produce a color dump. This option is the default; it need not be explicitly specified.

INVERT. The driver normally dumps an image in the colors shown on the screen. The INVERT option causes the driver to reverse black and white in the dump. All other colors are unchanged.

The APPEND Keyword

If the APPEND keyword is used in the DUMP DEVICE IS statement, the GIF dump driver appends all dump images after the first one to the existing file. Note, however, that the screen type and colormap are stored when the first image is dumped. If the screen type or colormap changes, the dump images after the first one will not be correct. Also note that most software that uses the GIF format cannot process multiple images in one file.

Chapter 5
Graphic Input Drivers

HTBasic supports loadable GRAPHIC INPUT drivers. The GRAPHIC INPUT driver is used by the DIGITIZE, READ LOCATOR and SET LOCATOR statements. The following table lists the drivers available at the time of this manual printing.

Name	**For These Devices**
KBD	Keyboard arrow keys or Mouse
KBDA	Keyboard arrow keys or Mouse (Absolute)
ARROW KEYS	Same as KBD
HPGL	HPGL Plotters or Digitizers
TABLET	Most available digitizing tablets

The GRAPHICS INPUT IS Statement

The GRAPHICS INPUT IS statement is used both to load drivers and switch among them. The GRAPHICS INPUT IS statement also specifies the interface connecting the device to the computer. The syntax for loading the driver is

GRAPHICS INPUT IS *device-selector*, "*driver-name* [;*options*]"

The *device-selector* specifies the device or interface to use to communicate with the graphic input device. This is usually KBD, an IEEE-488 device selector or the Serial interface select code. The *driver name* and *options*, shown in literal form above, can be specified with a string expression. *driver-name* is from the table above and *options* are described in the following descriptions. Here are some examples of GRAPHIC INPUT IS statements:

```
GRAPHICS INPUT IS KBD,"KBD"
GRAPHICS INPUT IS KBD,"KBDA"
GRAPHICS INPUT IS KBD,"ARROW KEYS"
GRAPHICS INPUT IS 705,"HPGL"
GRAPHICS INPUT IS 705,"TABLET;BIN-2,0,5000,0,5000"
```

Loading Drivers

The first time a driver is specified in a GRAPHICS INPUT IS statement, the driver is loaded and used for graphics input. When the driver is subsequently specified, it is not loaded again, but is again used for graphics input.

It is recommended that GRAPHICS INPUT statements be included in your AUTOST file to load any necessary drivers. Up to ten graphic and dump drivers can be loaded at a time. HTBasic automatically loads the "KBD" driver when it starts.

Under Windows, to find the driver file HTBasic takes the language specified in the GRAPHICS INPUT IS statement and performs several operations upon it to find the correct file. ".DW6" is appended to the name. Then the following locations are searched, in the specified order:

1. The directory containing the HTBasic executable.
2. The current directory.
3. The Windows system directory (such as \WINDOWS\SYSTEM).
4. The Windows directory.
5. The directories listed in the PATH environment variable.

KBD or ARROW KEYS Driver

The keyboard (KBD) graphics input driver provides support for input of X and Y coordinates from the mouse. The KBD driver is loaded at start up. The command to switch back to the KBD graphics input driver from another driver is

```
GRAPHICS INPUT IS KBD,"KBD"
 or
GRAPHICS INPUT IS KBD,"ARROW KEYS"
```

The following example program shows how to set up the KBD driver and get coordinate information from the input device.

```
10  PLOTTER IS CRT,"INTERNAL"
20  GRAPHICS INPUT IS KBD,"KBD"
30  TRACK CRT IS ON
40  FRAME
50  DIGITIZE X,Y,S$
60  PRINT X,Y,S$
70  END
```

KBDA Driver

The KBDA driver is identical to the KBD driver except that the mouse driver uses absolute coordinates instead of relative mouse movements when reporting the current position. The KBDA driver works better if you are tracking the mouse movements from a touch screen monitor or digitizer tablet that emulates a mouse.

HPGL Driver

The HPGL graphics input driver provides support for any input device that accepts Hewlett Packard's HPGL language. Some HPGL compatible devices are the HP 9111A and HPGL plotters. The following example assumes an HPGL capable device is attached to the IEEE-488 bus at primary address 5:

```
GRAPHICS INPUT IS 705,"HPGL"
```

TABLET Driver

The TABLET graphics input driver provides support for most digitizers currently available. It can use either the serial port or the IEEE-488 (GPIB or HP-IB) bus to communicate with the tablet. The following guidelines will help you in loading the driver and in selecting the proper tablet configuration and data communication options. The command to load the TABLET graphics input driver is as follows.

GRAPHICS INPUT IS Isc,"TABLET;[*mode*[,]][*resolution*]"

The *mode* option allows you to specify the method in which the tablet's data is interpreted by the driver. If both *mode* and *resolution* options are specified, specify the *mode* option first and separate the two by a comma. The following table gives the legal values for *mode*:

Mode	**Meaning**
(None)	Comma separated ASCII
BIN-1	Summagraphics MM Binary Format
BIN-2	Hitachi Binary Format
BIN-3	UIOF Binary Format.

If no *mode* is specified, the driver assumes the tablet is using a comma separated, CR/LF terminated, ASCII data format. The ASCII format and the different binary formats are discussed below.

The *resolution* option is of the form Xmin,Xmax,Ymin,Ymax. The *resolution* option is only necessary if the tablets range of X & Y values are different from the default values of 0-11000 in both the X & Y directions. The *resolution* option is discussed in greater detail below.

Examples:

```
GRAPHICS INPUT IS 9,"TABLET"
GRAPHICS INPUT IS 9,"TABLET;BIN-1"
GRAPHICS INPUT IS 705,"TABLET;0,5000,0,5000"
GRAPHICS INPUT IS 705,"TABLET;BIN-2,0,5000,0,5000"
```

Communication

The TABLET driver can use either the serial port or the IEEE-488 bus to communicate with the digitize tablet. This is specified by the interface-select-code in the GRAPHICS INPUT IS statement. For example:

```
GRAPHICS INPUT IS 702,"TABLET"  !IEEE-488 Address 2
GRAPHICS INPUT IS 9,"TABLET"    !First Serial Port
```

Communication with the tablet over the IEEE-488 bus is straight forward. You specify the device-selector (i.e. 702) and the control and data messages proceed without further setup.

Communication with the tablet over the serial port is more involved because of the many serial configuration options. The SERIAL driver can change the number of data bits, parity, stop bits and the baud rate. Make sure that the switches on the tablet are set to match the values used by the SERIAL driver or use CONTROL statements after loading the SERIAL driver to make the SERIAL driver use the same settings as the tablet.

With the SERIAL driver, the tablet may support either XON/XOFF handshaking or hardware handshaking. Find out which method your tablet supports and set the SERIAL driver to use the same handshaking. By default the SERIAL driver uses XON/XOFF handshaking, the following line is all that is needed to set the driver to this method.

```
10  LOAD BIN "SERIAL" !Loads SERIAL device driver
```

If you need to use hardware handshaking, you will have to set a number of other registers within the SERIAL driver. The following program lines specify hardware handshaking.

```
10  LOAD BIN "SERIAL" !Loads SERIAL device driver
20  CONTROL 9,5;0     !Use DTR and RTS
30  CONTROL 9,12;0    !Read DSR, CD and CTS
40  CONTROL 9,100;0   !Disable XON/XOFF handshaking
```

With some digitizers the RTS line must be held active to make the TABLET driver work correctly, otherwise an error will occur after several successful reads. To hold the RTS line active change program line 20 to CONTROL 9,5;2.

ASCII Data Format

The ASCII method of data transmission is preferred over binary; it is easier to set up and get working. The ASCII format can be used with either XON/XOFF handshaking or hardware handshaking. XON/XOFF handshaking is the preferred method. The ASCII data needs to be comma separated and CR/LF terminated. The ASCII data format needs to look something like the following line.

Sxxxx,Syyyy,F<CR><LF>

The meaning of these symbols is given in the following table:

Symbol	**Meaning**
S	sign flag
xxxx	X value
yyyy	Y value
F	button flag
<CR>	Carriage Return
<LF>	Line Feed

The sign flag doesn't need to be present and the number of X & Y digits doesn't matter either. The data cannot contain any decimal points within the string.

Binary Data Formats

Three binary data formats are supported: Summagraphics MM format, Hitachi format and UIOF format. The type of binary format is selected by specifying the BIN-1, BIN-2 and BIN-3 strings respectively. When using the binary format, XON/XOFF handshaking **cannot** be used, only hardware handshaking is allowed. The meaning of each bit in the binary formats is listed in the tables below:

BIN-1 - MM Binary Data Format

Byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1st	1	PX	T	Sx	Sy	Fc	Fb	Fa
2nd	0	x6	x5	x4	x3	x2	x1	x0
3rd	0	x13	x12	x11	x10	x9	x8	x7
4th	0	y6	y5	y4	y3	y2	y1	y0
5th	0	y13	y12	y11	y10	y9	y8	y7

BIN-2 - Hitachi Binary Data Format

Byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1st	1	PX	Fd	Fc	Fb	Fa	x15	x14
2nd	0	x13	x12	x11	x10	x9	x8	x7
3rd	0	x6	x5	x4	x3	x2	x1	x0
4th	0	0	0	0	0	0	y15	y14
5th	0	y13	y12	y11	y10	y9	y8	y7
6th	0	y6	y5	y4	y3	y2	y1	y0

BIN-3 - UIOF Binary Data Format

Byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1st	P	1	0	0	0	0	T	PX
2nd	P	0	0	Fe	Fd	Fc	Fb	Fa
3rd	P	0	x5	x4	x3	x2	x1	x0
4th	P	0	x11	x10	x9	x8	x7	x6
5th	P	0	0	Sx	x15	x14	x13	x12
6th	P	0	y5	y4	y3	y2	y1	y0
7th	P	0	y11	y10	y9	y8	y7	y6
8th	P	0	0	Sy	y15	y14	y13	y12

The meaning of each of these symbols is given in this table:

Symbol	**Meaning**
Sx & Sy	sign flag for X & Y coordinates
x15,...,x0	X coordinate, x0 is least significant bit
y15,...,y0	Y coordinate, y0 is least significant bit
Fe,...,Fa	button flag, Fa is least significant bit
PX	proximity bit
T	tablet identifier

Resolution

The TABLET driver assumes a default maximum resolution of 11000 units in both the X and Y directions. This value is used to scale the digitizer X & Y coordinates to the display WINDOW coordinates. If this value is not correct for your digitizer or if you want to adjust for any distortion, you can change the scaling values with the following command:

GRAPHICS INPUT IS 9,"TABLET;*Xmin,Xmax,Ymin,Ymax*"

Xmin and *Xmax* are the digitizer's X values that correspond to the display's minimum and maximum X values respectively. *Ymin* and *Ymax* are the digitizer's Y values that correspond to the display's minimum and maximum Y values respectively. Please note that these values are specified in device units.

The TABLET driver scales the digitizer X & Y coordinates into the display WINDOW coordinates. For example, suppose the screen's WINDOW resolution is 0-133 in the X direction and 0-100 in the Y direction and the digitizer's X & Y resolution is 0-11000. If the digitizer returned 11000,11000 as the current X & Y location, the DIGITIZE statement will return a value of 100,133 to the user. If you want the X & Y values to be the same for equal movements in the X & Y directions, specify a square WINDOW. For example:

```
WINDOW 0,100,0,100
```

Option Configuration

The digitizer has several options that are critical to make it work properly with HTBasic. They are:

• Handshaking Mode
• CTS Handshaking (if hardware handshaking is used)
• Absolute coordinates

If a binary format is specified, make sure that the tablet format and the TABLET driver format match. If the ASCII format is specified, then the data needs to be comma separated and CR/LF terminated.
XON/XOFF handshaking may only be used when the tablet is set to ASCII format. Hardware handshaking may be used for either format.

Some other tablet settings that are not critical, but recommended are as follows:

• Data transmitted only in proximity.
• Disable Increment mode.
• Disable leading zero's.
• Enable RUN mode.
• Enable Maximum report rate.

Please consult your digitizer documentation for the correct switch settings for these options.

Here are several SERIAL driver settings for the HP SketchPro digitizer in the HP Mode, Summagraphics Emulation Mode and Hitachi Emulation Mode. Please note that the HP SketchPro does not function correctly in the Summagraphics MM binary format. The HP tablet always had bit 7 either on or off depending on HTBasic's serial setup. This makes it impossible to read the phasing bit and synchronize with the tablet's data format. The following examples assume that the tablet data format is set to ASCII.

HP Mode

```
10  LOAD BIN "SERIAL"
20  CONTROL 9,5;0
30  CONTROL 9,12;0
40  CONTROL 9,100;0
50  GRAPHICS INPUT IS 9,"TABLET"
60  READ LOCATOR X,Y,S$
70  PRINT X,Y,S$
80  END
```

Summagraphics MM1103 Emulation

```
10  LOAD BIN "SERIAL"
20  CONTROL 9,5;2
30  CONTROL 9,12;0
40  CONTROL 9,100;0
45  CONTROL 9,4;2
50  GRAPHICS INPUT IS 9,"TABLET"
60  READ LOCATOR X,Y,S$
70  PRINT X,Y,S$
80  END
```

Hitachi HDG-1111B Emulation

```
10  LOAD BIN "SERIAL"
20  CONTROL 9,5;2
30  CONTROL 9,12;0
40  CONTROL 9,100;0
50  GRAPHICS INPUT IS 9,"TABLET"
60  READ LOCATOR X,Y,S$
70  PRINT X,Y,S$
80  END
```

You can bypass the tablet driver and display the byte sequence returned by the tablet by deleting line 10 and substituting the following program lines for line numbers 50, 60 and 70 in the above program examples.

```
50  ASSIGN @Dev to 9;FORMAT OFF
60  ENTER @Dev USING "#,B";A
70  PRINT IVAL$(A,16)
```

Chapter 6
I/O Device Drivers

HTBasic provides loadable drivers for support of different interfaces and I/O ports. I/O drivers for the CRT, KBD and printer are built into HTBasic. Other drivers should be specified in the AUTOST file. A total of 16 I/O drivers can be loaded.

This chapter describes how to use the "WIN-PRINT" driver, an interface to the Windows Print Manager (and windows printer drivers). This chapter also describes how to load the Serial and IEEE-488 I/O drivers bundled with HTBasic.

WIN-PRINT Driver

The WIN-PRINT device driver is built into HTBasic and allows access to standard windows printer drivers. If at least one windows printer driver is installed, interface select code (ISC) 10 is assigned to the WIN-PRINT device driver. This ISC can be changed as explained later.

The OUTPUT, CONTROL and STATUS statements are supported by this driver, ENTER, READIO, WRITEIO, ON INTR and ENABLE INTR are not.

Printing to the Printer

Use the OUTPUT statement to print to the printer. The ISC can be specified explicitly in the OUTPUT statement. But it is usually better to ASSIGN an I/O path to the ISC.

```
20  ASSIGN @Prn TO 10      ! Assign to an I/O Path
30  OUTPUT @Prn;"PI = ";PI ! Now use the I/O Path
40  ! Insert other printer statements here
50  ASSIGN @Prn TO *       ! End of print job, print it
60  END
```

Because Windows is a multitasking environment in which several programs may try to print at once, Print Manager collects printer output into "jobs." Only when a job is done is it printed. Normally, the WIN-PRINT driver ends the job when the I/O path is closed. If you specify the ISC explictly, you must also explicitly end the print job. The following conditions end a print job: closing the I/O path, executing the RESET statement, resetting the interface through control register 0, or writing a value to control register 111:

```
10  OUTPUT 10;"MAXREAL = ";MAXREAL
20  CONTROL 10,111;1
30  END
```

Printer Control

Regardless of the actual printer type, the WIN-PRINT driver responds to the control characters in the following table. All other characters will be printed (if possible).

Character	Function
CHR$(10)	Move the print location down one line.
CHR$(12)	Move the print location to the top of the next page.
CHR$(13)	Move the print location to column one.

The default margins are 1.27 cm (1/2 inch). Characters printed past the right margin are discarded. When the line advances past the bottom margin, the print location moves to the top of the next page.

Change the margins using CONTROL registers 104 to 107 as shown in the example below. Margins are specified in 1/100's mm. In other words, a value of 100 specifies 1 mm. To specify values in inches, take the desired value in inches and multiply by 2540. Some printers have minimum margins. Attempting to set the margins smaller than allowed by the printer results in an error.

```
10  CONTROL 10,104;1*2540   ! Left   margin: 1 inch
20  CONTROL 10,105;2*2540   ! Top    margin: 2 inches
30  CONTROL 10,106;1/2*2540 ! Right  margin: 1/2 inch
40  CONTROL 10,107;4000     ! Bottom margin: 4 cm
```

Selecting a Printer

Two options are available for printing in HTBasic for Windows. Either direct printing to the parallel port or routing through the Windows Print Manager.

Direct access to the parallel port is available to ISC 26. This is particulary helpful when sending escape codes to the printer that the print manager normally removes from the print command.

When you OUTPUT to ISC 10, HTBasic sends the data to the default printer. At least one printer must be installed. To install a Windows printer driver, please consult your Windows documentation. Briefly, in Windows 95, double-click on "My Computer," double-click on "Printers Folder," and double-click on "Add Printer." In Windows 3.1, click on Program Manager, open the Main program group and double-click on the Control Panel icon. Then double click on the Printers Icon. Then click on the Help button for further information.

Printing in HTBasic is divided into jobs. A job begins when the print spool file is opened and ends when the file, if not empty, is sent to the printer. If no print job is currently associated with Isc, a new print job on device Isc begins when:

- An OUTPUT Isc command is executed.
- The first point is plotted after PLOTTER IS Isc.
- A DUMP GRAPHICS command is executed and the current DUMP DEVICE is Isc.
- A CAT, LIST, or PRINT command is executed after PRINTER IS Isc.

If there is a print job associated with Isc, it ends when:

- CONTROL Isc,111;1 (FLUSH) command is executed.
- RESET Isc command is executed
- PLOTTER IS Isc command is executed after PLOTTER IS Isc.
- PRINTER IS Isc command is executed after PRINTER IS Isc.
- DUMP GRAPHICS command finishes when the current DUMP DEVICE is Isc.
- CAT or LIST command with PRINTER IS Isc terminates.

To send data to a different printer, change the default printer or use CONTROL register 102 to access the Printer Setup dialog. The following statement causes a Printer Setup dialog box to pop-up. The user can then select the desired printer. The user can also change the orientation of the print job, portrait or landscape. If the printer does not support the current margin settings, they are automatically adjusted.

```
CONTROL 10,102;1 ! Let user choose printer
```

WIN-PRINT CONTROL Registers

The following CONTROL registers are supported for ISC 10 and the Windows print manager. When using ISC 26 for direct routing to the parallel port only control registers 0,101,102,and 111 and status register 0 are available.

0 - Reset. The value must be non-zero.

101 - Change interface select code. For example, to change the interface select code from 10 to 12, you would use

```
CONTROL 10,101;12
```

To later change it back to 10, you would use

```
CONTROL 12,101;10
```

102 - For ISC 10; invoke Printer Setup dialog box. The value must be one. The action of other values is undefined. For ISC 26; this allows changing the assigned lpt port. The value must be the number of a valid printer port, either physical, or network assigned.

103 - Invoke Select Font dialog box. The value must be one. The action of other values is undefined.

104 - Set left margin. Units are 1/100's mm. Default is 1270.

105 - Set top margin. Units are 1/100's mm. Default is 1270.

106 - Set right margin. Units are 1/100's mm. Default is 1270.

107 - Set bottom margin. Units are 1/100's mm. Default is 1270.

108 - Set line spacing. Units are 1/100's mm.

109 - Set the current print position. Units are 1/100's mm. Register 109 sets the X and register 110 sets the Y values.

110 - Set the current print position. Units are 1/100's mm. Register 109 sets the X and register 110 sets the Y values.

111 - End the current print job and flush spooled output. The value must be 1.

WIN-PRINT STATUS Registers

The following STATUS registers are supported.

0 - Interface identification. ISC 10 returns a 302. ISC 26 returns 300.

104 - Get left margin. Units are 1/100's mm. Default is 1270.

105 - Get top margin. Units are 1/100's mm. Default is 1270.

106 - Get right margin. Units are 1/100's mm. Default is 1270.

107 - Get bottom margin. Units are 1/100's mm. Default is 1270.

108 - Get line spacing. Units are 1/100's mm.

109 - Get the current print position. Units are 1/100's mm. Register 109 gets the X and register 110 gets the Y values.

110 - Get the current print position. Units are 1/100's mm. Register 109 sets the X and register 110 sets the Y values.

111 - Retrieve the "page dirty" flag. A value of 0 means no output exists yet on this page. A value of 1 indicates output has been made to the current page, but the print job is not yet complete.

Serial (RS-232) Driver

Two serial interface drivers named "SERIAL" and "SERIAL32" are included with "HTBasic". These drivers support Windows compatible serial interfaces. They should work with third-party Windows communications drivers if they adhere to the Windows interface standard for serial drivers. The 16-bit driver is called "SERIAL" and should be used with Windows 3.x. The 32-bit driver is called "SERIAL32" and should be used on Windows 95 and Windows NT. Many devices can be connected to the serial ports and controlled with this driver.

If you are using a serial port only for a printer, you should use the "WIN-PRINT" interface rather than the "SERIAL" interface. "WIN-PRINT" was described earlier in this section. If a mouse or printer is not using the serial port and if the required Windows drivers are present for that port, HTBasic can drive that port with its serial port driver, described in this section.

Loading

The SERIAL drivers are loaded by the HTBasic commands

```
LOAD BIN "SERIAL[;options]"
```

or

```
LOAD BIN "SERIAL32[;options]"
```

The brackets above indicate items that may be omitted. This driver controls up to nine serial ports, named COM1 to COM9. After the serial driver is loaded, HTBasic takes control of the serial ports assigned to it. Until HTBasic terminates, no other process can use those serial ports.

Options

To specify options for the driver, place a semicolon after "SERIAL" and place the option names after the semicolon. If more than one option is specified, separate the names by spaces.

The options available with LOAD BIN "SERIAL" are as follows:

DRIVER *n.* Since each invocation of the SERIAL driver can control four interfaces, the DRIVER option is needed to specify to which of the two drivers the given options apply. The option is followed by a value of 1 to 4. The option may be abbreviated, as long as the abbreviation is unique. For example:

```
LOAD BIN "SERIAL;DR 1 options DR 2 more options"
```

Options apply to DRIVER 1 and *more options* apply to DRIVER 2. Notice that "DRIVER" is abbreviated as "DR". "D" would be too short since it would be unclear whether it meant DRIVER or DISABLE. In the above example, "DR 1" could be omitted and *options* would apply to DRIVER 1.

DEVICE *name.* By default, HTBasic uses COM1 and COM2 for the serial ports. This option causes the driver to use the specified port names instead. For example,

```
LOAD BIN "SERIAL;DEVICE COM7 DRIVER 2 DEVICE COM8"
```

DISABLE. By default, HTBasic takes control of two serial ports, as mentioned above. This option causes HTBasic to not take control of the specified port. DISABLE leaves COM1 free for other programs to use. DRIVER 2 DISABLE leaves COM2 free for other programs to use. Note that any serial port in use by a mouse, printer, or any other program must be DISABLED or the LOAD BIN "SERIAL" command will fail.

ISC *n.* This option assigns the ISC (interface select code) *n* to the serial port. If this option is omitted, the first serial port is assigned ISC 9 and the second ISC 11. If the first serial port has been disabled by DRIVER 1 DISABLE, the second serial port has ISC 11 and ISC 9 is undefined. *N* must be in the range 7 - 31.

If the first port is disabled, the first ISC option assigns the given ISC to the second serial port.

Modes of operation

Baud rates. When the LOAD BIN "SERIAL" command is executed, the default baud rate is undefined. The baud rate may be changed by the CONTROL *n*,3;*rate* command, where *n* is the ISC of the serial port and *rate* is the baud rate. The possible baud rates are dependent on the hardware.

Character Framing. When the LOAD BIN "SERIAL" command is executed, the default character framing is undefined. The character framing (bits per character and parity) is set using the CONTROL *n*,4;*value* command, where *n* is the ISC of the serial port and *value* is one of the values listed in Chapter 11 of the *HTBasic User's Guide* for control register 4. The possible character framing is dependent on the hardware.

Handshaking. When the LOAD BIN "SERIAL" command is executed, the serial port is set to use XON/XOFF handshaking and to ignore the CTS and RTS lines.

The use of the RTS and CTS lines can be enabled using the CONTROL *n*,5;*value* and CONTROL *n*,12;*value* commands. XON/XOFF software handshaking can be enabled or disabled using the CONTROL *n*,100;*value* command. These commands are explained in the *HTBasic User's Guide*, Chapter 11.

Interrupts

Interrupts are handled by the operating system and not by HTBasic. The ON INTR and ENABLE INTR commands are not supported for the serial ports and the STATUS registers dealing with interrupts and line states (registers 1, 2 and 8 - 12) all return 0.

READIO/WRITEIO

READIO and WRITEIO are supported and can be referenced in the serial.txt document included with the HTBasic software.

IEEE-488 Drivers

Many IEEE-488 boards are supported by HTBasic loadable device drivers. In fact, multiple IEEE-488 boards (even from different manufacturers) can often be used at the same time by loading the appropriate drivers. Five drivers are distributed with HTBasic: "GPIB", "GPIBN", "GPIBH", "HPIBS", and "GPIBTNT."

The following table lists the most common IEEE-488 boards, the driver to use ("GPIB", "GPIBN", "GPIBH","HPIBS" or "GPIBTNT") and the board type you must specify when loading the driver. If you are interested in a board that is not supported or not listed, you should first contact the board manufacturer and ask if the board is compatible with any of the boards that are supported (usually the National Instruments boards). If it is, proceed as if you have that type of board. If not, contact TransEra to see if a special driver is available for that board.

Compatibility Table

Company	Model	Driver	Board Type
Capital Equipment	CE-01000-00200	GPIBH	CAPITAL
Capital Equipment	CE-01000-00300	GPIBN	PC2
Hewlett-Packard	Most Models	GPIBH	HP
Hewlett-Packard	SICL Cards	HPIBS	-
IBM	GPIB	GPIBN	PC2A
INES	PCMCIA	GPIBN	INES
IOtech	GP488	GPIBN	PC2
IOtech	GP488A	GPIBN	PC2A
IOtech	GP488A	GPIBN	PC2A
IOtech	Personal 488 (GP488B)	GPIBN	PC2A
IOtech	Personal 488/2	GPIBN	MC-IOTECH
IOtech	Power488	GPIBN	PC2A
MetraByte	MBC-488	GPIBN	MBYTE
National Instruments	NAT488.2	GPIBTNT**	-
National Instruments	PC2	GPIBN*	PC2
National Instruments	PC2A	GPIBN*	PC2A
National Instruments	PC3 (old)†	GPIBN	PC3OLD
National Instruments	PC3 (new)	GPIBN*	PC3
National Instruments	AT-GPIB	GPIBN*	AT-GPIB
National Instruments	MC-GPIB	GPIBN*	MC-NI
Scientific Solutions	IEEE488	None	-
Scientific Solutions	IEEE488 LM	GPIBN	PC2
Tecmar	IEEE488	None	-
Tecmar	IEEE488 LM	GPIBN	PC2
TransEra	HM900	GPIB	TRANSERA
Ziatech	ZT-1444	GPIB	ZIATECH
Ziatech	ZT-1488A	GPIB	ZIATECH

*In mid-1990 the controller chip on some National Instruments boards changed from the 7210 to the NAT4882. If your board's controller chip is the 7210, you must use "GPIBN". If it is the NAT4882 chip then you may use either "GPIBN" or "GPIB"; we recommend using "GPIB" since it is 100% compatible with Series 200/300 HP-IB hardware.

**In Windows 32-bit OS, if your National Instruments card uses NI's 488.2 software, it is recommended to use the GPIBTNT driver.

†PC3 boards with revisions A, B and C are old. PC3 boards with revision C.1 or later are new.

The "GPIB" driver supports most I/O mapped IEEE-488 boards that use the TI 9914 IEEE-488 controller chip. The "GPIBH" driver supports most memory mapped IEEE-488 boards that use the TI 9914 IEEE-488 controller chip. The TI 9914 controller chip gives register compatibility with the HP 9000 Series 200/300 workstation HP-IB. The "GPIBN" driver supports most I/O mapped IEEE-488 boards that use the NEC PD7210 chip.

Although the NEC PD7210 chip is not register compatible with the HP 9000 Series 200/300 controllers it provides most of the functionality needed by HTBasic programs. Chapter 10, "IEEE-488 Interface Bus" in the *User's Guide* details the STATUS, CONTROL, READIO and WRITEIO register differences.

Loading

The driver is loaded by including a line like the following in your AUTOST file:

```
LOAD BIN "driver;options"
```

The value of *driver* is given in the compatibility table for the boards listed. If you purchase a TransEra driver separately, the documentation will tell you the driver name.

If an error is returned when the LOAD BIN is executed, the driver will not be loaded. Often, more specific error messages are available by pressing the PRT ALL key and then repeating the LOAD BIN statement.

Multiple GPIB Cards

Any combination of up to four GPIB interfaces may be active at one time. Each GPIB interface is associated with an Interface Select Code (ISC) by the LOAD BIN command. To use more than one GPIB interface, each interface must have a unique ISC. Therefore, a separate LOAD BIN command is required for each interface. However, you cannot use the same driver file name in more than one LOAD BIN command.

To use more than one GPIB interface that use the same driver, you must load copies of the driver with altered names. Here is an example which uses one GPIB card with ISC 7 and a second card with ISC 8. This example uses the GPIB driver.

1. Go to the HTBasic directory. For example:

```
c:> cd \HTBWIN
```

2. Make a copy of the GPIB driver. The copy can have any arbitrary base name, but must end with the .DW6 suffix. For example:

```
c:>copy GPIB.DW6 GPIB2.DW6
```

3. Run HTBasic for Windows.

4. Load one driver for ISC 7. For example:

```
LOAD BIN "GPIB;DEV PC2 ISC 7"
```

5. Load the driver for ISC 8. For example:

```
LOAD BIN "GPIB2;DEV PC2 ISC 8"
```

Options

The legal *options* for the IEEE-488 drivers are:

BOARD *board-type*
BASE *address*
DMA *n*
FAST
INTERRUPT *i*
ISC *n*
LPT *n*
NOTSYS
SLOT *n*
SYSTEM

One or more options can be specified, each separated by a space. The option may be abbreviated, as long as the abbreviation is unique.

BOARD Option

The BOARD option tells the driver the type of board you have. Legal types are given in the compatibility table, above. You do not need to specify the board type if you are using any of the driver and board combinations listed in this table:

Driver	Board Type
GPIB	TRANSERA
GPIBN	PC2
GPIBH	HP
GPIBP	NB488

Examples of the BOARD option are:

```
LOAD BIN "GPIBN;BO PC2A"     ! or
LOAD BIN "GPIB"
```

The first line will load the driver for a board of type PC2A. The second line will load the driver for a TransEra board.

BASE Option

The BASE option tells the driver the base address of your board. You must specify the address in hexadecimal, with no leading "&H". Consult the manufacturer's documentation for your board to find the current address setting.

Board	**Type Base (hex)**
AT-GPIB	2C0
CAPITAL	C000
HP	DC00
PC2	2B8
PC2A	2E1
PC3	280
PC3OLD	250
MBYTE	300
TRANSERA	2B8
ZIATECH	210

For example, if you have a PC2A board at address 42E1(hex), you would include the following line in your AUTOST file:

```
LOAD BIN "GPIBN;BO PC2A BA 42E1"
```

INTERRUPT Option

The INTERRUPT option tells the driver the interrupt number used by the board. Again, consult the manufacturer's documentation if you do not know the interrupt used by your board. If you do not specify an interrupt, then 5 is assumed.

To expand our example, if your PC2A board at 42E1(hex) is set to use interrupt 7, you would use the following LOAD BIN statement:

```
LOAD BIN "GPIBN;BO PC2A BA 42E1 IN 7"
```

ISC Option

The ISC option is used to specify the interface select code that HTBasic will use when referring to the board. Normally, you do not specify an ISC and it is automatically set to 7 to match an HP BASIC workstation. But if you are using multiple IEEE-488 boards in your PC, you must use the ISC option when loading additional drivers so that each has a unique ISC.

For example, suppose in addition to our example PC2A board at 42E1(hex) we also have a TransEra board at 2B8(hex) using interrupt 5. The following two lines would load drivers for the two boards:

```
LOAD BIN "GPIBN;BO PC2A BA 42E1 IN 7"
LOAD BIN "GPIB;ISC 8"
```

The PC2A would have an ISC of 7 and the TransEra board would have an ISC of 8. Note that since the board type, base address and interrupt numbers of the TransEra board matched the "GPIB" driver defaults, they didn't need to be specified.

DMA Option

The DMA option is used to specify the DMA channel to use for TRANSFER. Specify the DMA channel, a number from 1 to 7, in decimal. A value of 0 causes TRANSFER to use interrupts instead of DMA and is the default. For example:

```
LOAD BIN "GPIB;DMA 3"
```

Since no other switches are specified, this example defaults to a TransEra board, base address 2B8, interrupt 5 and ISC 7.

SYSTEM and NOTSYS Options

The NOTSYS option is used if another computer on the bus is system controller. The option stands for "NOT SYStem controller" and causes the board to act as a Talker/Listener Device that can become active controller if control is passed to it.

The SYSTEM option is used to make the board act as system controller. These options are useful for overriding the hardware options on the board. HTBasic assigns a primary address of 21 to the board if it is system controller and 20 if not. CONTROL register 3 can be used to change the primary address.

Remember: Only one system controller can be connected to the bus.

Micro Channel Slot Number

For the National Instruments MC-GPIB and the IOtech Personal 488/2 boards, if you have more than one of these boards installed, you must use the SLOT option. Normally, HTBasic scans for micro channel boards starting from the first slot. If multiple boards are installed, the first board is always found. For boards after the first, specify the slot number to begin scanning. The number should be in the range 1 to 8. For example, if three boards are installed in slots 1, 3 and 7, the following example will correctly find all three boards:

```
10  LOAD BIN "GPIBN;BO MC-NI"
20  LOAD BIN "GPIBN;BO MC-NI SLOT 3"
30  LOAD BIN "GPIBN;BO MC-NI SLOT 7"
```

LPT Option

The LPT option is only used by the GPIBP driver. It specifies the printer port that the Personal 488/NB interface is attached to. Numbers should be in the range 1 to 4. If you do not specify a printer port, then LPT1 is assumed. The following example sets the driver to use LPT3:

```
LOAD BIN "GPIBP;LPT 3"
```

FAST Option

The FAST option is only used by the GPIBP driver. It instructs the driver to use the faster 8-bit transfer when accessing the Personal 488/NB interface through the printer port. Some printer ports are not capable of using the 8-bit transfer reliably, so the slower 4-bit transfer is selected by default. If you run the program NB488.EXE supplied with the IOtech Personal 488/NB interface, it will check to see if your printer port can handle the faster 8-bit transfers. The following example sets the driver to use the faster 8-bit transfer.

```
LOAD BIN "GPIBP;FAST"
```

TransEra's HM900 Board

This section explains how to set up the I/O address, interrupt number and DMA channel number for the TransEra HM900 Board. Earlier parts of this chapter explained how to load the device driver required by HTBasic.

The TransEra IEEE-488 bus controller board includes the National Instruments 9914 IEEE-488 controller chip to provide complete hardware compatibility with an HP workstation's HP-IB hardware.

The board is shipped with the jumpers and options already set for correct operation with HTBasic. If you have another board in your computer that uses interrupt 5 or I/O addresses in the range &H2A0 to &H2BF, you will need to change either the TransEra board or the other board so that the two no longer conflict. The following paragraphs explain how to change the default address, interrupt number or disable the System Controller status of the board.

I/O Address

The default I/O address for the HM900 board is &H02B8. Addresses in the range &H02A0 to &H02BF are used. If these addresses conflict with other hardware installed in your computer, the address can be changed. It can be set to any address between &H0018 - &H03F8, in increments of &H20. The board uses 24 I/O addresses below and 8 above the specified address. This range is listed as the Address Range in the table below:

TransEra IEEE-488 Card I/O Address Table

Chip			Switch Number					Address
Address	7	6	5	4	3	2	1	Range
018	on	on	on	on	on	on	on	000-01F
038	on	on	on	on	on	on	off	020-03F
058	on	on	on	on	on	off	on	040-05F
078	on	on	on	on	on	off	off	060-07F
098	on	on	on	on	off	on	on	080-09F
0B8	on	on	on	on	off	on	off	0A0-0BF
0D8	on	on	on	on	off	off	on	0C0-0DF
70F8	on	on	on	on	off	off	off	0E0-0FF
118	on	on	on	off	on	on	on	100-11F
138	on	on	on	off	on	on	off	120-13F
158	on	on	on	off	on	off	on	140-15F
178	on	on	on	off	on	off	off	160-17F
198	on	on	on	off	off	on	on	180-19F
1B8	on	on	on	off	off	on	off	1A0-1BF
1D8	on	on	on	off	off	off	on	1C0-1DF
1F8	on	on	on	off	off	off	off	1E0-1FF
218	on	on	off	on	on	on	on	200-21F
238	on	on	off	on	on	on	off	220-23F
258	on	on	off	on	on	off	on	240-25F
278	on	on	off	on	on	off	off	260-27F
298	on	on	off	on	off	on	on	280-29F
2B8	on	on	off	on	off	on	off	2A0-2BF
2D8	on	on	off	on	off	off	on	2C0-2DF
2F8	on	on	off	on	off	off	off	2E0-2FF
318	on	on	off	off	on	on	on	300-31F
338	on	on	off	off	on	on	off	320-33F
358	on	on	off	off	on	off	on	340-35F
378	on	on	off	off	on	off	off	360-37F
398	on	on	off	off	off	on	on	380-39F
3B8	on	on	off	off	off	on	off	3A0-3BF
3D8	on	on	off	off	off	off	on	3C0-3DF
3F8	on	on	off	off	off	off	off	3E0-3FF

If you change the I/O address, you must inform the software that uses the board. If you are using HTBasic, specify the new address in the LOAD BIN statement:

```
LOAD BIN "GPIB;BASE 3B8"
```

This statement corresponds to an I/O address of &H3B8 and the options should be set as follows:

Chip			Switch Number					Address
Address	7	6	5	4	3	2	1	Range
3B8	on	on	off	off	off	on	off	3A0-3BF

Interrupt and DMA Jumpers

The jumpers at J1 specify the I/O interrupt number used by the board, and should be set to 5. You may use another interrupt if it is not being used by another device. Because interrupt 2 is used as a bridge to the upper range of interrupts, it is not recommended that this interrupt be used with an IEEE interface. To use interrupt 7 place the jumper on I7 and use the following LOAD BIN statement in your AUTOST file:

```
LOAD BIN "GPIB;INT 7"
```

Note: The TransEra HM900 cards shipped prior to February 1998 were set to interrupt 2 by default.

DMA Channel

The jumpers at J2 specify the DMA request and acknowledge channels used by the board. If DMA is not used by the software driver, no jumpers should be placed on J2. This is the default. If DMA is used, the same channel should be used for both. Pick a channel that is not being used by another device. Generally, channels 1 and 3 are the only available channels. Put on the jumpers and inform the software driver. If you are using HTBasic, specify the DMA channel in the LOAD BIN statement with the DMA option. For example, to use DMA channel 1, place the jumpers on DR1 and DA1 and include the following option in your LOAD BIN statement:

```
LOAD BIN "GPIB;DMA 1"
```

System Controller

Switch 8 is used to enable or disable the System Controller capabilities of the board. If switch 8 is on, the board will act as system controller. When using the board with HTBasic, the switch setting specifies the default state, which is overridden with either the -SYSTEM or -NOTSYS driver options.

Only one system controller can be attached to the bus at a time. If you have another computer on the bus that will be the system controller, you may disable the system controller capabilities of the HM900 board by setting switch 8 off. The board then defaults to a Talker/Listener device. If control is passed to the board, it can become the active controller.

HP SICL Driver -HPIBS

The HPIBS driver supports Hewlett-Packard interfaces 82340A, 82341A,and 82341B. Before this driver can be used, the HP I/O Libraries (SICL) software must be installed, and the interface must be configured correctly for SICL.

If you are using an HP 82335 interface, choose the GPIBH driver. This interface is not supported by any other driver.

If you are using an HP 8234X interface, you can choose from two drivers: GPIBH or HPIBS.

Choosing a GPIB Interface Driver

- The GPIBH driver provides the traditional level of support for GPIB interface cards, but does not access SICL features.
- The HPIBS driver uses the HP SICL library, but does not have the low-level hardware access of the GPIBH driver.

Some guidelines for making your choice follow:

Features of the HPIBS Driver

Choose the HPIBS driver if any of these features are more important to you than the features listed for the GPIBH driver.

• Maximum performance from the 8234X interfaces. For example, the HPIBS driver can achieve large-block ENTER rates of 470 kbytes per second with the 82341B card. The GPIBH driver has a rate of about 40 kbytes per second in the same test.

• Interoperability with SICL locks. If you are running another SICL application with HTBasic for Windows, the HPIBS driver allows the two applications to cooperate by using interface locks.

The GPIBH driver does not support SICL locks and does not provide Control Register 255.

• HP 82341A support. The SICL library supports the now-obsolete HP 82341A interface. The GPIBH driver supports its replacement, the HP 82434lB, but does not support the original 'A" version.

Features of the GPIBH Driver

Choose the GPIBH driver if READIO and WRITEIO support are important to you. The GPIBH driver supports READIO and WRITEIO. Drivers based on the SICL library do not support READIO and WRITEIO.

The following options are recognized by the HPIBS driver:

HPIBS Driver Options

DEVICE Option. The general technique of using options with the LOAD BIN command is explained earlier in this chapter. Several GPIB hardware options, such as INTERRUPT and NOT SYS, are not used with the HPIBS driver.

These characteristics are controlled by the I/O Config utility and cannot be changed by options to LOAD BIN.

This option specifies the SICL symbolic name of the interface to be controlled. The symbolic name of each interface is set by the SICL I/OConfig utility. If you do not specify a DEVICE name, the default name "hpib7" is used. An example of specifying a different symbolic name is:

LOAD BIN "HPIBS;DEV ieee7"

ISC Option. The ISC option specifies the Interface Select Code that BASIC will use when communicating with the interface. If you do not specify an Isc, the default is 7. The following example specifies an Interface Select Code of 8:

LOAD BIN "HPIBS;Isc 8"

TIMEOUT Option. The TIMEOUT option provides a way to recover from "hung" keyboard commands, in addition to the CLR-I/O key. Note that this timeout for keyboard commands is independent from the ON TIMEOUT action specified in your program. The timeout value is specified in seconds, and a value of 0 deactivates keyboard timeouts. If you do not specify a TIMEOUT option, the default is 0 (never timeout). The following example sets the timeout for keyboard commands to 12 seconds:

LOAD BIN "HPIBS;TIME 12"

Interface Registers

CONTROL Registers. The IEEE-488 chapter of the *User's Guide* documents the STATUS and CONTROL registers normally available for GPIB interfaces. The HPIBS driver supports all of these registers, with the following exceptions:

STATUS Registers. The HPIBS driver does not support user control of NDAC holdoff. CONTROL register 4 is not implemented and will give Error 55.

The following ENABLE INTR events are not supported by the HPIBS driver. Status register 5 always reports them as 0:

Bit	Value	Meaning
6	64	Handshake Error
5	32	Unrecognized universal command
4	16	Secondary command while addressed
2	4	Unrecognized addressed command

All other HP-IB interrupt events are supported and reported properly in STATUS registers 4 and 5.

The "LSB of last address" bit (bit 8) in STATUS register 6 is not implemented by the HPIBS driver and is always reported as 0.

STATUS register 7 has a high byte that reports bus control lines and a low byte that reports bus data lines. The HPIBS driver does not support the direct reading of bus data lines The upper byte of this register is supported, but the lower byte is always 0.

CONTROL and STATUS Register 255. In addition to the STATUS and CONTROL registers described in the User's Guide, the HPIBS driver provides register 255. This register is similar to register 255 of HP BASIC/UX drivers. Unused bits are ignored by the CONTROL statement and return 0 to the STATUS statement. The register map follows.

Bit	Value	Meaning
7	128	Not Used
6	64	Not Used
5	32	Reserved for future use
4	16	Reserved for future use
3	8	Reserved for future use
2	4	ENTER buffering
1	2	Not Used
0	1	Interface Lock

Some examples using register 255 follow:

```
CONTROL 7,255;4   !Enables buffering for ENTER

CONTROL 7,255;0   !Disables buffering for ENTER

CONTROL 7,255;4+1 !Lock GPIB, enable buffering

STATUS7,255;Stat  !Get status of register 255
```

Interface locking can be used to help HTBasic cooperate with other Windows applications that might also access SICL-based interfaces. Setting bit 0 locks the interface, while clearing bit 0 unlocks the interface.

ENTER buffering can increase the speed of free-field ENTER statements. A free-field ENTER is one without a USING clause, such as:

```
ENTER @Dev;A$
```

Unsupported Keywords

READIO and WRITE/IO of interface registers are not supported by the HPIBS driver. Attempts to use READIO or WRITE/IO cause Error 170.

The TRANSFER statement is not supported by the HPIBS driver. Attempts to use TRANSFER cause Error 471.

PPOLL Note. Parallel Poll configuration is automatic with the HPIBS driver. The Active Controller can configure a non-controller PC GPIB interface for parallel poll, and the PC interface will respond correctly based upon that configuration.

If bit 14 of the Interrupt Enable register is set, HTBasic for Windows will receive an interrupt indicating that a parallel poll configuration has occurred. There is no way, however, to know the value of the configuration byte that came from the Active Controller.

You can set the parallel poll response by using CONTROL register 2 or 5, or you can automatically accept the configuration sent by an Active Controller.

NI's API Driver - GPIBTNT

This Windows driver calls the National Instruments Software that is shipped with the NI GPIB card. For details on installation of this software, see your Getting Started Guide for your GPIB card. To use this driver, you must first install the National Instruments software. With this installed, load the driver from inside of BASIC with the following statement:

LOAD BIN "GPIBTNT;DEV device_name"

where device_name is the name of the device assigned to the GPIB card in the NI GPIB control panel.

PCMCIA Cards

The following information covers using PCMCIA cards with both versions of HTBasic:

National Instruments Cards

To access a PCMCIA GPIB interface from National Instruments, you can use either of the following LOAD BIN lines:

```
LOAD BIN "GPIB;BO PCMCIA"
```

```
LOAD BIN "GPIBN;BO PCMCIA"
```

The HTBWIN drivers will work without loading any NI driver software. The "GPIB" driver is recommended if you want compatibility with HP BASIC workstations, since it causes the GPIB chip to switch to the TI9914 mode.

INES cards

To access a PCMCIA GPIB interface from INES, type the following command before you start Windows:

```
c:>IEPCMCIA -IO 0 -INT 0
```

This registers the card. The BASIC driver can be loaded using the following LOAD BIN line:

```
LOAD BIN "GPIBN;BO INES"
```

If you encounter problems with these LOAD BIN commands, you may need to use a different brand of card services. Please contact Technical Support for details if you have problems.

If problems persist with the "BO INES" or "BO PCMCIA" options, there is an alternative. Follow the instructions provided with the interface to get the PCMCIA card recognized and registered, then determine the base address and interrupt line. The National Instruments utility "gpibinfo.exe" provides this information. You may also be able to get the information from the registry. A typical menu path is:

START >SETTINGS >CONTROL PANEL >SYSTEM >DEVICE MANAGER >PROPERTIES >

Once you know the base address and interrupt, you can specify them on the LOAD BIN line, with the appropriate board type. NI emulates the AT-GPIB board type, and INES emulates the PC2 board type.

When installed correctly, the INES card returns a message during boot up from the program loaded with the AUTOEXEC.BAT. The message gives the BOARD ADDRESS or IO in hexadecimal and the INTERRUPT selected. The message "DONE" at the end of this batch program indicates everything is ready. The following is an example LOAD BIN command for the INES card:

```
LOAD BIN "GPIBN;BO PC2 120 IN 5"
```

This indicates a board type (BO) of PC2 which is the NEC7210 GPIB controller chip, a Board Address (BA) of 0x120h as shown from AUTOEXEC.BAT, and an interrupt (IN) of 5 as indicated by AUTOEXEC.BAT. HTBasic for Windows generates a message on the lower portion of the screen for a successfully loaded GPIB driver.

Chapter 7
Transferring Programs and Data from HP BASIC

This chapter presents what you need to know to transfer programs and data files between HP BASIC and HTBasic. It discusses HTBasic file types, remote access across networks or serial link, files created by other computers and the LIF diskette file transfer utilities. Utilities available from third parties allow transfer of files residing on LIF diskettes, hard disks and tape. *Hpcopy* is by far the easiest way to transfer programs and data.

File Types

HTBasic can work with both typed files (such as BDAT, ASCII and PROG) and files without a type or "ordinary" files. Both kinds of files are discussed in the following paragraphs.

Typed Files

Most HTBasic or HP BASIC files have a file type, such as BDAT, ASCII or PROG. The typed file information is stored in a file header at the beginning of the file. Two types of headers are in common use: HP LIF and HTBasic.

HTBasic can identify and use HP BASIC typed files with HP LIF headers. Data can be in BDAT or ASCII files. HP BASIC PROG files must be saved in ASCII or HP-UX format; use the SAVE statement, not the STORE statement to save programs for exchange. HP LIF file headers are either 512 or 768 bytes in length.

HTBasic can also, of course, use HTBasic file headers. HTBasic file headers are 256 bytes in length.

The LIF diskette utilities, described later in this chapter, supports typed files with either type of file header.

Ordinary Files

Recent releases of HP BASIC support files without a type. HP BASIC calls these files "HP-UX" files when present on a LIF or UNIX volume and "DOS" files when present on a DOS volume. HTBasic calls these files "ordinary" files. An ordinary file is created with the CREATE statement (as opposed to CREATE ASCII or CREATE BDAT). In a catalog of a LIF diskette, an ordinary file is listed as "HP-UX". When an ordinary file is copied to a Windows disk, it remains an ordinary file, but is listed without a type by the HTBasic CAT statement. Files created on the PC by other programs are ordinary files.

Remote Access

HTBasic can access files on other computers over a network by accessing a remote mounted volume or by using file transfer utilities across the network or across a serial link.

Mounted Network Volumes

HTBasic running on a computer can access files on remote computers directly if the disks containing the files on the remote computer(s) are mounted as network volumes on the computer running HTBasic. To access the remote files, simply include the local name of the remote directory when specifying file names. See your system manager for this information.

Network File Transfer

If there are files on a computer connected to your network but not mounted in your computer's file system, if that computer has an ftp server and if you have TCP/IP utilities for your PC, you can use the *ftp* and *rcp* file transfer utilities to copy the files to your computer's disk and then use them. Note that typed files are considered to be binary files by ftp, so you must use the binary command with ftp when copying these file types.

Other File Transfer Utilities

Two computers connected through a network or with a serial link can transfer files using a file transfer utility, for example, the Kermit program. As with other types of connections, BDAT and LIF ASCII files are considered to be binary files, so you must use the binary mode on both the sending and receiving sides of the file transfer when copying these file types. In Kermit, this is done with the "set file type binary" command.

In some instances, data can be transferred between two computers by hooking them together with a serial or IEEE-488 interface and writing a small program on each computer to transfer programs and data between the two. Transfer programs as if they were ASCII data.

File transfer utilities are available from third parties. A list is included in the *HTBasic User's Guide,* Chapter 2, "Porting HP BASIC Programs to HTBasic."

Files Created by Other Computers

LIF ASCII, BDAT and ordinary files created by other computers can be used by HTBasic. PROG files created by other computers generally cannot. The following paragraphs discuss each file type and any associated restrictions.

LIF ASCII Files

LIF ASCII files created by HTBasic on PC's or other computers or by HP BASIC can be directly read and written by HTBasic. If the files are transferred using a file transfer utility, binary mode should be used.

BDAT Files

BDAT files created by HTBasic on PC's or other computers or by HP BASIC can be read and written by HTBasic. This is true even if the two computers use different byte ordering. If the files are transferred using a file transfer utility, binary mode should be used.

PROG Files

PROG files created by different versions of HTBasic can be exchanged only if the two computers use the same byte ordering. Presently, the DOS and Windows versions produce LSB PROG files while the Sun SPARCstation and HP 700 Workstation versions produce MSB PROG files. PROG files created by HP BASIC **cannot** be read or written by HTBasic. LOADing an incompatible PROG file results in error 58, "Improper File Type." ASCII and ordinary format programs can be used to exchange programs between versions with incompatible PROG files. Use GET to load an ASCII or ordinary format program file.

HTBasic PROG files with the wrong byte order are listed by the CAT statement with a file type of "PROGL" or "PROGM", where the trailing L or M indicates the file has LSB or MSB byte ordering. PROG files created by HP BASIC are listed with the file type "HPPRG". Compatible PROG files are listed with a file type of "PROG".

Ordinary Files

Ordinary files (DOS, UNIX or HP-UX ASCII files) can be both read and written by HTBasic. If ordinary files written with FORMAT OFF are transfered using a file transfer utility, binary mode should be used. If ordinary files written with FORMAT ON are transfered using a file transfer utility, use its ASCII mode ("ascii" in *ftp* and "set file type ascii" in Kermit).

HTBasic is able to work with ordinary, FORMAT ON files regardless of whether the file was written with a line termination of CR/LF (carriage-return/line-feed) or just LF. Other programs, such as text editors, may not be as flexible. CR/LF is considered the standard for DOS and Windows. LF is considered the standard for UNIX.

LIF Diskette Utilities

The LIF diskette utilities are used to copy program and data files between LIF diskettes and DOS disks. Where necessary, the file header is converted and then the data is copied to the new file. These utilities work with single or double sided diskettes, formatted with 256 or 1024 byte sectors.

Hpcat prints a CATalog of files on an HP LIF diskette. *Hpcopy* copies files between LIF diskettes and DOS disks. *Hppurge* deletes files on LIF diskettes. These commands are used at the DOS prompt. To use them while running HTBasic, use the EXECUTE command, i.e. EXECUTE "hpcat 0:". You may use these commands if they are in the current directory or if a PATH has been set up to the directory in which they are stored (see your DOS manuals).

Problems

It is fairly common to find a LIF floppy that cannot be read by one or more PC floppy disk drives. If you have problems, try several computers from different manufacturers and you can usually find one that works. If you have problems, read the "Common Problems" section later in this chapter.

HPCAT

Display a CATalog of files on an HP LIF diskette.

Syntax:

HPCAT drive:

where:

drive = 0, 1, ...

Sample:

```
HPCAT 0:         ! drive A
HPCAT 1:         ! drive B
```

Description:

This DOS command allows you to display a catalogue (directory) of the files on an HP LIF diskette. An HP LIF diskette is one that was initialized with the HP BASIC INITIALIZE command on an HP 9000 Series 200/300 workstation. The diskette must be inserted into a PC diskette drive. Both 5-1/4 and 3-1/2 inch diskette drives are supported. Disk drives connected to the HP-IB are not supported.

The first drive, "A", is number 0, the second drive is number 1. It is recommended that you only use diskette drives "A" and "B". In limited circumstances, other drives will work, but some experimentation is needed to use them. For other drives, you must discover the diskette drive number. Try values from 0 to 9. If none work, your drive or system may not support the 256 or 1024 byte sectors required by HP diskettes.

HPCOPY

Copies files between an HP LIF diskette and a DOS disk.

Syntax:

HPCOPY [drive:]lif-filename [disk:]dos-filename [-LIF]

HPCOPY [disk:]dos-filename [drive:]lif-filename

where:

drive = 0, 1, 2,...

disk = A, B, C,...

lif-filename = a legal LIF filename, may include wildcards

dos-filename = a legal DOS filename, may include wildcards

Sample:

```
HPCOPY 0:hpfile C:DOSFILE
HPCOPY C:GOOD.BYE 3:Hello
HPCOPY A:DOSFILE 1:HPbdat
HPCOPY 0:lifASCII C:\DIR2\DOSFILE
HPCOPY 0:* C:
HPCOPY C:\HTB\DATA\D?T* 0:
```

Description:

This DOS command allows you to copy ASCII, BDAT and ordinary files from HP LIF diskettes to DOS disks or vice-versa. An HP LIF diskette is one that was initialized with the HP BASIC INITIALIZE command on an HP 9000 Series 200/300 workstation. The diskette must be inserted into a PC diskette drive. Both 5-1/4 and 3-1/2 inch diskette drives are supported. Disk drives connected to the HP-IB are not supported.

Programs **must** be in ASCII format. PROG files are **not** supported. BDAT files can be transferred with the limitations noted below.

When files are copied no translation is done on the file contents; a LIF ASCII file remains a LIF ASCII file. However, you may write simple HTB programs that do the translation if you need to use the data files with other DOS programs. An example translation program is given in Chapter 9, "Files" of the *User's Guide.* This conversion is not required if the data files will be used only by HTBasic, because HTBasic knows how to use HP BASIC files.

Filenames

The lif-filename should be the legal name (including correct capitalization) of a LIF file. If the LIF diskette is in drive A or B, prefix the name with "0:" or "1:". If the LIF diskette is in another drive, specify the drive number (as explained above under "HPCAT").

The dos-filename should be the legal name of a DOS file. This name optionally can include a DOS drive letter and a full path. If no drive is specified, the default drive is used. This drive must be different from the drive containing the LIF diskette. If no path is given, the present directory is used.

If the destination file already exists on a DOS disk, it is overwritten. If the destination file already exists on a LIF diskette, an error is reported and the file is left unchanged.

Copying Files to HP BASIC

HTBasic can create typed files that are compatible with Series 200/300 computers, but that is not the default. If you plan on transferring files back to HP BASIC systems or sharing the files with HP BASIC systems on a network, you should execute the CONFIGURE CREATE "HP" statement before creating any files. Any BDAT or ASCII files created after this statement is executed are completely compatible with Series 200/300 computers.

If you do not plan on transferring files back to HP systems, it is best to use CONFIGURE CREATE "HTB" (the default), since the file header is smaller and the fastest byte ordering will be used.

Using Wildcards

Wildcards can be used to transfer more than one file at a time. If a wildcard is used, it should only be used in the source filename, not the destination filename. The destination should specify only the drive (and directory if the destination is a DOS disk). Legal wildcards are "*" and "?". An asterisk will match any one or more characters starting at that location. A question mark will match any one character at that location. These conventions are also explained in your DOS manual.

Because of the differences in legal LIF and DOS filenames, filenames may be translated. A LIF filename is limited to at most ten characters and all ASCII characters except "space,:,<,|" are legal LIF filename characters. A DOS filename is limited to 8 characters, a period and 3 characters and all characters except "\/:|<>+=;," and control characters whose ASCII value is less than the space character are legal DOS filename characters. Also, lowercase letters are converted to uppercase by DOS.

When transfering a file from a DOS to a LIF disk the first ten characters of the filename, including the period, are used for the LIF name. Any illegal LIF characters in the DOS filename are translated to an underscore character.

When transfering a file from a LIF disk to a DOS disk the filename is converted to uppercase and if necessary, a period is inserted after the eighth character. Because DOS converts lowercase letters to uppercase, two LIF files named "Aa" and "aa" will be transfered into the DOS filename "AA". The first file transfered will be overwritten by the second file transfered.

Note: Wildcards are not supported by the HPCOPY that is supplied with the Demonstration Version of HTBasic.

-LIF Option

With the -LIF header option, when copying files from a LIF diskette to DOS, if the file type is ASCII or BDAT, the file is created on the DOS disk with an HP LIF file header rather than an HTBasic file header. When copying files from a LIF diskette to DOS, all other file types besides HP-UX are automatically created on the DOS disk with an HP LIF file header. When copying files from a DOS disk to LIF, this option is ignored; files with HP LIF file headers are handled automatically.

HPPURGE

Deletes files from an HP LIF diskette.

Syntax:

HPPURGE [drive:]lif-filename

where:

drive = 0, 1, ...

lif-filename = a legal LIF filename

Sample:

```
HPPURGE 0:LIFFILE
HPPURGE 1:Hello
```

Description:

This DOS command allows you to delete ASCII, BDAT and ordinary files from HP LIF diskettes. An HP LIF diskette is one that was initialized with the HP BASIC INITIALIZE command on an HP 9000 Series 200/300 workstation. The diskette must be inserted into a PC diskette drive. Both 5-1/4 and 3-1/2 inch drives are supported. Diskette drives connected to the HP-IB are not supported.

The lif-filename should be the legal name (including correct capitalization!) of a LIF file. The first drive, "A", is number 0, the second drive is number 1. It is recommended that you only use diskette drives "A" and "B". In limited circumstances, other drives will work, but some experimentation is needed to use them. For other drives, you must discover the diskette drive number. Try values from 0 to 9. If none works, your drive or system may not support the 256 or 1024 byte sectors required by HP LIF diskettes.

Common Problems

The following paragraphs document some common problems you may experience trying to use HPCAT, HPCOPY and HPPURGE. If you experience a problem, glance through the headings to find the answer to your question.

HPCOPY Says the File is Not Present, But It Is. Use HPCAT to find the *exact* spelling, including upper and lower case. Remember that LIF filenames are case-sensitive and DOS filenames are not. "HELLO" and "hello" refer to different LIF files, but the same DOS file.

HPCOPY Gives An Error Part Way Through the File and Then Stops. This error is common on double-sided 3-1/2 inch diskettes. It usually means your computer ROM BIOS is incapable of reading some LIF diskettes. Try another PC or another LIF diskette. Often a single sided LIF diskette will work where a double sided will not. A 9122 drive can be instructed to initialize a double sided (or single sided) diskette as if it were single sided by using INITIALIZE option 4. If the diskette has previously been initialized, you need first to remove the HP format information from the second side by formatting on your PC (or by using INITIALIZE option 2) before initializing as single sided.

The Error "Bad command or filename" Is Reported. HPCAT or HPCOPY is not in the current directory and no PATH is set up to point to them. Read your DOS manual or use CD \HTB to change to the HTB directory.

Error 910 Is Reported. You are currently running HTBasic, not DOS. HPCAT, HPCOPY and HPPURGE are DOS commands, not BASIC commands. Issue the commands in a MS-DOS window. Or use the QUIT command to exit to DOS before proceeding. Or use the HTBasic EXECUTE statement:

```
EXECUTE "HPCAT 0:"
```

The Error "Sector not found..., Abort, Retry, Ignore?" Is Reported. This or similar errors usually mean that you have inserted a LIF diskette in the current DOS drive. While your DOS prompt is "A" you cannot put a LIF diskette in drive A.

The Light on the Drive Does Not Turn On. If your diskette drive has a letter other than "A" or "B", you may not be able to use it. In limited circumstances, other drives will work, but some experimentation is needed to use them. For other drives, you must discover the diskette drive number. Try HPCAT with values from 0 to 9 until the light comes on the drive you are trying to use.

HPCOPY Does Not Work With DR-DOS. If HPCOPY does not work with DR-DOS, try putting the following switch in your CONFIG.SYS file:

```
SUPERPCK -d
```

where *d* is the drive where the HP LIF disk is located. The SUPERPCK option will not cache information from the specified drive.

Chapter 8
Customizing the Environment

HTBasic provides a keyboard function key macro facility and a powerful CONFIGURE statement that can both be used to customize HTBasic to more nearly match the configuration of an HP 9000 Series 200/300 computer and to alter the way the HTBasic environment works. Each of these features will be discussed on the following pages.

Softkey Macros

The softkey macro facility allows you to define a macro sequence and associate it with a function key. When you press that function key the macro sequence is entered into the input buffer just as if you had typed it in from the keyboard. A softkey definition can be quite complex and it can even call other softkey definitions. You can assign, edit, read, delete, list, store and load softkey macros.

Assigning Softkeys

The HTBasic softkey keyboard macro facility allows great flexibility in defining special purpose function keys. For Example, to define a softkey macro to RENumber and INDENT the current program and then put you into Edit mode, you could define function key 9 as follows:

```
10 L$ = CHR$(255)&"#"   !Clear Line
20 S$ = CHR$(255)&"K"   !Clear Screen
30 E$ = CHR$(255)&"E"   !Execute Input
40 D$ = CHR$(255)&"D"   !EDIT
50 SET KEY 9,"INDENT  RENUMBER"&L$&S$&"REN"&E$"INDENT"&E$&D$&E$
```

You are only limited by your imagination in defining softkey macros.

Editing Softkeys

The EDIT KEY statement allows interactive editing of softkey macros. Enter EDIT KEY *n*, where *n* is the softkey number or press EDIT followed by the softkey function key you wish to edit and then the ENTER key. If the softkey has been defined, the old definition will be displayed. You can modify the old definition by overstriking the existing characters or inserting new characters. Pressing ENTER saves the new softkey definition.

Reading Softkeys

READ KEY returns one or more softkey macro definitions. Specify the key number of the softkey macro to read. If a simple string or array element is specified, then only one key is returned. If a string array is specified, then successive keys, starting with the one specified, are returned into the elements of the string array. For example

```
READ KEY 2,Keytwo$          ! One macro
READ KEY First_key,Keys$(*) ! Several macros
```

Deleting Softkey Macros

Softkey macro definitions are removed from memory with the SCRATCH KEY statement. For example:

```
SCRATCH KEY 3
```

removes the definition for softkey macro number 3. If no key number is specified all softkey macro definitions are removed from memory.

Listing Softkey Macros

The LIST KEY statement lists the current softkey macro definitions on the PRINTER IS device. For example:

```
LIST KEY
```

outputs all the softkey macro definitions. You also may specify an interface select code for the output.

Storing Softkey Macros

The STORE KEY statement saves all the current softkey macro definitions in a new file of type BDAT. HTBasic softkey BDAT files are compatible with HP BASIC softkey BDAT files when HPCOPY is used to move the files. The definition for each defined softkey is written to the file by outputting two items using FORMAT OFF. The first item is an integer, specifying the key number. The second item is a string, giving the key definition. For example:

```
STORE KEY "Helpers"
```

saves the current softkey macro definitions into a new BDAT file named Helpers. If the file already exists, an error is reported. Use RE-STORE KEY to update an existing file.

Loading Softkey Macros

Use the LOAD KEY statement to reenter the softkey macros into the computer from a BDAT file. An HP BASIC softkey BDAT file is compatible with HTBasic and may be directly loaded. Each softkey macro defined in the file replaces any previous definition. For example:

```
LOAD KEY "Helpers"
```

reloads the softkey macros defined in the file Helpers into the computer.

CONFIGURE Statement

The CONFIGURE statement allows you to customize HTBasic to more nearly match the configuration of an HP 9000 Series 200/300 computer and to alter the way the HTBasic environment works. Using the CONFIGURE statement will allow many existing HP BASIC programs to run without alteration. The CONFIGURE statements should be placed in your AUTOST file so that they are executed each time you use HTBasic. The CONFIGURE statement allows you to:

• create HP workstation compatible BDAT files
• create typed files with HP LIF file headers
• control implicit variable and string dimensioning
• configure the DUMP statement for different printer or image file formats
• customize keyboard key assignments
• define additional LABEL characters
• enables use of long file names
• specify a Windows path specifier to use in the place of an HP disk drive
• change the value of PRT
• create Windows ASCII program files.

HP Compatible BDAT Files

CONFIGURE BDAT {MSB|LSB} FIRST specifies the byte ordering to use with each BDAT file created after this statement is executed. By default, BDAT files are created with the native byte order of the computer. This statement was previously used to create BDAT files that could be copied (using HPCOPY) to an HP BASIC workstation. CONFIGURE CREATE is now the recommended method for creating files for interchange with HP BASIC. CONFIGURE BDAT only affects files created with HTBasic headers; files with HP LIF headers are not affected.

Controlling File Header Type

CONFIGURE CREATE specifies the kind of file header to use when creating a typed file (LIF ASCII or BDAT). HTBasic can always use files with either header, regardless of the setting of CONFIGURE CREATE. The setting affects file creation only. A CAT listing in SRM format shows the kind of file header of each file in the System Type column.

Use HP LIF headers if you wish to create data files that are simultaneously accessed over a network by HTBasic and HP BASIC. Files with HP LIF headers can also be "binary" copied among DOS or UNIX media for access by the HP Language Coprocessor (Viper card), HP BASIC and HP BASIC/UX. To specify HP LIF headers, use:

```
CONFIGURE CREATE "HP"
```

By default, HTBasic creates HTBasic file headers, since they are two or three times smaller than HP LIF headers. BDAT files with HTB headers can also be created with data in either LSB or MSB byte ordering (see CONFIGURE BDAT). File operations are much faster when the byte ordering of the file matches the byte ordering of the computer. Files with HTB file headers, when copied with HPCOPY, are completely compatible with HP BASIC. Example:

```
CONFIGURE CREATE "HTB"
```

Controlling Implicit Dimensioning

CONFIGURE DIM turns implicit variable and string dimensioning on or off. By default it is on and if a variable is never declared, it is assumed to be of type REAL. If a string is never declared, it is assumed to have a maximum length of 18. If an array is never declared, it is implicitly declared having the number of subscripts found in its first occurrence, with each dimension having the default OPTION BASE lower bound and an upper bound of ten. Example:

```
CONFIGURE DIM ON
```

When CONFIGURE DIM is OFF, then each variable, string and array **must** be explicitly declared using a REAL, INTEGER, COMPLEX or DIM statement. During prerun, any undeclared variables generate an error message that is written to the message line. Turning off implicit variable and string dimensioning can greatly simplify finding misspelled variable names. If a program has already been prerun, CONFIGURE DIM OFF will not report any undeclared variables until another prerun occurs. To force a prerun to occur, change a program line and press the STEP key. Example:

```
CONFIGURE DIM OFF
```

DUMP GRAPHICS Printer Type

CONFIGURE DUMP specifies what graphic printer language or image file format the DUMP GRAPHICS statement uses. Chapter 4, "Printer and Image File Drivers," explains how to use CONFIGURE DUMP so that DUMP GRAPHICS works with many different printer types and file formats. For example, most HP printers support PCL or the HP Raster Interface Standard. If you wish to use an HP LaserJet for screen dumps, use the following statement to change to the HP printer control language:

```
CONFIGURE DUMP TO "PCL"
```

Non-Latin-1 Character Set Keyboard Remapping

CONFIGURE KBD defines keyboard mappings for character sets other than Latin-1. When in effect, CONFIGURE KBD substitutes characters from the specified string for characters that come from the keyboard. This remapping is good for ASCII characters in the range 0 to 255, but does not apply to function keys. CONFIGURE KBD is not intended to be a complete keyboard driver, it merely substitutes one ASCII value for another. The syntax is:

CONFIGURE KBD *first-char* TO *string-name$*

The range of ASCII values that are remapped starts at the character number specified by *first-char*. The string specifies the ASCII values that should be substituted for values in that range. For example, to remap four keys starting with character number 65 use the following:

```
CONFIGURE KBD 65 TO "DCBA"
```

Redefining Function Keys

The CONFIGURE KEY statement can be used to redefine key assignments. A single key can have more than one definition, if the definitions apply to different shift states or conditions. For example, a key can be defined to have one function when the shift key is pressed, and another definition when the shift is not pressed. A key definition consists of these four parts:

1) The key number,
2) The shift conditions to be examined (masked) when the key is pressed,
3) The value those shift conditions must have,
4) The RMB function to execute when the proper key is pressed with the proper shift conditions.

The CONFIGURE KEY statement has four forms that are used for the different aspects of defining a key. The first form,

CONFIGURE KEY *key-number* TO 256

removes all prior definitions a key may have. It is usually a good idea to remove prior definitions of a key to avoid unwanted side-effects. *Key-number* specifies what key the action applies to. A utility, keynum, is included with HTBasic to determine the *key-number* for a key.

In the HTBWin program group or program folder, double-click on the KeyNum or KEYNUM.EXE icon. Press the key of interest. The *key-number* for that key is printed in the window. Press Alt-F4 to quit.

A key definition is made using the final three forms of the CONFIGURE KEY statement:

CONFIGURE KEY *shift-mask* TO 257 ! Set shift-mask for define
CONFIGURE KEY *shift-value* TO 258 ! Set shift-value for define
CONFIGURE KEY *key-number* TO *function* ! Create a definition

Shift-mask and *shift-value* specify what shift conditions must be in effect when the key is pressed and what conditions will be ignored. In the table below, find the conditions you wish to have an affect and write down the associated *Shift-mask* and *Shift-value*. Leave out the values for the conditions to be ignored. Then add together all the *shift-mask* values and all the *shift-value* values.

Table for Shift-mask and Shift-value Calculation

Shift Conditions		Shift-mask	Shift-value
Ignore all shift conditions	0	0	
Shift key not pressed	1	0	
Shift key pressed	1	1	
Control key not pressed	2	0	
Control key pressed	2	2	
Alt key not pressed	8	0	
Alt key pressed 8	8		
Caps Lock off	64	0	
Caps Lock on	64	64	
System Menu	1792	0	
User 1 Menu	1792	256	
User 2 Menu	1792	512	
User 3 Menu	1792	768	
KBD CMODE OFF	2048	0	
KBD CMODE ON	2048	2048	

Only one menu may be specified: system, user 1, user 2 or user 3. The Alt key is an alternate shift key and may have different names on different keyboards. It is not a good idea to use the control key, because that prevents its conventional use. (The control key is usually used to insert a function key's two characters into a string literal rather than execute it.)

Function specifies the Keyboard Function to assign to the key. The function is specified using the second character produced when the function key is pressed. Check the "Alphabetical Keyboard Functions List" in Chapter 2 for the second characters used by each HTBasic function. The numeric value of the character is used in the CONFIGURE KEY statement. For example, the table lists "V" as the 2nd character for the Previous Line (DOWN) keyboard function. If you wish to define a key to do the DOWN function, substitute NUM("V") for *function*.

A key redefinition is usually done in groups of three CONFIGURE KEY statements. And it is always a good idea to delete any previous definitions made to a key before making any new definitions. The reason is that if a previous definition specified *Shift-mask* and *Shift-value*s that are less restrictive (more inclusive) than the new definition, then the previous definition will be used; the new definition will never be used.

CONFIGURE KEY Example. Suppose you wish to swap the actions of the and ↓ keys. By default, two functions are assigned to each of these keys:

Keyboard Function	Generic Name	PC Keyboard
Next line	UP	
End of output/program	END	Shift-
Previous line	DOWN	↓
Begin of output area	BEGIN	Shift-↓

After we redefine the keys the definitions will be

Keyboard Function	Generic Name	PC Keyboard
Next line	UP	↓
End of output/program	END	Shift-↓
Previous line	DOWN	
Begin of output area	BEGIN	Shift-

Use the keynum utility to find the key numbers for the and ↓ keys. Under Windows, these values are usually 38 and 40, respectively.

```
10    ! Define Up-Arrow Key
20    CONFIGURE KEY 38 TO 256       ! Delete previous  key defs
30    CONFIGURE KEY 1 TO 257        ! Shift mask: Shift key
40    CONFIGURE KEY 0 TO 258        ! Shift value: Not pressed
50    CONFIGURE KEY 38 TO NUM("V")  !  key is DOWN
60    CONFIGURE KEY 1 TO 257        ! Shift mask: Shift key
70    CONFIGURE KEY 1 TO 258        ! Shift value: Pressed
80    CONFIGURE KEY 38 TO NUM("T")  ! Shift- is BEGIN
90    ! Define Down-Arrow Key
100   CONFIGURE KEY 40 TO 256       ! Delete previous ↓ key defs
110   CONFIGURE KEY 1 TO 257        ! Shift mask: Shift key
120   CONFIGURE KEY 0 TO 258        ! Shift value: Not pressed
130   CONFIGURE KEY 40 TO NUM("^")  ! ↓ key is UP
140   CONFIGURE KEY 1 TO 257        ! Shift mask: Shift key
150   CONFIGURE KEY 1 TO 258        ! Shift value: Pressed
160   CONFIGURE KEY 40 TO NUM("W")  ! Shift-↓ is END
170   END
```

Lines 20 and 100 delete any previous definitions. Then the four definitions are made using groups of three: 30-50, 60-80, 110-130 and 140-160.

If the *Shift-mask* is not specified, the last *Shift-mask* specified will be used. If no *Shift-mask* has ever been specified, a value of 0 is used. The same is true for the *Shift-value*. Thus, lines 60, 110 and 140 in the previous example are redundant. If line 100 is moved to 21, 130 to 51 and 160 to 81, then lines 120 and 150 can also be eliminated.

CONFIGURE KEY Example. CONFIGURE KEY is sometimes useful for eliminating an HTBasic key definition so the key reverts to its default use. For example, HTBasic defines Alt-F4 as Pause Program. If you wish to use it to Quit HTBasic, you must eliminate the HTBasic definition. Assuming F4 is key number 115, the following example eliminates all definitions for F4, including Alt-F4, then redefines all definitions except Alt-F4:

```
10    CONFIGURE KEY 115 TO 256  ! Delete all F4 definitions
20    !
30    CONFIGURE KEY 3849 TO 257 ! Mask:  KBD CMODE, Menu, Alt, Shift
40    !
50    CONFIGURE KEY 0    TO 258 ! Value: KBD CMODE OFF, System
60    CONFIGURE KEY 115  TO 80  ! Def:   KBD CMODE OFF, System, F4
70    CONFIGURE KEY 1    TO 258 ! Value: KBD CMODE OFF, System, Shift
80    CONFIGURE KEY 115  TO 33  ! Def:   KBD CMODE OFF, System, Shift-F4
90    CONFIGURE KEY 256  TO 258 ! Value: KBD CMODE OFF, User 1
100   CONFIGURE KEY 115  TO 52  ! Def:   KBD CMODE OFF, User 1, F4
110   CONFIGURE KEY 512  TO 258 ! Value: KBD CMODE OFF, User 2
120   CONFIGURE KEY 115  TO 99  ! Def:   KBD CMODE OFF, User 2, F4
130   CONFIGURE KEY 768  TO 258 ! Value: KBD CMODE OFF, User 3
140   CONFIGURE KEY 115  TO 107 ! Def:   KBD CMODE OFF, User 3, F4
150   !
160   CONFIGURE KEY 2057 TO 257 ! Mask:  KBD CMODE, Alt, Shift
170   !
180   CONFIGURE KEY 1    TO 258 ! Value: KBD CMODE OFF, Shift
190   CONFIGURE KEY 115  TO 118 ! Def:   KBD CMODE OFF, User n, Shift-F4
200   CONFIGURE KEY 2048 TO 258 ! Value: KBD CMODE ON
210   CONFIGURE KEY 115  TO 51  ! Def:   KBD CMODE ON,  F4
220   CONFIGURE KEY 2049 TO 258 ! Value: KBD CMODE ON, Shift
230   CONFIGURE KEY 115  TO 100 ! Def:   KBD CMODE ON,  Shift-F4
240   END
```

Defining New LABEL Characters

CONFIGURE LABEL defines additional characters for use with the LABEL statement. Characters in the range 33 to 255 may be defined. You may define one character by giving the character number and a simple string or several characters by giving the starting character number and a string array. To delete a definition, use a zero length string for the definition. Each character in the definition strings has the form CHR$(Move + x*16+ y), where Move is 0 or 128, x ranges from 0 (far left) to 7 and y ranges from 0 (bottom) to 15. The baseline is y=5.

The following example defines the character "H":

```
A$ = CHR$(133)&CHR$(14)&CHR$(238)&CHR$(101)&CHR$(138)&CHR$(106)
CONFIGURE LABEL 72 TO A$
```

More information is presented in Chapter 13, "International Language Support" in the *User's Guide.*

Enabling Use of Long File Names

Under Windows 95 and NT, long filenames are allowed in addition to the standard 8.3 names. The filenames can be about 256 characters long and can have embedded spaces. However, by default HTBasic removes spaces from file specifiers and CAT listings don't have enough room for long filenames. To enable display and use of long filenames, use the statement

```
CONFIGURE LONGFILENAMES ON
```

With LONGFILENAMES ON, spaces are not deleted from directory and file specifiers since they may be significant. Also, the listing format for CAT is changed to accommodate varying length filenames. It is roughly modelled after the Windows NT DIR command listing format.

HP MSUS to Windows disk substitutions

CONFIGURE MSI defines a table of msus substitutions. Whenever an HP msus is used as part of a file-specifier, the table is checked for an exact match, including case. It is not sufficient for the msus to be equivalent, i.e. ":CS80,700,0" does not match ":CS80,700". If an exact match is found, the Windows path-specifier is used in its place. If no match is found, an error will be reported. The syntax is:

CONFIGURE MSI hp-msus TO path-specifier

In the DOS path-specifier, directory names must end with a directory separator character "\. To replace an existing entry in the table, specify a new DOS path-specifier. To delete an existing entry in the table, specify a zero length DOS path-specifier. To turn CONFIGURE MSI off or on, use

CONFIGURE MSI {ON|OFF}

As an example, let's assume that you have lots of programs written that use two HP disk drives on your HP BASIC workstation. They are ":INTERNAL,4", a floppy and ":,700,0", a hard disk. You wish to use Drive A on your PC whenever the programs would have used the internal drive on your workstation. And you wish to use the subdirectory "\HARDDISK" of Drive C on your PC whenever the programs would have used the hard disk on your workstation. This is done by putting the following two lines in your AUTOST file:

```
CONFIGURE MSI ":INTERNAL,4" TO "A:\"
CONFIGURE MSI ":,700,0" TO "C:\HARDDISK\"
```

Changing the value of PRT

On most HP BASIC workstations a printer is hooked to the IEEE-488 interface and is set to primary address 1. Thus, the device selector is 701. This convention is followed so closely that a special function, PRT, was created with the value 701. Many programs use this fact to switch output to the printer with the statement.

```
PRINTER IS PRT
```

Under Windows, printers are accessed through the printer driver, which is accessed in HTBasic through device selector 10. Many existing programs use PRT as the device selector for the printer. So that these program continue to work, PRT has been changed to 10. If you are using an IEEE-488 printer with ISC 1, you may wish that PRT had the value 701 as it did under HP BASIC. To allow you to change PRT, the CONFIGURE PRT statement has been added. CONFIGURE PRT need only be used if you are porting existing programs that reference PRT and you do not wish to use the Windows printer driver.

CONFIGURE PRT specifies the device-selector that the PRT function returns. If you are using an IEEE-488 bus printer at address one, you will probably want to change PRT back to 701 to match the HP BASIC definition of PRT:

```
CONFIGURE PRT TO 701
```

Windows Compatible ASCII Files

CONFIGURE SAVE ASCII sets the file type SAVE uses when saving a file to disk. CONFIGURE SAVE ASCII ON, the default, produces a LIF compatible ASCII file. This type of file is useful for exchanging programs with older HP workstations that cannot use Ordinary files. See Chapter 7, "Transferring Programs and Data from HP BASIC." CONFIGURE SAVE ASCII OFF produces Windows ASCII ordinary files. Such files are compatible with all popular program editors, most word processors and recent revisions of HP BASIC. RE-SAVE produces the same file type as an existing file or the file type specified by CONFIGURE SAVE ASCII if no file exists. GET can read either file type, as well as Viper-I ASCII and Viper-II ASCII.

Note: If you use CONFIGURE SAVE ASCII OFF you should not embed carriage-returns or line-feeds in string literals since GET will interpret them as end-of-line indicators.

Chapter 9
If You Have Problems

The following sections describe common problems users of HTBasic may face. If you are having problems, you should read through the following headings to see if the answer to your question is given here.

Keyboard Lockup

Sometimes HTBasic seems locked up and some sort of mouse interaction is necessary to revive it. This is not a bug, but a "feature" of the Windows environment and occurs when the Window's System Menu (present in the upper-left corner of the window) is selected. Clicking on the system menu or pressing and releasing the Alt key both cause the menu to be selected, as indicated by a change in color. To deselect it, press Alt again or click anywhere on the screen with the mouse.

Graphics Disappear

When graphics buffering is off (the default), hiding and then exposing any part of the HTBasic window causes the graphics to be discarded. To preserve the graphics, turn graphics buffering on as shown in the example below. Turning buffering on slows down the performance of graphic operations, but allows the graphics to be restored when hidden and then exposed.

```
C> htbwin -gr on
```

Error 163, Interface not present

If you plan on using a serial (RS-232) interface or an IEEE-488 card you must modify your AUTOST program to load the needed drivers. Chapter 6, "I/O Device Drivers" gives instructions for loading and customizing the drivers. If no driver is loaded, a statement such as

```
OUTPUT 719;A$
```

will produce error 163, Interface not present.

Softkeys Not Visible

Depending on the width of the window and the size of the font used, the softkeys and run indicator on the right side of the screen may be clipped off. For example, using the Windows VGA display driver and default font, 79 columns are displayed and the single-character run indicator in column 80 is clipped. Maximize the window to display 80 characters. Alternately, use a display driver with higher resolution or a smaller font. The following example selects a small size of the "Courier New" truetype font:

```
C> htbwin -fn "Courier New,15"
```

Colors Wrong or Dithered

When using the -cu ReadOnly color map usage method, colors may not exactly match those expected. Line and text colors are chosen from the available Windows static colors. Area fill colors are dithered to approximate the desired color, even when AREA PEN is used. These problems can be solved by using a 256-color Windows display driver rather than a 16-color driver. Chapter 1 contains more information about color map usage methods.

Cannot be Run in DOS Mode

HTBWin cannot be started from a Windows 3.1 MS-DOS prompt window. If you attempt to do so, you will receive the message, "This program cannot be run in DOS mode." Windows NT or Windows 95 or later are required to start Windows programs from a DOS prompt. See "Starting HTBasic" in Chapter 1, "Installing the Windows Version."

This error can also result from trying to run HTBWin while Windows is not running. Start Windows and then start HTBWin. Also, older versions of some programs that replace Program Manager, such as Dashboard, give this error. If this occurs, use Program Manager to start HTBWin.

Bad Command Error

If you get an error when you try to run HTBasic from an MS-DOS window (something like "Bad command or file name"), then you may have mistyped the HTBasic command. Check your spelling and make sure you leave a space before any switches. For example,

```
htbwin-w 200k
```

is wrong. If you are sure you have typed the command correctly, then you are in the wrong directory and haven't set up a PATH to the HTBasic directory. You can use the DOS "CD" command to move to the correct directory each time you wish to run HTBasic or you can add the HTBasic directory to your PATH. Consult your Windows/DOS manual for information on both of these commands.

Error larger than 10000

An error number larger than 10000 can occur if Windows returns an error that HTBasic was not expecting. This error is a combination of the Windows error number and 10000. For example 11157 is Windows error 1157.

Error 2009, Wrong Revision

If you get Error 2009, Wrong Revision, you are trying to load a 1.x/2.x PROG file without converting it. Some of the improvements made since Release 3.0 are made possible by a new PROG format, "Power-PROG Format." This new format has the advantage of being smaller in most cases, faster in many cases, speeds up prerun and will allow TransEra to add significant extensions to RMB in the future. This internal method of representing programs is the result of incrementally compiling your programs.

To convert an old PROG format program run the old version of HTBasic and execute the following statements:

```
LOAD "prog_name"
SAVE "temp_name"
QUIT
```

Then run the new release of HTBasic and execute the following statements:

```
GET"temp_name"
       Now press the STEP key (Alt-F1)
RE-STORE"prog_name"
PURGE"temp_name"
```

You should, of course, substitute the actual filename for *prog_name* and some unused filename, like "TEMP.BAS" for *temp_name*. If the file is not in the current directory, remember to include the full pathname.

Chapter 10
Changes From Earlier Releases

The following sections document the differences between the current release and earlier releases. "Changes From 5.0 to 6.0," "Changes From 4.0 to 5.0," "Changes From 3.0 to 4.0" and "Changes From 1.x/2.x to 3.0" are the main sections.

Changes From 5.0 to 6.0

If you are upgrading from 4.x to this release, read the following material to see what changes may affect you. Changes are categorized as New Features, Full Screen Editor, LIF Utilities, UNIX Versions, Numeric Compiler, HP BASIC Files, and Windows Version.

HTBasic for Windows no longer requires the use of a hardware key, it now uses a software solution for protection against unauthorized reproduction. There is a serial number provided in the packaging that is required to be input during installation. Proper installation requires the correct serial number.

Other changes to Version 6.0 include enhancements to the SERIAL32 device driver, including:

- Support for STATUS register 5 which is the Read hardware handshaking output line3
- Support for STATUS register 11 which is the Modem Status line.
- Increased Baud rates up to 115200 bps.

Changes From 4.0 to 5.0

Many new features have been added to HTBasic release 5.0. The most significant are listed here. Note that some features, by nature, are specific to a particular version of HTBasic. The new features are:

- Program line numbers increased to 4,194,303
- Run DOS Version under Windows
- New Full Screen Editor
- SUB Mode shows only SUB/DEF lines in editor
- Confirmation required before program changes are discarded
- Support for screen width larger then 80 columns
- Read and write HP BASIC/DOS and HP BASIC/UX files directly
- CONFIGURE CREATE specifies file header type (HTBasic, HP LIF)
- CD synonym for MSI
- HP BASIC version supported changed from 6.0 to 6.2
- SEPARATE with 256 color DOS/UNIX drivers supported
- CONFIGURE LONGFILENAMES
- New DOS Extender in DOS Version
- IOtech NB488 Driver
- DELSUB TO END speed up
- Larger recall buffer
- Screen dumps, begin/end row added for some drivers
- Support for longer files
- Digitize cursor color matches HP BASIC
- S3 and ATI display drivers
- PCMCIA IEEE-488 Card Drivers
- 16-Bit GPIO Card Driver
- Bug fixes

New Editor

HTBasic supports a new full screen editor that makes program development even easier. Unlike the old editor that locked the edit line to the center of the screen, the new editor allows editing of any visible line. Arrow keys work as expected and the cursor moves in a consistant column when moving up and down. Editing productivity is increased substantually on high resolution displays where slow scrolling speeds hampered editing.

The new editor, however, has been carefully implemented so that old ways of doing things still work. To discard your program line changes, you can still move up or down off the line. And you can still clear the line number from a line and type and execute a BASIC command, statement, or expression.

SUB Mode is a feature of the new editor that displays a list of the contexts (SUB, DEF, etc.) in the program and allows quick navigation among them. Press the SUB Mode key once to display and move through the list; press it again to switch back to regular editing.

The new editor keeps track of program changes and asks for confirmation before discarding them. For example, if you make some changes to your program, then get distracted, and then attempt to QUIT HTBasic, you will be alerted to the fact that the changes have not been saved yet. You then have an opportunity to save the changes or discard them.

The old editor limited program lines to two lines of 80 columns each. The new editor supports lines up to the Rocky Mountain Basic limit of 255 characters, regardless of the screen width. Furthermore, if your screen or window size is wider than 80 characters, the full screen width is utilized. Lines longer than the screen width are scrolled as necessary to allow editing of any part of the line.

LIF Utilities

The LIF diskette file transfer utilities have been improved in several ways. HPCOPY now supports LIF floppies with either 256- or 1024-byte sectors. Previously, it only supported 256-byte sectors. HPCOPY also now preserves the file date.

Now that HTBasic can recognize typed files with either HP LIF or HTBasic style file headers, HPCOPY has been upgraded to support both styles of headers. When copying files to a LIF floppy, the header style is automatically recognized and handled accordingly. When copying files from a LIF floppy, a command line switch gives you the choice of header styles.

Numeric Compiler

The Numeric Compiler has had significant improvements added in release 5.0. Previously, only integer and real variables were allowed in flow control statements. Now any data type, including complex and string, are allowed. The compiler's internal tables (for relocation items) were expanded to allow larger and more complex SUBs to compile. The PAUSE statement is also allowed in a compiled SUB, but operates the same as a STOP statement. Several bugs were fixed.

HP BASIC Files

HTBasic now recognizes HP LIF style file headers. This allows networked HP 9000 series 300 computer systems, PCs, SUN SPARCstations and HP 700 workstations to transparently read and write the same data files. To create network compatible typed files from HTBasic, use CONFIGURE CREATE "HP". The CONFIGURE CREATE statement specifies the header types created by the CREATE ASCII and CREATE BDAT statements. When copying typed files to LIF floppy with HPCOPY, HPCOPY automatically recognizes the header type and correctly copies the files to LIF floppy, even if the file resides on a Series 300 or Series 700 volume and was created by HP BASIC. HPCOPY also has a new command line switch to specify HP LIF headers rather than HTBasic headers when copying files from a LIF floppy. The resultant files are directly usable by HP BASIC.

Changes From 3.0 to 4.0

If you are upgrading from 3.x to this release, read the following material to see what changes may affect you. Changes are categorized as New HP BASIC 6.x features, New Features from HP BASIC/UX, New features Unique to HTBasic, Other Enhancements and Features Not Included.

New HP BASIC 6.x features

The following features from HP BASIC 6.x have been added to HTBasic, bringing HTBasic up to 6.2 compatibility:

• Additional error messages
• ASSIGN ... ;APPEND
• CALL string-containing-sub-name [WITH (...)]
• COPY ... ;PURGE
• CSUBs have been enhanced
• DUMP DEVICE IS ... ;APPEND
• DUMP DEVICE IS file
• INMEM(string-containing-sub-name)
• LINK ... ;PURGE
• PLOTTER IS ... ;APPEND
• PRINTALL IS ... ;APPEND
• PRINTALL IS file
• PRINTER IS ... ;APPEND
• READ KEY
• RUNLIGHT [ON|OFF]
• SCRATCH ALL, B, COM, RECALL
• SYSTEM$("WILDCARDS")
• WILDCARDS [DOS | OFF | UX;ESCAPE chr]

New Features from HP BASIC/UX

Another set of features from HP BASIC 6.x that has been added to HTBasic merges HP BASIC/UX statements into the main language definition. Where the BASIC/UX statements do not make sense in a non-UNIX environment, HTBasic, like HP BASIC 6.x, either ignores the statement or returns an error.

- Additional error messages
- ASSIGN pipe support
- CHGRP
- CHOWN
- CONTROL pipe support
- DUMP DEVICE IS file or pipe
- ENTER pipe support
- EXECUTE
- LINK
- OUTPUT pipe support
- PERMIT
- PLOTTER IS pipe support
- PRINTALL IS file or pipe
- PRINTER IS pipe support
- QUIT
- SET TIME with no value specified
- SET TIMEDATE with no value specified
- STATUS pipe support
- SYSTEM$("PROCESS ID")
- SYSTEM$("VERSION:OS")
- SYSTEM$("WINDOW SYSTEM")
- TIMEZONE IS
- -workspace switch specifies memory allocation
- -title switch specifies window title

New features Unique to HTBasic

A large number of new features have been added to HTBasic that are not present in Hewlett-Packard versions of RMB.

HTBasic now supports loadable graphic drivers in addition to loadable I/O drivers. Up to 10 graphic drivers and up to 16 I/O drivers may be loaded. A number of new graphic drivers are now available (not every driver is available in every version of HTBasic):

- PCX File DUMP driver
- GIF File DUMP driver
- DXF AutoCAD interchange file PLOTTER driver
- HPGL2 PLOTTER (and printer) driver
- Super VGA 256-color CRT driver
- VGAB CRT driver for 2-byte Versions of DOS like Japanese DOS/V
- MGA 640x400 CRT driver for laptops with Super-CGA capability
- Canon laser printer DUMP driver
- Color Desk Jet printer DUMP driver
- Epson 24-pin printer DUMP driver
- PostScript printer PLOTTER driver (for vector output)
- PostScript printer DUMP driver (for bit-mapped screen dumps)
- PCL Enhanced HP-PCL driver

The PLOTTER IS statement has been enhanced to support printers with built-in HPGL support, like the LaserJet III. The statement allows specification of p-points when output is directed to an interface select code. HP BASIC does not allow the p-point, ISC combination.

The HPGL driver has been upgraded to support polygons. This distinction is important when importing files into CAD programs so that polygons are imported as polygons rather than separate lines. For compatibility with older plotters, this capability is off by default. Use GESCAPE 104 to enable it.

Three binary modes have been added to the TABLET driver, allowing support of a wider range of tablets, as well as speeding up tablet operation on many of the tablets previously supported.

HTBasic improves on two of the HP BASIC 6.0 additions. The CALL statement and the new INMEM function allowed the name of a SUB to be specified in a string expression. HTBasic expands this list to include the LOADSUB, DELSUB and XREF statements:

- LOADSUB string-containing-sub-name
- DELSUB string-containing-sub-name
- XREF [SUB] string-containing-sub-name

This allows program constructs such as

```
10  IF Case=1 THEN
20    M$="Real"
30  ELSE
40    M$="Complex"
50  END IF
60  IF NOT INMEM(M$) THEN LOADSUB M$ FROM Lib$
70  CALL M$ WITH(X,Y,Z)
80  DELSUB M$
90  END
```

Several new GESCAPE operations have been added and the GLOAD and GSTORE statements have been enhanced. GESCAPE 102 allows the current VIEWPORT and WINDOW values to be read. GESCAPE 103 returns the current PEN and AREA PEN values. GESCAPE 104 allows device specific graphic extensions. For example, the HPGL driver uses this GESCAPE operation to enable HPGL/2 features.

The GLOAD and GSTORE statements have been enhanced so that any rectangular portion of the screen can be specified in addition to full screen images. The syntax closely resembles the HP BLOAD and BSTORE subprograms, but no LOADSUB is required since the functionality is an extension to the GLOAD and GSTORE statements.

HTBasic has been ported to the Sun Workstation and the Hewlett-Packard Series 700 Workstations. These represent the first RISC ports of the Rocky Mountain BASIC language and the first UNIX ports of HTBasic.

In the UNIX environment, as well as the DOS environment, it is helpful for a program to be able to know the command line used to start the program and to read environment variables from the operating system's environment. The COMMAND$ and ENVIRON$ functions have been added to HTBasic to do these operations.

ON and OFF options have been added to the CONFIGURE MSI statement to disable or enable CONFIGURE MSI translations. This is useful when a UNIX filename contains a colon.

CONTROL and STATUS have been enhanced to allow a single register to accept an array. This feature is used by several HTBasic device drivers. For example, the data acquisition drivers communicate channel gain lists using this feature.

Window versions of HTBasic support dynamic window resizing.

Other Enhancements

Online HELP has been enhanced to allow a secondary keyword. For example, HELP ON ERROR.

Speed improvements have been made in several areas, including:

- CALL times for programs with many SUBs.
- MAT REORDER.
- ENTER/OUTPUT involving implicit byte order correction.
- DISP statement.
- ASCII file operations.

HPCOPY has been enhanced to support BASIC 6.0 filenames, Series 500 BDAT files and to perform LIF ASCII to DOS ASCII translations.

The cursor size now indicates the insert/replace mode. Previously, the different cursor size was available only when using the CRTA driver with ALPHA ON. End-of-file error handling has been made more compatible. And SYSTEM$("VERSION:bin-name") is now supported.

Changes From 2.0 to 3.0

If you are upgrading from 1.x/2.x to this release, read the following material to see what changes may affect you.

The names used to start the DOS Version of HTBasic have been changed. If you have a 80387 math coprocessor or a 486 processor, use "HTB386". If you do not have a 387, use "HTBNO387".

The number of screen lines supported by a VGA adapter using the CRTB mode has increased from 25 to 30. (If you are using a VGA in CRTA mode, the increase only has effect when GRAPHICS is ON.) If you wish to continue using 25 lines rather than 30, read the information in Chapter 3, "CRT and Graphic Device Drivers" titled "Choosing a Text Font."

PLOTTER IS Change

Previously, HTBasic supported only one screen driver, INTERNAL, which contained support for CGA, EGA, VGA and HGC. The driver would automatically detect which was present, but an option could be used in the PLOTTER IS statement to override the choice. Now that HTBasic supports loadable device drivers, the method to override the default has changed. To override, you must manually specify which driver to use. See Chapter 3, "Loading CRT and Graphic Drivers."

Power-PROG Format

Many of the extensions added since 2.x are possible because of the new "Power-PROG Format". Power-PROG format is different from the 1.x/2.x PROG format but has the advantage of being smaller in most cases, faster in many cases, speedier during prerun and extendible. This internal method of representing programs is the result of incrementally compiling your programs: all the normal steps of compilation are made except native CPU code generation. Power-PROG format also allows expandability.

It also allows device drivers to be separate and loadable. This means TransEra or 3rd parties can develop future drivers for high-performance graphic cards, data acquisition cards, etc. in addition to the IEEE-488 and serial drivers supplied with this release.

To upgrade your existing PROG files, use the old version of HTBasic to LOAD and SAVE them into temporary files. Next, use the new version of HTBasic to GET and STORE them into a new directory. To help when moving a large number of programs to a new version of HTBasic, the HTBasic Developers Toolkit contains a utility program that converts old PROG format files to SAVE format files.

CONFIGURE

The CONFIGURE DEVICE syntax is no longer supported. To change an ISC, specify the desired ISC when the driver is loaded with LOAD BIN. Since the parallel printer port drivers are built-in, you must use CONTROL isc,101 to change the parallel port ISC.

Additions and Enhancements

Many additions have been made to HTBasic. The following features have been added: ACSH, ALTERNATEAUTOST, ARG, ASNH, ATN2 (angle), ATNH, BINEQV, BINIMP, CINT, CMPLX, COMMAND$, COMPLEX, COMPLEX numbers, CONFIGURE DIM, CONFIGURE KBD, CONFIGURE LABEL, CONFIGURE SAVE ASCII, CONJG, CONTROL CRT,100, COSH, EDIT FNName, EDIT SUB Subname, editor move left/right by word, ENVIRON$, FIX, FRE, hex constants (&H1FFF), HTBFULL, IMAG, INP, INPW, International language support, keyboard auto-repeat rate, loadable device drivers, MAT ABS, MAT ARG, MAT CMPLX, MAT CONJG, MAT IMAG, MAT REAL, multiple IEEE-488 board support, multiple serial port support, octal constants (&O00377), OUT, OUTW, REAL, Run-time Version, SINH, STATUS(), Student Version, SYSTEM$("DISP LINE"), TANH, and two monitor support.

The following features have been enhanced: * + - / ^ = <>, ABS, ALLOCATE, AREA COLOR, AREA INTENSITY, ATN, COM, COS, DATA, DEF FN, DISP, ENTER, EXP, GET, IMAGE, INPUT, INTEGER, LEXICAL ORDER IS, LGT, LOCK, LOG, MAT REORDER, MAT SEARCH, MAT, ON INTR for serial ports OUTPUT, PRINT, PROG format RE-SAVE, READ, REAL, SAVE, SIN, SQRT, SUB, Super-VGA graphic resolution, TAN, UNLOCK, several additional IEEE-488 boards are supported, and 30 text lines on VGA.

Diferences with RMB

The section lists the most common differences between HTBasic for Windows and HP BASIC series 200/300/700 (RMB).

To transfer BASIC programs between HTBasic and RMB use SAVE/GET. Prog files cannot be directly shared between HTBWIN & RMB.

Use SYSTEM$("VERSION:HTB") to programmatically determine if executing on HTBasic for Windows.

When in the editor, results from directly executed commands are displayed in the message line at the bottom of the screen.

Operations which cause an error may not produce exactly the same error number as RMB. For example, if an out-of-range value is passed to an INTEGER parameter, one platform might report error 19 (value out of range) and the other platform might report error 20 (integer overflow).

The PRT printer address constant returns 10 (the ISC of the Windows Print Manager Driver) instead of 701. Use CONFIGURE PRT TO 701 to change.

Refer to either online help "keyboard functions" or the "Installing and Using Guide" chapter 2 for the mapping of keyboard functions. You can use CONFIGURE KEY to remap any key to the function you wish.

RMB uses the Roman-8 character set; HTBWIN uses the Latin-1 character set. Therefore, characters below CHR$(32) and above CHR$(128) display differently. Refer to the User's Guide Chapter 13 "International Language Support." Also see the New Font Support description in the New Features section of this document.

RMB and HTBWIN have different actions for negative pen numbers. Depending upon the device and drawing mode, RMB negative pens either erase or complement the corresponding color. In HTBWIN, a negative pen complements the color map index, which is not the same as complementing the color.

HTBWIN restarts a line-type pattern with each line segment drawn. RMB tries to continue the line-type pattern across sequential line segments.

The destination of an ON...CALL statement can be deleted in HTBWIN. In RMB, attempts to delete such SUBs cause an error.

When doing a CAT TO S$(*), the first array element has the text "DIRECTORY:" before the actual directory name.

I/O Differences

ON INTR is not supported for RS-232 serial interfaces.

Some error handling and EOF handling in the ENTER statement is different in HTBasic for Windows and HP BASIC/WS.

The GPIB and HP-IB interface drivers in HTBWIN do not use DMA mode.

Mass Storage Differences

Filename wildcards are only available through the CAT statement.

INITIALIZE is not supported, and therefore, HTBasic for Windows cannot create RAM volumes. For RAM volumes, use a DOS RAM disk program instead (e.g., VDISK.SYS, RAMDISK.SYS).

There can be differences in file formats and byte ordering within files. See the ASSIGN statement in the "Reference Manual" for more information.

MSUS Format

```
RMB format: [directory path][filename][:msus]
```

```
HTBasic for Windows format: [drive:][directory path][filename]
```

You can map the RMB format to the HTBWIN model using the CONFIGURE MSI statement.

RMB CSUB

The following RMB CSUB utilities are not provided:

PHYREC: no substitute

GDUMPC: automatically provided by DUMP GRAPHICS if current Windows printer supports color.

BPLOT: Bload() and Bstore() functionality now provided through extensions to the GLOAD and GSTORE statements.

Performance Tuning:

HTBWIN provides control register "KBD,202" that allows you to customize performance. The tradeoffs of increased HTBWIN performance are decreased Windows response and decreased performance in other simultaneously executing Windows applications. The decreased Windows response is most noticeable as delayed response to mouse and keyboard input in all applications, including HTBWIN.

The syntax for performance tuning is:

```
CONTROL KBD,202;Value
```

...where Value is in the range of 0 to 32767, inclusive (default = 32)

The performance gain with increasing Value is non-linear, and most improvement occurs in the bottom 10% of the parameter's range.