

---

# TransEra HTBasic For Windows

## Porting Considerations

### Porting HP BASIC/9000 Programs to HTBasic for Windows

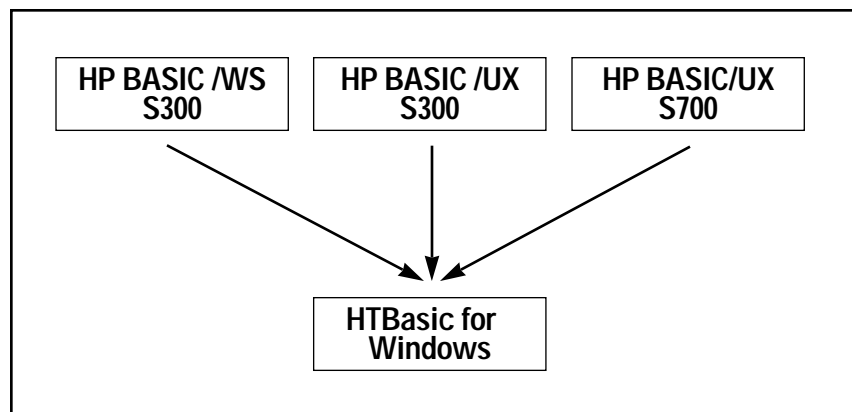
One of the hallmarks of RMB (Rocky Mountain Basic) has been the ability to develop applications which will run on a variety of computer platforms.

HTBasic for Windows adds the PC platform to the RMB family. This product note describes differences between HP BASIC/9000 (including Series 200, 300, 400, and 700 computers) and HTBasic for Windows. Differences in hardware, file system, and keywords are discussed to help you move programs to HTBasic for Windows from other HP BASIC platforms.

The CONFIGURE command allows you to customize HTBasic for Windows to more nearly match the configuration of an HP 9000 computer. This command is described completely in the documentation supplied with the product.

### Hardware Differences

HTBasic for Windows compensates automatically for many of the hardware differences that exist between an HP BASIC/9000 and other hardware platforms. The following sections outline many of the differences and explain what limitations still apply.



### Disk Drives

GP-IB disk drives are not supported. No mass storage device is supported across the GP-IB. HP LIF disks are not compatible with DOS disk formats and cannot be read or written directly by HTBasic for Windows. However, there are LIF utilities provided with HTBasic for Windows for transferring files from LIF disks. These utilities work with most PC floppy drive mechanisms.

### Softkey Layout

Different keyboard layouts are used by HP BASIC/9000 and PCs. Some PC keyboards have the softkeys down the left side of the keyboard, and some have them across the top of the keyboard. To make up for these differences and the lack of spatial coherence between the physical keys and the softkey menu, HTBasic for Windows menu labels

have been numbered to provide numerical coherence. The label numbered "1" always corresponds to the softkey with the 1 printed on it. The number is not meant to be the softkey number.

To provide compatibility with programs written for the 9836 Nimitz keyboard (which starts with softkey K0), the command KBD CMODE can be used to turn on or off compatibility mode.

### A New Backspace

Under HTBasic, the BACKSPACE key was assigned to CHR\$(255)&"B". This function does the same thing that the LEFT arrow key does. HP BASIC is unique in this treatment of the BACKSPACE key. For compatibility with the remainder of the computing community, HTBasic for Windows assigns a new function

---

to the BACKSPACE key. Pushing BACKSPACE deletes the character to the left of the cursor. This function is named DEL LEFT and is equivalent to the function LEFT followed by DEL CHR. If you do not like this definition, you may be able to redefine the key using the CONFIGURE KEY statement.

### **Series 300 Bit-Mapped (CRTB) Displays**

With an HP 9816, 9836, or other HP computer with separate alpha and graphics hardware, some of the differences you find in HTBasic for Windows will be the same differences you would find moving to an HP 310 or another HP computer with a fully bit-mapped alpha/graphics display. On bit-mapped displays, the ALPHA image is written into one or more of the graphic planes. On HP BASIC/9000 workstations, one plane can be separated from the others for use solely for the ALPHA image. Or, all the planes can be merged for shared use between ALPHA and GRAPHICS. HTBasic for Windows supports both separated and merged mode.

When ALPHA and GRAPHICS are merged, ALPHA text is converted to graphic bits and written into the graphic planes, overwriting any graphics data that might be present. Therefore, ALPHA and GRAPHICS cannot be dumped separately. And when the ALPHA text is scrolled, any graphic data present will be scrolled also.

CRTB Mode is the default for Windows system drivers. This mode gives the best HP 9000 emulation.

### **9836C (CRTA) Displays**

The HP 9836C display is driven by CRTA Mode drivers. A CRTA display has distinct ALPHA and GRAPHICS

images. Either the ALPHA or GRAPHICS images can be displayed, or both can be displayed at once overlapping each other. The hardware for the two images is independent. HTBasic for Windows emulates CRTA Mode drivers.

### **Processor**

The instruction sets for the Intel x86 family of processors, Motorola 680x0 processors, and HP Precision Architecture processors are all different. Since a CSUB contains processor code, a CSUB cannot be moved from HP BASIC/9000 or HTBasic for Windows on one processor and executed under another. The same holds true for assembly routines called with WRITEIO 9827.

### **Speed**

If speed is critical to your application, these suggestions may be helpful: Buy a processor with math instructions built-in (such as the 486, 486DX, or Pentium), or to add a math coprocessor. Pentium systems can meet the performance needs of most applications.

### **Clock**

Millisecond timing resolution is not available on the IBM PC. The PC clock runs at about 18.2 ticks a second. Thus, all timing functions within HTBasic for Windows have a best resolution of about 55 milliseconds.

### **GP-IB/IEEE-488**

The GP-IB bus is also known by the name IEEE-488 or GPIB bus. Most computers are not sold with an integral IEEE-488 interface. Contact TransEra for information on supported interfaces for the PC and other platforms running HTBasic for Windows.

Some interfaces lack minor functionality. For example, PC IEEE-488 interface cards that use the NEC 7210 IEEE-488 controller chip are incompatible in the following areas:

- All READIO and WRITEIO registers are different
- Bit 0, IFC, is not supported in STATUS register 5 or the ENABLE INTR mask.
- The REM and LOC bits of STATUS register 6 are not supported as well as the DAV, NDAC, NRFD, and IFC bits of STATUS register 7.

### **Character Set**

The default character set used by the IBM PC is different than the HP Roman-8 character set used by HP BASIC/9000. The differences exist in characters CHR\$(128) and above. In HTBasic for Windows, all characters from CHR\$(128) through CHR\$(254) are allowed in variable names. If running a program with HP BASIC/9000 and HTBasic for Windows, you should restrict variable names to characters from CHR\$(161) to CHR\$(254) and the legal characters less than CHR\$(128).

### **Shared Resource Manager (SRM)**

HTBasic for Windows can be a classical SRM client on an SRM server. Also, with SRM/UX servers, DOS computers can access the files on the server using NFS. HTBasic for Windows is also compatible with most industry standard networks such as NetWare, LAN Manager, and Windows NT.

### **Miscellaneous Differences**

The serial interface is enhanced with XON/XOFF flow control, and is enabled by default. If you are porting existing programs that transfer binary data or ^S and ^Q characters as part of the data, you should turn

---

off XON/XOFF flow control and turn on hardware handshaking with the statements:

```
CONTROL 9,100;0
CONTROL 9,5;0
CONTROL 9,12;0
```

The maximum line number in HTBasic for Windows is 4,194,304. If you plan on also running a program on an HP BASIC/9000 series 200/300 computer, you should not use line numbers over 32,766.

The maximum length of a line, in general, is the same as HP BASIC/9000. However, in rare instances (such as a program with hundreds of variable names and a program line with little used operands and thirty or more variable references), it is possible that a line of legal length in HP BASIC/9000 will be considered too long in HTBasic for Windows. Such a line will generate an error when entered, however, the error is extremely rare. To correct the error, divide the statement into two statements.

### ASCII Files

When entering data from files, data should be entered into variables of the same type as those used to output it. This general rule applies to HTBasic for Windows as well, and if violated can produce unexpected results. Consider this program:

```
10      CREATE "TEMP",1
20      ASSIGN @I TO
        "TEMP";FORMAT ON
30      OUTPUT @I,"1,2"
40      OUTPUT @I,"3,4"
50      RESET @I
60      ENTER @I;A,B
70      PRINT A,B
80      ASSIGN @I TO *
90      PURGE "TEMP"
100     END
```

The program violates the matching-type rule by outputting strings and then entering numerics. Intuitively, you expect line 70 to print 1 and 2, which it does. When line 10 is changed to CREATE "BDAT", the same result is produced. But if line 10 is changed to CREATE "ASCII", then HP BASIC/9000 prints 1 and 3. This discrepancy has been corrected in HTBasic for Windows. If you have written programs that use ASCII files and violate the matching-type rule, you should correct them before running them with HTBasic for Windows. If your program must run with both HP BASIC/9000 and HTBasic for Windows, you must either adhere to the matching-type rule, or use another file type.

### File Systems

A primary difference between HP BASIC/9000 and HTBasic for Windows is that HP BASIC/9000 does its own file I/O, while HTBasic for Windows has the operating system do all file I/O. This has advantages and disadvantages. Any hard disk, diskette, network, or device accessible through the file system is accessible from HTBasic for Windows. HP LIF capabilities are not available in MS-DOS so you cannot use HP LIF diskettes, GP-IB disks, or tape drives, from HTBasic for Windows.

A couple of warnings are in order about the way most operating systems work. If a file is currently ASSIGNED, the operating system buffers some data in memory to make I/O faster. This buffering can produce unexpected results if the same file has multiple I/O paths ASSIGNED to it concurrently. Also, you should not remove a diskette or turn the power off while a file is ASSIGNED.

HP BASIC/9000 and HTBasic for Windows maintain compatibility with LIF file types by keeping a special file header for typed files (BDAT, LIF ASCII, and PROG files). The extra information that must be stored for these file types is kept in the header. The header is kept totally hidden from BASIC programs. However, to programs outside HTBasic for Windows, the header will be accessible as the first bytes of the file. Any files without the special header are known as "ordinary files". In a CAT listing, the file type is blank for ordinary files or gives the operating system.

In addition to HTBasic for Windows file headers, HTBasic for Windows can recognize and use HP LIF headers. This allows networked computers to directly interchange data between HP computer systems running HP BASIC/9000 and systems running HTBasic for Windows. The CONFIGURE CREATE {"HP" | "HTB"} statement allows the program to specify the type of file header to use when creating a new BDAT or LIF ASCII file.

### LIF ASCII vs. Ordinary ASCII

Early versions of HP BASIC/9000 did not have a file type that matched DOS ASCII or UNIX ASCII files. Starting with BASIC 5.0, it does have an ordinary file type that can hold DOS ASCII or UNIX ASCII data. With HP BASIC/WS or HP BASIC/UX, ordinary files are called "HP-UX" files. With the Measurement Coprocessor Card, ordinary files are called "DOS" files. No special header or other embedded information is placed in ordinary files. An ordinary file written with FORMAT ON is a DOS ASCII file. An ordinary file written with FORMAT ON, EOL CHR\$(10) is a UNIX ASCII file. Use the CREATE statement to create an

---

ordinary file. In a CAT listing, an ordinary file is listed with the file type blank, or gives the name of the operating system.

### DOS (FAT) File System

The following sections describe some of the differences between the DOS (FAT) file system and the HP BASIC/9000 Logical Interchange Format (LIF) file system. This discussion is not meant to be a substitute for your DOS manual. You should read it for complete information on the topics presented here.

The FAT file system, like the Shared Resource Manager (SRM) file system, is a Hierarchical File System (HFS). An understanding of HFS, UNIX, or SRM file systems may aid you in understanding the FAT file system.

### DOS (FAT) File Specifier

Under DOS, a file specifier has the form:

[d:] [directory path] filename[.ext]

where:

**d:** is the drive letter. Usually this is A: and B: for the two diskette drives, and C: for the hard disk. Higher drive letters, D:, E:, and so on, usually refer to RAM (MEMORY) disks, CD-ROMs, or network drives. This part of the file specifier corresponds to the msus of the HP file system, but is included on the front, not the end of the file specifier. It is optional, and if omitted, the current (or default) drive is used.

**directory path** is optional. It is explained later in this section.

**filename** is the main part of the file specifier. It may be one to 256 characters long. These characters

are not allowed to be in a filename: `."^[]:|<>+=,;`, and control characters whose ASCII value is less than the space character. All other characters are legal. If lowercase letters are used in a filename they are converted to uppercase by DOS. Thus, you cannot have a file named "AbC" and another file named "aBC". DOS will consider both names to be "ABC". Be aware that a DOS file specifier is often referred to simply as a filename.

**ext** is the filename extension. The same characters that may be used in a filename may also be used in an extension. Certain conventions are used for filename extensions. Most of them are arbitrary, with a few exceptions noted below. Some conventional extensions are:

Extension	Conventional Use
.BAS	BASIC program
.BDT	BDAT file
.AS	ASCII file
.DAT	BDAT or DOS data files
.TXT	Text file
.LIS	Text files or compiler listings
.LST	Another common listing file extension
.PRG	BASIC program stored in prog type using STORE
.BAT	DOS batch files MUST have this extension
.COM	An executable DOS command or program
.EXE	An executable DOS command or program. DOS commands and programs MUST have one of these two extensions

Note that compared to an HP file specifier, the msus is replaced by a drive letter and moved from the back to the front. Also, an extension has been added and no password is used or available. While DOS does not have passwords, it does have some access capabilities (similar to SRM access capabilities). These are discussed below. If passwords are present they generate Error 462 - Invalid Password.

If an HP style msus is present, it can be translated to a DOS path if specified by the CONFIGURE MSI statement. If no matching translation can be found, an error is returned. The CONFIGURE MSI statement has been added to HTBasic for Windows to allow HP style file specifiers to be used with a DOS file system. If DOS style file specifiers are used, then the CONFIGURE MSI statement is not needed.

### DOS Directory Path

The DOS file system is "tree" structured, almost exactly like the HFS and SRM file systems. The "/" character of the HFS file system is replaced with "\" in the DOS file system. Passwords do not exist in the DOS file system and will cause an error if included in a directory name.

The file system is organized as a tree. Actually, it is usually thought of as an upside-down tree. The top of the tree is, thus, the root. (Not roots. Directory trees are considered to have only one root, and the term trunk is usually not used.) The tree is composed of directories and files. Each directory may contain files and additional directories which act like branches down the tree. Directory names follow the same rules as filenames with extensions.

A directory path is the path you climb through in the tree to get from the root of the tree to a certain directory. It consists of the names of each directory that must be climbed through in order to get to that certain directory. The directory names are separated with the backslash "\" character.

---

## DOS File Types and Access Capabilities

The DOS file system stores certain information about a file in addition to storing information in the file. This information consists of:

- file name
- number of bytes in the file
- modification date and time
- access capabilities: hidden, read-only, system
- location on the disk of the file contents

DOS does not store a record length or a file type. It does have access capabilities, although they are different from the SRM access capabilities. The access capabilities are called "attributes" by DOS and may be changed by the DOS ATTRIB command. They may also be changed by the HTBasic for Windows PROTECT command. The meanings of the attributes are:

Attribute	Meaning
hidden	The filename is not shown in a disk catalog although the file is there
read-only	The file may be read but not written to or deleted
system	The file is a system file

## DOS Devices

HP BASIC/9000 accesses devices through interface and device select codes. The same is true of HTBasic for Windows. However, DOS also allows access to devices as if they were files. Special names are given to devices by DOS, and when used in the place of a filename, access a device instead of a file. These names are called DOS Device Names and are typically: CON, AUX, COM1, COM2, PRN, LPT1, LPT2, and NUL. Often, data acquisition hardware

for the PC comes with a DOS device driver. To access such a data acquisition board, treat it as you would a file and use ASSIGN, OUTPUT, and ENTER.

## DOS Wildcards

The question mark "?" and the asterisk "\*" have special meaning to DOS. They are called wildcards and are used in DOS commands like DIR and the HTBasic for Windows CAT command in order to select more than one file. A filename with wildcard characters in it will be compared with existing filenames using special rules and all filenames that "match" will be selected. These are the rules used to match an actual filename with wildcards:

The "?" character will match any one character in the same position of an actual filename. For example, the string "?AT" will match the strings "CAT", "BAT", "MAT", or any other string three letters long that has an "A" as the second letter, and "T" as the third letter.

The "\*" character will match zero or more characters starting at that position. For example, "\*" will match all filenames. "\*.BAS" will match all filenames that have the ".BAS" extension.

## Keyword Differences

The following sections present, by keyword, various implementation differences between HTBasic for Windows and HP BASIC/9000. Most differences are the result of hardware differences, file system differences, or extensions in HTBasic for Windows. In some cases, DOS or UNIX commands that relate to BASIC commands are given.

Some differences exist because of enhancements HTBasic for Windows. Many enhancements can be included in programs that run under both HP BASIC/9000 and HTBasic for Windows. One approach is to hide statements that will not parse under HP BASIC/9000 in OUTPUT KBD statements:

```
OUTPUT KBD;"CONFIGURE PRT TO
701"&CHR$(255)&"E";
```

Another approach is to set up a section of code that is executed only by HTBasic for Windows. When you GET the program under HP BASIC/9000, the lines with HTBasic for Windows syntax enhancements will be commented out and ignored by HP BASIC/9000:

```
10      INTEGER Htbasic
20      Htbasic=SYSTEM$(
"VERSION:HTB")<>"0"
30      IF Htbasic THEN
40      CONFIGURE PRT TO 701
50      END IF
60      END
```

When porting programs between HP BASIC/9000 and HTBasic for Windows, note the differences for the following keywords.

## ALLOCATE

Under HTBasic for Windows, GOSUB and ALLOCATE use the same stack. Intermixing these statements can cause changes in available memory that are different from HP BASIC/9000.

## ASSIGN

When an ASSIGN fails, the previous state of the I/O path is not preserved. Also, the CONVERT and PARITY options are not implemented.

Under Windows, if changes are made to an ASSIGNed file, the directory entry is not updated until the file is

---

closed. Windows buffers reads and writes to disk. You should not remove a diskette or turn the power off while a file is ASSIGNED. Exchanging diskettes while a file is ASSIGNED on the first can destroy the next diskette. Two I/O paths ASSIGNED simultaneously to the same file can produce slightly different results than HP BASIC/9000, depending on the buffering Windows does.

Under HTBasic for Windows ASSIGN includes two new options, FORMAT LSB FIRST and FORMAT MSB FIRST, to specify byte ordering of binary numeric data transfers. This provides the ability to do binary transfers with any device or computer, regardless of the byte ordering that device uses.

#### **ATN2**

ATN2 is an HTBasic for Windows function that returns the angle to a point.

#### **BEEP**

The BEEP statement generates a tone with the PC speaker. Both the frequency and the duration are adjustable.

#### **BINEQV/BINIMP**

These commands are HTBasic for Windows functions that perform a bit by bit equivalence operation.

#### **BLOAD/BSTORE**

The functionality of the HP BLOAD, BSTORE compiled subroutines is an integral part of the HTBasic for Windows language and does not require CSUBs. (See the GLOAD and GSTORE section.)

#### **BUFFER**

In HTBasic for Windows, it is usually incorrect to access numeric data in a buffer through the array name. ENTER and OUTPUT should be used instead.

#### **CAT**

HP BASIC/9000 uses wildcards as a primary filter and the SELECT option as a secondary filter in choosing file-names to display. HTBasic for Windows is designed to be used with one or the other. Use wildcards or the SELECT option, but not both.

#### **CDIAL**

CDIAL is not supported.

#### **CHECKREAD**

CHECKREAD is not supported.

#### **CINT**

CINT is an HTBasic for Windows function that converts a value to INTEGER.

#### **COMMANDS**

COMMANDS is an HTBasic for Windows function that returns a copy of the command line.

#### **CONFIGURE**

The CONFIGURE statement is an enhancement to HTBasic for Windows that allows the environment to be customized to a user's preference, or to match the user's HP hardware.

#### **CONTROL and STATUS**

Depending on the hardware interface, some CONTROL and STATUS registers may be different.

In some instances HTBasic for Windows can pass arrays to and from a single register. This capability is used for things like gain control lists in data acquisition drivers.

#### **COPY**

HTBasic for Windows does not support the copy of a full disk to another disk. Use the operating system for full disk copies. Under Windows you can use either the File Manager or the DOS "DISKCOPY" or "XCOPY" commands. The DOS XCOPY command is an extended COPY command that, among other things, allows you to copy entire disks when the disk sizes are not the same. The DOS COPY command is used to copy individual files. If wildcards are included in the command, then several files can be copied with a single command.

#### **CREATE**

Because Windows supports extendible files, the number of records specified in the CREATE statement is ignored. An invalid number does not generate an error as it will under HP BASIC/9000. Programs that depend on errors occurring by writing past the last specified record will not function correctly, as HTBasic for Windows will simply extend the file as needed. Programs that depend on the pre-allocation of the requested records should write dummy data at the time the file is CREATED. Under Windows it is sufficient to write data in the last record.

Do not confuse a LIF ASCII file, created with CREATE ASCII, with a DOS ASCII or UNIX ASCII file, created with CREATE.

Use the CONFIGURE BDAT MSB FIRST statement before creating BDAT files that will be moved to HP BASIC/9000.

---

## CREATE DIR

This command is exactly like the HP BASIC/9000 command of the same name. It is the equivalent of the DOS MD command.

## DEF FN

Nested I/O is not allowed under HP BASIC/9000. For example:

```
10      PRINT FNX
20      END
30      DEF FNX
40      PRINT "DEBUG:START"
50      RETURN 0
60      FNEND
```

will produce an error under HP BASIC/9000. Nested I/O does not return an error under HTBasic for Windows, but should not be used.

HTBasic for Windows limits the depth that recursion can occur. The depth is limited by the size of the processor stack, not the BASIC workspace size.

## DELSUB

HTBasic for Windows allows a string variable to specify the name of the subprogram or function to delete.

## DUMP DEVICE IS

By default, HTBasic for Windows uses the WIN-DUMP driver to provide support for any printer supported by Windows that supports bit maps. WIN-DUMP is only supported through select code 10. There are several other drivers for files and printer languages.

## EDIT

EDIT SUB and EDIT FN are extensions in HTBasic for Windows. Several new edit functions are also included. (See the OUTPUT KBD section.)

## ENABLE INTR

Depending on the hardware interface, some ENABLE INTR mask values may be different.

## ENVIRON

ENVIRON\$ is an HTBasic for Windows function that returns information from the operating system environment.

## ERRDS

ERRDS is not supported.

## ERRMS

HTBasic for Windows error messages are usually similar to those in HP BASIC/9000. Programs that depend on ERRMS\$ returning the exact same message as HP BASIC/9000 should be modified accordingly. In particular, where an HP BASIC/9000 error message has seemed less descriptive than it should be, HTBasic for Windows returns a more descriptive message.

## ERRN

Any error number of 2000 or greater is an HTBasic for Windows extension to HP BASIC/9000. Not all errors that can occur under HP BASIC/9000 can occur under HTBasic for Windows.

In general, and whenever possible, the error numbers returned for errors are the same as those returned by HP BASIC/9000. But in some instances the operating system or environment in which HTBasic for Windows runs makes it impossible or impractical to return the same number.

## EXECUTE

The EXECUTE statement has been added to run operating system commands or other programs while HTBasic for Windows is running.

## FIX

FIX is an HTBasic for Windows function that truncates a value to INTEGER.

## FRACT

HTBasic for Windows allows the FRACT of a complex value, returning the fractional part of the real part of the complex value. HP BASIC/9000 gives error 620.

## FRE

FRE is an HTBasic for Windows function that returns the amount of free memory.

## GESCAPE

Only HP BASIC/9000 operation selectors 1 to 6 are supported by GESCAPE. Operation selectors greater than 29 are enhancements to HTBasic for Windows.

## GET

HTBasic for Windows turns lines with syntax errors into comments by inserting "!" instead of just "!" after the line number. This allows FIND "!" to quickly identify lines needing corrections.

## GLOAD and GSTORE

HTBasic for Windows enhances GLOAD and GSTORE with the capabilities of the HP BASIC/9000 BLOAD and BSTORE CSUBs to load and store partial screen images (rectangular blocks). When performing a full screen GSTORE, HTBasic for Windows translates the image into the Windows .BMP format. If the resulting image array is output to a file, any application that recognizes the .BMP format can read, display, or modify that image. However, when performing partial screen GLOAD/GSTOREs, the format used is not sharable with other applications. In either case, the images are not compatible with those generated by HP BASIC/9000.

These subprograms can be used in place of the BLOAD and BSTORE CSUBs for users that want to continue calling BLOAD/BSTORE instead of switching to the new GLOAD/GSTORE syntax:

```

10      SUB Bstore(INTEGER
      Int_array(*),W,H,OPTIONA
      L Rule,REAL X,Y)
20      SELECT NPAR
30      CASE 3
40      GSTORE
      CRT,Int_array(*),W,H
50      CASE 4
60      GSTORE
      CRT,Int_array(*),W,H,Rule
70      CASE 5
80      WHERE X0,Y0
90      GSTORE
      CRT,Int_array(*),W,H,Rule,
      X,Y0
100     CASE 6
110     GSTORE
      CRT,Int_array(*),W,H,Rule,
      X,Y
120     END SELECT
130     SUBEND
140     SUB Bload(INTEGER
      Int_array(*),W,H,OPTIONA
      L Rule,REAL X,Y)
150     SELECT NPAR
160     CASE 3
170     GLOAD
      CRT,Int_array(*),W,H
180     CASE 4
190     GLOAD
      CRT,Int_array(*),W,H,Rule
200     CASE 5
210     WHERE X0,Y0
220     GLOAD
      CRT,Int_array(*),W,H,Rule,
      X,Y0
230     CASE 6
240     GLOAD
      CRT,Int_array(*),W,H,Rule,
      X,Y
250     END SELECT
260     SUBEND

```

Note that only rule 3, replace, is currently supported.

## GRAPHICS INPUT IS

Both HP BASIC/9000 and HTBasic for Windows do an implicit GRAPHICS INPUT IS assignment if you attempt to use graphic input statements before an explicit GRAPHICS INPUT IS. The difference is that HTBasic for Windows does the implicit GRAPHICS INPUT IS as soon as HTBasic for Windows is started, and HP BASIC/9000 waits until the first graphic input statement is executed. The only known effect of this different approach is that under HP BASIC/9000, a SYSTEM\$ ("GRAPHICS INPUT IS") returns "0" until the first graphic statement is executed and HTBasic for Windows returns the correct value anytime.

## HELP

The HELP statement is an HTBasic for Windows statement. It provides on-line help for all language statements. Just enter HELP followed by the keyword of interest and HTBasic for Windows brings up the Reference Manual page for the requested keyword. It is no longer necessary to get the Reference Manual when you have a question on statement syntax or functionality.

## HIL

HIL related statements are not supported.

## IMAGE

Entering data from a string using:

```
ENTER L$ USING "Y"
```

will always use the internal byte ordering of the computer. For PCs and compatibles, the byte ordering is LSB FIRST. For HP BASIC/9000, the byte ordering is MSB FIRST. This limitation applies to ENTER and OUTPUT with strings only. With devices, the byte ordering can be selected in the ASSIGN statement.

## INITIALIZE

HTBasic for Windows does not support INITIALIZE. To initialize a new LIF disk, use "INITIALIZE" on an HP BASIC/9000. To initialize a new DOS disk, use the "FORMAT" command under DOS. To initialize a new HP-UX floppy disk, use the "mediainit" and "newfs" commands under HP-UX.

RAM disks are not supported with the INITIALIZE":MEMORY,0" command. Many excellent RAM disk programs are available for the PC that make the RAM disk available to all DOS programs, including HTBasic for Windows. These programs can usually make RAM disks in conventional, expanded, or extended memory. A simple one is provided with DOS and is called VDISK.SYS or RAMDISK.SYS.

A RAM disk is accessed like any other DOS disk by specifying its drive letter (for example, E:).

## INP and INPW

INP and INPW are HTBasic for Windows functions for communicating with devices having no HTBasic for Windows device driver.

## KBD CMODE

HP BASIC/9000 and HTBasic for Windows both use KBD CMODE for Nimitz keyboard softkey compatibility. The Nimitz keyboard is used on the 9836 and has ten softkeys, the lowest of which is labeled k0. The softkey labels are displayed at the bottom of the screen in two rows, each row containing five labels and each label 14 characters wide.

The difference between HP BASIC/9000 and HTBasic for Windows implementation of KBD CMODE ON is that HTBasic for Windows exactly emulates the screen format for the labels, while HP BASIC/9000 uses an emulation that gives physical correspondence with the ITF 4-2-4 softkey layout.



---

## LEXICAL ORDER

Several extensions are present in the LEXICAL ORDER statement of HTBasic for Windows to allow user definition of upper/lowercase rules for languages that are not built-in. The rules can also be loaded from a file.

## LINE TYPE

LINE TYPE sets the style or dash pattern and repeat length of lines. This is, however, limited to available Windows line types. In the LINE TYPE statement, the repeat length is ignored by most graphic drivers.

## LINK

LINK is not supported.

## LIST BIN

LIST BIN lists the name and version number of each BIN currently in memory. (Programmable in HTBasic for Windows, but not in HP BASIC/9000.)

## LOAD

HP BASIC/9000 PROG files and HTBasic for Windows PROG files are not compatible. To move programs between the two environments, use ASCII program files.

## LOAD BIN

The LOAD BIN statement has been enhanced to allow software switches to be passed to device drivers. HP BASIC/9000 BIN files are not compatible with HTBasic for Windows.

## LOADSUB

HTBasic for Windows allows a string variable to specify the name of the subprogram or function to load.

## MASS STORAGE IS

The current "MASS STORAGE IS" includes not only the device, but also the current directory. In other words,

it specifies not only which tree (device) you are in, but where in the tree (current directory) you are. The DOS command CD changes the current directory when not running HTBasic for Windows.

## OUT and OUTW

OUT and OUTW are HTBasic for Windows statements for communicating with devices having no HTBasic for Windows device driver.

## OUTPUT KBD

Three editor functions have been added to HTBasic for Windows and should not be used in programs that will be executed with HP BASIC/9000: DEL LEFT, NEXT WORD, and PREV WORD. Otherwise, all the two-character function key sequences (i.e., CHR\$(255)&CHR\$(X) ) used by HP BASIC are compatible with HTBasic for Windows. If multiple statements are output in a single OUTPUT KBD statement, they are all executed before the next BASIC line. HP BASIC/9000 sometimes intermixes the execution with multiple BASIC lines, based on the presence or absence of "closure keys".

## PERMIT

PERMIT is used under UNIX to set the permissions (mode) of a file, directory, or device. Permissions specify who can read, write, or execute a file, and who can search a directory. To change file attributes under DOS, use the PROTECT statement.

## PHYREC, Phyread, Phywrite

The PHYREC utilities under HP BASIC/9000 allow physical disk sectors to be read or written. Such access usually provides enhancements or file utilities for LIF formatted disks. Programs using Phyread and Phywrite most likely will not work under Windows, and in fact will probably destroy the files on the disk involved. No PHYREC utilities are supplied with HTBasic.

## PLOTTER IS

Both HP BASIC/9000 and HTBasic for Windows do an implicit PLOTTER IS assignment if you attempt to use graphic statements before an explicit PLOTTER IS. The difference is that HTBasic for Windows does the implicit PLOTTER IS as soon as HTBasic for Windows is started, and HP BASIC/9000 waits until the first graphic statement is executed. The only known effect of the different approaches is that under HP BASIC/9000, a SYSTEMS ("PLOTTER IS") returns "0" until the first graphic statement is executed and HTBasic for Windows returns the correct value anytime.

HP BASIC/9000 supports only "INTERNAL" and "HPGL" graphic languages. HTBasic for Windows supports loadable graphic device drivers so it is not limited to these two choices. HTBasic for Windows also allows clip-limits to be specified when output is directed to a device, allowing use of plotters or printers that are incapable of returning p-points.

## PRINT

HTBasic for Windows has been extended to allow the displacement of the attribute and color control characters that normally have the values CHR\$(128) to CHR\$(143). Since the PC has characters in this range that sometimes need to be displayed, HTBasic for Windows has the capability of moving the range with the statement CONTROL CRT,100;1.

## PRINT LABEL and READ LABEL

PRINT LABEL and READ LABEL are used to set and read the volume label of a disk drive. HTBasic for Windows does not support PRINT LABEL; you must use the DOS "LABEL" command. To change the label of a disk from an HTBasic for Windows program, use the EXECUTE "LABEL x:" command.

---

## PROTECT

PROTECT is used to set LIF file passwords under HP BASIC/9000 and DOS file attributes under HTBasic for Windows. A special form of PROTECT is used by HTBasic for Windows to change file attributes. The syntax is:

PROTECT file-specifier, protect-code

where protect-code is a string containing zero or more of these characters:

Character	Meaning
(none)	no protection
R	read-only: File cannot be written or deleted
S	system file: For the most part this attribute has no meaning
H	hidden system: File will not be listed by a CAT command

If a character is not included, that attribute is cleared. If the string is blank, all attributes are cleared.

## PRT

Under HTBasic for Windows, PRT is set by default to ISC 10, the default Windows printer. Therefore, PRT is set to ISC 10 instead of 701. Programs with statements that use PRT explicitly, such as "PRINTER IS PRT" need not be changed if a parallel printer is used on the PC. To change PRT back to 701, for IEEE-488 printers at primary address 1, use the statement "CONFIGURE PRT TO 701". It may be convenient to include this statement in your AUTOST file. This statement is not necessary if you use the value 701 (or any other value) explicitly.

## PURGE

PURGE is similar to a combination of the DOS DELETE and RD or RMDIR commands. Unlike DELETE, PURGE will only delete one file at a time, and will also delete directories. PURGE will not delete a directory unless there are no files in that directory. Also, DOS will allow you to PURGE an ASSIGNED file if SHARE.EXE has not been loaded. The actual PURGE takes place after the file is closed.

## QUIT

The QUIT command exits HTBasic for Windows and returns you to Windows. Any program or data in memory is lost. You will be prompted to store any program changes before quitting.

## READIO and WRITEIO

READIO and WRITEIO access hardware registers directly, and therefore, if the interface hardware is different than the hardware on an HP BASIC/9000, the READIO and WRITEIO registers will not be compatible.

## RENAME

RENAME is used to change the name of a file or directory, or to move a file from one directory to another directory on the same disk. However, RENAME cannot move one directory to another.

## SAVE

The SAVE statement in HTBasic for Windows can be set to save programs in either ordinary ASCII or LIF ASCII. This is done with the CONFIGURE SAVE ASCII statement.

## SCRATCH BIN

SCRATCH BIN is not supported for HTBasic for Windows binaries. You must QUIT and restart HTBasic for Windows to scratch all binaries.

## SEPARATE ALPHA

SEPARATE ALPHA is supported.

## SET ALPHA/DISPLAY MASK

SET MASK is not supported.

## SET CHR

SET CHR is not supported.

## SET TIME/TIMEDATE

HP BASIC/UX keeps a BASIC time that is separate from the actual system time. SET TIME and SET TIMEDATE, specified without any time value, resynchronizes the two. However, in HTBasic for Windows, SET TIME and SET TIMEDATE with no time value, is ignored.

## SOUND

SOUND is not supported.

## STATUS

STATUS @Iopath,2 always returns a 4.

STATUS @File,3 returns the current length, not the CREATE length. This is because files are extendible under DOS and UNIX.

The STATUS() function (as opposed to the STATUS statement) is an addition to HTBasic for Windows. Any STATUS or CONTROL registers greater than 99 are also additions.

As in HP BASIC/9000, STATUS register 0 of interface cards contains the card ID. Interface cards that are available on a PC, but not on an HP BASIC/9000 are identified with ID numbers greater than or equal to 300.

## STORE

HP BASIC/9000 PROG files and HTBasic for Windows PROG files are not compatible. To move programs between the two environments, use ASCII program files.

---

## STORE SYSTEM

In HP BASIC/9000 this statement stores a copy of the operating system with all loaded BINs already linked in. Under HTBasic for Windows this is not possible. Use AUTOST to load HTBasic for Windows device drivers.

## SYMBOL

LORG 5 moves the symbol origin from (0,0) to (5,8). In HP BASIC/9000 it moves the origin to (4.5,7.5).

## SYSBOOT

HTBasic for Windows does not support SYSBOOT, which under HP BASIC/9000 reboots the computer. Since HTBasic for Windows runs as a guest of the operating system, it is considered inappropriate to reboot the computer. Under some operating systems, rebooting the computer inappropriately can cause loss of data.

## SYSTEM

The SYSTEMS("DISP LINE") function is an HTBasic for Windows extension that returns the present contents of the display line.

The SYSTEMS("VERSION:HTB") function returns the HTBasic for Windows version description, for example, "Windows Release 7.1". This function can be useful for programs that run on both HP BASIC/9000 and HTBasic for Windows systems, enabling them to determine which system they are currently running on. This example sets a variable according to the system running the program:

```
10      SUB Which_system
20      COM /Which_system
         /Basic_9000
30      Basic_9000= (SYSTEM$
         ("VERSION:HTB")="0")
40      SUBEND
```

## TIMEZONE

HTBasic for Windows does not require this statement and will return an error if an attempt is made to execute it. The editor will allow it to be entered, and the syntax checker will check it for correctness to allow you to develop programs and run them under HP BASIC/9000. HP BASIC/9000 requires this statement for two reasons:

- HP BASIC/UX keeps a time clock independent of the UNIX time
- It is possible to boot HP BASIC/WS on a computer whose real-time clock is set to Greenwich Mean Time (GMT)

## TRANSFER

Currently TRANSFER is only supported for inbound Serial. TRANSFER with any other device or file generates an error.

## WRITEIO

See the explanation under READIO for some differences. Other processors cannot execute the Motorola code accessed by WRITEIO 9827. The code must be rewritten.

## XREF

HTBasic for Windows allows a string variable to specify the name of the subprogram or function.

## Summary

This overview of the differences between HTBasic for Windows and other HP BASIC platforms should give you a good understanding of how easily your applications will move to this platform. If you have any further questions, please refer to the documentation supplied with the product or contact TransEra or your international distributor.