

HP 9000 Series 200 and 300 BASIC 4.0 Language System



Technical Data,* January 1986



BASIC 4.0 Capabilities

The BASIC 4.0 Language System is anything but basic. It adds to the inherent simplicity of BASIC the computational power usually found in FORTRAN, ALGOL, and APL. Beyond that, high-performance, I/O-intensive constructs are incorporated, giving this BASIC the highest performance found in interactive instrument control systems on the market today. BASIC 4.0 puts mini-computer architecture and power on your desk, bench or test-bay, not down the hall. With the BASIC Language System, high performance and easy programming are not mutually exclusive — both are provided for your benefit. This increase in your software productivity is applicable to a wide spectrum of task and project demands because this BASIC provides an impressively rich command set at your finger tips. Sophisticated applications can be met with state-of-the-art hardware power as your resource. At the same time simple tasks can be solved in minutes, not hours.

BASIC 4.0 increases your software productivity by providing:

- **A computer system at your fingertips.** You don't wait to "log-on" — just sit down and start developing solutions.
- **Interactive editing.** The editor is always present and provides powerful features such as the ability to search and replace strings and copy blocks of lines. The entire program is viewable while you edit lines.
- **Statement syntaxing at program entry.** You don't have to run your program to find syntax errors; BASIC 4.0 proofreads your code.
- **Extensive debug and trace tools.** You can single-step your program one line at a time to check program logic flow and variable assignments. Tracing tools allow a line-by-line log of line numbers and variable changes to be printed while your software is running.

* Data subject to change.

- **Structured programming constructs** that organize your code. Complex algorithms are easier to develop, document and maintain.
- **Numerous pre-defined functions.** Mathematical and string functions and matrix statements enable you to develop more compact code.
- **Independent subprograms.** You can break your software down into smaller modules for easier development and coding. Subprograms can be stored and loaded separately so you can develop disc-resident subprogram utility libraries.
- **Dynamic variable allocation.** This increases the flexibility of your subprogram libraries so you can efficiently manage memory space as you go.
- **Labeled common,** to increase the flexibility of your subprograms. You can have several independent "common" variable blocks, allowing easy subprogram variable usage and subprogram communication.
- **A unified device and mass storage I/O system.** You can easily redirect I/O to and from devices and files. In essence, your I/O routines can access a mass storage file in the same way as an external device providing an excellent test harness for your I/O dependent code.
- **Support for color.** Simple BASIC statements make it easy to use bit-mapped color displays as well as the Model 236C display and the HP 98627A color video interface.
- **Support for a wide variety of mass storage devices,** including large capacity discs, EPROM, bubble memory, RAM volumes and the Shared Resource Management System.
- **Support for a wide variety of peripherals,** including the HP-HIL mouse, knob, data tablets and touchscreen.
- **Printer and plotter spooling** on the Shared Resource Management System. Your computer can be freed up from performing lengthy printing or plotting tasks.

BASIC 4.0 increases your software performance by:

- **Modular packaging.** For optimum memory utilization you load only the language extensions and driver modules that you need.
- **High performance.** Computational and I/O performance can be improved due to the range of processors available (8 to 16.6 MHz). Support of the MC68881 floating point coprocessor with the MC68020 or the floating point card allows for improvement of floating point performance. BASIC programs can call Pascal or assembly language subroutines for further improvement of speed-critical routines.
- **Powerful I/O drivers.** Advanced I/O features such as fast handshake and interrupt buffer transfers, DMA and event controls are provided. Memory mapped I/O allows for high volume data acquisition directly to memory.
- **Optimized transactional I/O.** I/O path set-up times are reduced to the bare minimum. I/O paths can be turned around quickly, allowing you to go from input to output quickly. BASIC 4.0 gives your I/O-dependent software unparalleled agility.
- **The richest HP-IB I/O command set in the industry.** Instrument control over HP-IB is quick and to the

point. More than 20 commands are dedicated to HP-IB I/O to give you performance and flexibility in letting HP-IB work for you.

- **15 levels of prioritized software interrupt.** You can optimize I/O operations to closely match peripheral device speeds without tying down the system. With 15 levels of interrupts, a multitude of external events can be serviced with varying levels of priority set by you. Even the softkeys can be used as a system interrupt source.

These are just a few reasons why the BASIC 4.0 Language System can increase your software performance and productivity. BASIC 4.0 gives you high performance AND easy software development, taking you from problem definition to solution quickly and efficiently. You generate results, not just software.

BASIC 4.0 Operating Characteristics

Language System Memory Requirements

Option	RAM Req'd.*	Disc Req'd.*
BASIC 4.0 Core System†	279	237
Language Extensions		
Clock	4	4
Error Messages†	7	7
Graphics†	46	43
Graphics Extensions	27	27
I/O†	11	11
Keyboard Extensions	13	12
Lexical Order	10	10
Mass Storage	9	9
Matrix	20	21
Program Development†	13	13
Shared Resource Management	46	44
Transfer	32	30
XREF	6	6
Knob 2.0	0	0
Interface Card Drivers		
BCD	2	2
Datcomm†	7	6
GPIO†	5	5
High-speed Disc Interface	2	2
HP-IB†	12	12
Serial†	4	5
Mass Storage Device Drivers		
Bubble Memory	3	3
CS/80 Disc	11	4
EPROM	3	3
HP 9885 Disc†	4	4
Small Disc†	7	7
CRT Drivers		
Alpha plane CRT Driver‡§	11#	5
Bit-mapped display CRT Driver§	28#	10

* RAM and disc requirements are rounded to the nearest K bytes (K=1024).

† Contains keywords or drivers from the BASIC 2.0 system.

‡ Not supported on all models.

§ One of these CRT drivers must be loaded.

Measured while binary is in use. If binary is loaded but never used, the RAM size is similar to the disc size for the same binary.

Range

Math uses 64-bit IEEE-P754 floating point format.
Optional hardware floating point is fully compatible.

Real Precision:

-1.797693134862315E+308 to
-2.225073858507202E-308
0
2.225073858507202E-308 to
1.797693134862315E+308

Integer Precision:

-32768 to +32767



Math Hierarchy

Highest Priority: ()

User defined and built-in functions

^

*, /, MOD, DIV

+, - (unary and binary + and -)

=, <, >, <=, >=, ≥

NOT

AND

Lowest Priority: OR, EXOR

BASIC 4.0 Keyword Summary

BASIC 4.0 Core System

General Functions

ABS — returns the absolute value of its argument.

ACS — returns the principal value of the arc cosine of its argument.

AND — returns the logical conjunction of its arguments.

ASN — returns the principal value of the arc sine of its argument.

ATN — returns the principal value of the arc tangent of its argument.

BINAND — returns the value of a bit-by-bit AND of its arguments.

BINCMP — returns the value of the 1's complement of its argument.

BINEOR — returns the value of a bit-by-bit exclusive-OR of its arguments.

BINIOR — returns the value of a bit-by-bit inclusive-OR of its arguments.

BIT — returns an INTEGER representation of the contents of a specified bit of its argument.

COS — returns the cosine of the argument.

CRT — returns the select code for the internal CRT.

DIV — returns the integer portion of the quotient of the dividend divided by the divisor.

DROUND — returns the value of a numeric expression, rounded to the specified number of significant digits.

ERRDS — Error Device Status - returns the address of the I/O involved in the most recent I/O error.

ERRL — returns a value of 1 if the most recent execution error occurred during the specified line, otherwise 0 is returned.

ERRN - returns the number of the most recent program execution error.

EXOR — returns the exclusive-OR of its arguments.

EXP — raises the base of the natural logarithm (Naperian $e = 2.71828182845905..$) to the power of the argument.

FRACT — returns the fractional part of the specified value.

INT — returns the greatest integer which is less than or equal to the evaluated expression.

KBD — returns select code of the built-in keyboard.

KNOBX* — returns the number of horizontal pulses generated by the rotary control knob since the KNOBX counter was last zeroed.

KNOBY* — returns the number of vertical pulses generated by the rotary control knob since the KNOBY counter was last zeroed.

LGT — returns the logarithm (base 10) of its argument.

LOG — returns the natural logarithm (base e) of the argument.

MAXREAL — returns the largest positive REAL number available in the range of the computer.

MINREAL — returns the smallest positive REAL number available in the range of the computer.

MOD — returns the remainder of a division of its arguments.

MODULO — returns the integer remainder resulting from a division.

NOT — returns the logical complement of its argument.

NPAR — returns the number of parameters passed in the call to the currently executing subprogram or multi-line function.

OR — returns the logical inclusive-OR of its arguments.

PI — returns an approximate REAL value for pi.

PROUND — returns a value rounded to the specified power-of-ten.

PRT — returns the value 701, the most common select code for a peripheral printer.

RES — returns the last keyboard numeric computation result.

RND — returns a pseudo-random number which is greater than 0 and less than 1.

ROTATE — returns an INTEGER value representing the value obtained by creating a bit-string version of its argument and rotating the argument the number of bit positions specified.

SC — returns the select code of an I/O path name.

SGN — returns a value of 1 if the given expression is positive, 0 if equals 0, and -1 if it is negative.

SHIFT — returns an INTEGER value representing the value obtained when its argument is converted to a 16-bit string pattern and shifted the number of positions specified.

SIN — returns the sine of its argument.

SQR — returns the square root of the argument.

TAB — moves the print position to the specified column on the current printing device.

TABXY — moves the print position to the column and line specified on the internal CRT.

TAN — returns the tangent of its argument.

TIMEDATE — returns the current value of the real-time clock.

String Functions

CHR\$ — converts a numeric value into a character.

DVAL — returns the REAL whole number value of the string expression in the radix specified.

DVAL\$ — returns the ASCII string containing the specified whole number converted to the radix specified.

* KNOBX and KNOBY function differently if BIN file KNB2_0 is loaded.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

ERRM\$ — Error Message — returns the text of the error message of the most recent error.

IVAL — returns the INTEGER value of the string expression from -32768 to 32767 in the radix specified.

IVAL\$ — returns the ASCII string of the integer value converted to the specified radix.

KBD\$ — returns the contents of the keyboard buffer.

LEN — returns an integer representing the current number of characters in a string expression.

LWC\$ — Lowercase — converts alpha characters to their lowercase equivalents in the current lexical order.

NUM — converts the first character of the string expression to its equivalent decimal value ASCII character code.

POS — returns the first position of a substring within a string.

REV\$ — returns the specified string in reverse order.

RPT\$ — returns a string containing its argument string repeated the specified number of times.

SYSTEM\$ — returns system status and configuration information. SYSTEM\$ request items are:

AVAILABLE MEMORY	MASS MEMORY
CRT ID	MASS STORAGE IS
DUMP DEVICE IS	MSI
PRINTALL IS	SYSTEM ID
PRINTER IS	SYSTEM PRIORITY
SERIAL NUMBER	TRIG MODE
KBD LINE	VERSION: option name

TRIM\$ — returns a string with any leading or trailing blanks removed.

UPC\$ — Uppercase — converts all lowercase alpha characters in a string to their corresponding uppercase characters in the current lexical order.

VAL — converts a string expression into a numeric value.

VAL\$ — returns a string which represents the value of the argument.

General Statements

ALLOCATE — dynamically reserves memory space for string variable, string arrays and numeric arrays during program execution.

BEEP — outputs an audible tone with programmable frequency and duration.

CALL — transfers program execution to the specified subroutine subprogram and specifies values for the pass parameters.

CASE,CASE ELSE — structured sequence used with SELECT for selecting alternative actions depending on the result of a conditional test (see SELECT).

COM — dimensions and reserves space for simple and array variables in a special "common" memory area so more than one program segment can access the variables.

CONT — resumes program execution without pre-run initialization (see PAUSE).

CONTROL — sends control information to an interface or to the internal table associated with an I/O path name.

CSUB — the compiled SUB statement is used to identify a compiled Pascal or Assembly language subprogram.

DATA — allows numeric values or string literals to be defined and assigned with the READ statement.

DEALLOCATE — reclaims the memory space reserved by the ALLOCATE statement.

DEF FN — indicates the beginning of a function subprogram, defines its parameters and determines whether it returns a REAL or string value.

DEG — selects degrees as the unit of measure for expressing angles.

DEL — deletes all program lines within the range specified by the beginning and ending line identifiers.

DELSUB — deletes one or more subprograms or multiple line functions from memory.

DIM — dimensions and reserves memory for REAL numeric arrays, simple strings and string arrays.

DISABLE — disables all active ON (event) statements except ON ERROR, ON END and ON TIMEOUT.

DISP — causes the values of the print list to be sent to the display line on the CRT.

DISP USING — causes the values of the print list to be sent to the display line on the CRT according to the format specified by IMAGE.

DUMP ALPHA — transfers the contents of the CRT screen to the device currently specified by DUMP DEVICE IS.

EDIT — accesses the editor to enter a new program or modify program lines.

ELSE — is part of the IF...THEN...ELSE...END IF construct, provides an alternative action to be performed.

ENABLE — re-enables all ON (event) statements which were suspended by DISABLE.

END — marks the end of a main program segment.

END IF — marks the end of an IF...THEN...ELSE...END IF construct.

END LOOP — marks the end of a LOOP construct.

END SELECT — marks the end of a SELECT construct.

END WHILE — marks the end of a WHILE construct.

EXIT IF — provides conditional exit from a structured loop. (See LOOP).

FN — is used to call a user-defined, multiple-line function subprogram.

FNEND — is the last statement in a multiple-line function subprogram.

FOR — defines a loop which is repeated until the loop counter exceeds a specific value.

GOSUB — transfers program control to a subroutine at the specified line.

GOTO — transfers program execution to the specified line.

IF...THEN,IF...THEN...ELSE...END IF — provides conditional branching or execution of one or more statements when the specified condition is true.

INTEGER — declares INTEGER variables, dimensions INTEGER arrays and reserves memory for either.

LET — assigns a value to a simple numeric variable or assigns a set of characters to a simple string variable (LET is optional).

LIST — causes the entire program, or lines specified within a range list to be output to the current PRINTER IS device.

LIST # — causes the entire program, or lines specified within a range list to be output to the specified printing device.

LIST BIN — lists options currently loaded in memory.

LOOP — repeats statements in a structured loop as long as the EXIT IF expression is FALSE.

NEXT — used with FOR in looping. (See FOR).

OFF/ON ERROR — disables/enables an event-controlled branch occurring whenever a trapped error occurs.

OFF/ON KBD, OFF/ON KBD ALL — disables/enables interrupt branching when a key is pressed.

OFF/ON KEY (LABEL) — disables/enables an event-controlled branch to occur when a special function key (softkey) is pressed.

OFF/ON KNOB — disables/enables an event-controlled branch to occur every specified number of seconds if the rotary control knob or HP-HIL mouse has generated pulses since the last interrupt.

ON event GOTO/GOSUB/CALL/RECOVER — is used with ON ERROR, ON END, ON KBD, ON KEY, ON INTR, ON KNOB and ON TIMEOUT to cause a branch in program execution based on the specified GOTO, GOSUB, CALL or RECOVER statement. With RECOVER, the branch will occur regardless of the current program environment.

ON expression GOSUB — transfers program execution to one of several subroutines depending on the value of the expression.

ON expression GOTO — transfers program execution to one of several line identifiers depending on the value of the expression.

OPTION BASE — specifies the default lower bound of subscript values for all array dimensions.

OPTIONAL — is used in formal parameter lists to declare which parameters are optional when passing to a subroutine or function subprogram.

PAUSE — halts program execution without altering the data or state information so program operation can be continued.

PRINT — prints the specified items in the print list to the PRINTER IS device.

PRINT USING — outputs the items in the list according to the format specified by IMAGE.

PRINTALL IS — assigns a destination printing device for output which is normally sent to the system message line of the CRT.

PRINTER IS — specifies the current printing device or file for any PRINT, PRINT USING, CAT and LIST statements.

RAD — sets radians as the unit of measure for expressing angles.

RANDOMIZE — is used to modify the seed used by the computer's random number generator.

READ — reads values from DATA statements and assigns them to variables.

READIO — allows the reading of either bytes or words from any interface register.

REAL — reserves storage for full precision, floating point variables and arrays.

REM or ! — allows comments to be inserted into your program.

REN — renumbers the program lines.

REPEAT-UNTIL — this loop construct repeats the statements in a structured loop until the expression following UNTIL is true.

RESTORE — specifies the DATA statement which will be READ next.

RESUME INTERACTIVE — enables the execute, PAUSE, STOP, STEP, CLR I/O, and RESET keys after a SUSPEND INTERACTIVE statement.

RETURN — marks the end of a subroutine.

RETURN expression — the last executed statement of a multi-line function.

RUN — begins program execution at a specified line.

SCRATCH — erases the program in memory.

SCRATCH A — erases the program in memory and all variables, resets most system modes to a power-up condition.

SCRATCH BIN — performs a SCRATCH and erases all options in memory.

SCRATCH C — clears all variables including those in COM.

SELECT — this structured statement allows the execution of several different actions depending upon the result of a conditional test when used with CASE or CASE ELSE. SELECT must end with END SELECT.

SET TIME — sets the time-of-day given by the real-time-clock.

SET TIMEDATE — sets the time-of-day and the date given by the real-time-clock.

STATUS — provides the status value from an interface register, or the internal table associated with an @ name, into the specified numeric variables.

STEP — provides an increment size in a FOR loop.

STOP — terminates execution of the program.

SUB — is the first statement in a subroutine subprogram and specifies the formal parameter list for the subprogram.

SUBEND — is the last statement of a subroutine subprogram and transfers execution back to the calling program segment.

SUBEXIT — can be used within the body of a subroutine subprogram to transfer execution back to the calling program segment.

SUSPEND INTERACTIVE — disables execute, PAUSE, STOP, STEP, CLR I/O and (optionally) RESET keys.

SYSBOOT — returns system control to the Boot ROM.

SYSTEM PRIORITY — sets system priority for event interrupts.

UNTIL — used with REPEAT for looping. (See REPEAT).

WAIT — nondestructively suspends program execution for an approximate amount of time in seconds.

WHILE — repeats a structured loop as long as the WHILE expression is true.

WRITEIO — writes either bytes or words to any interface register.

Mass Storage and I/O Statements

ASSIGN @ name — allows the current attributes associated with the @ name to be changed without cancelling it.

ASSIGN @ name TO — links an @ name to a device or file, providing a unified I/O structure to a program.

ASSIGN @ name TO * — cancels an @ name assignment and its attribute.

CAT, CAT TO — lists the contents of a mass storage medium's directory or information about a specific PROG file.

COPY — provides the capability to copy single files or to BACKUP entire media.

CREATE ASCII — creates an ASCII type file on the mass storage device.

CREATE BDAT — creates files which hold binary data types.

ENTER, ENTER @ name — is used to read data from a device, file or string and assign the values read to the variables in the list.

ENTER USING — is used to read data from a device, file or string and assign the values read to the variables in the list according to the specified IMAGE.

FORMAT OFF/ON — used in an ASSIGN statement to specify whether data is to be interpreted as ASCII (ON) or HP 9000 BASIC internal format (OFF).

GET — reads the specified ASCII file into the computer's program memory.

IMAGE — is referenced by the USING clause of the PRINT, OUTPUT, DISP, LABEL and ENTER statements to provide formats for I/O operations.

INITIALIZE — sets up a directory for a new medium. Data previously on a medium is destroyed by initialization.

INPUT — is used to assign keyboard input to program variables.

LINPUT — assigns characters entered from the keyboard to a string variable or substring.

LOAD — retrieves and places in memory any program file (PROG) which was previously stored with the STORE statement.

LOAD BIN — retrieves and puts into memory a binary program file (BIN).

LOADSUB ALL FROM — loads all of the BASIC subprograms in a program file (PROG) into the computer's memory.

LOADSUB subprogram FROM — used to load a single subprogram by specifying the subprogram name.

MASS STORAGE IS — specifies the current system mass storage device (see MSI).

MSI — can be used to abbreviate MASS STORAGE IS.

OFF END — deactivates the end-of-statement branch previously activated by an ON END statement.

ON END — enables an event-controlled branch to occur when an end-of-file condition occurs for a mass storage device.

OFF/ON TIMEOUT — disables/enables an event-controlled branch to occur when an I/O timeout occurs on the specified interface.

OUTPUT, OUTPUT @ name — copies data from the variables in the output list to the specified destination.

OUTPUT USING — copies data from the variables in the output list to the specified destination using the specified IMAGE.

PROTECT — establishes (or changes) the protect code used on non-ASCII files.

PURGE — erases a file by deleting the file's entry from the mass storage medium's directory.

RE-SAVE — copies all, or part, of the program in the computer's memory onto a mass storage medium in ASCII format.

RENAME — changes the file's name on the mass storage medium's directory.

RE-STORE — copies a BASIC program in the system's memory onto a mass storage medium in internal format.

SAVE — creates an ASCII file and stores all, or part, of the BASIC program in the computer's memory in source code strings.

STORE — creates a PROG file and stores in it the entire BASIC program in internal format.

STORE SYSTEM — stores BASIC and options currently in memory into a SYSTM file.

USING — used by the PRINT, OUTPUT, DISP, LABEL and ENTER statements, provides formats for I/O operations.

Clock Option

DATE — accepts a date in the form of DD MMM[-]YYYY and converts it to the number of seconds between that date and the Julian date: 24 Nov - 4713.

DATE\$ — computes the date in DD MMM[-]YYYY format from Julian date.

OFF CYCLE — disables ON CYCLE event branching.

OFF DELAY — disables ON DELAY event branching.

OFF TIME — disables event branches set by ON TIME.

ON CYCLE — sets up and enables a periodic event branch each time the specified number of seconds elapses.

ON DELAY — sets up and enables an event branch to occur the specified number of seconds after ON DELAY is executed.

ON TIME — sets up and enables an event branch when the time specified matches the time in the computers real-time clock.

TIME — converts a string of the form HH:MM:SS to seconds between midnight and the specified time.

TIME\$ — converts the Julian time to a string of the form HH:MM:SS.

Cross Reference Option

Provides the XREF keyword which obtains a cross reference listing of identifiers in a program or subprogram.

Error Messages Option

Contains the text for error messages.

Graphics Option

ALPHA ON/OFF — turns the CRT alphanumeric area ON or OFF.

AXES — draws a pair of axes with optional equally spaced tick marks.

CLIP — redefines the soft clip area and turns clipping on.

CLIP ON/OFF — turns clipping on or off.

CSIZE — sets the size and aspect ratio for characters. (Used with the LABEL and SYMBOL statements.)

DRAW — draws a line from the pen's current position to the specified X,Y coordinate position using the current line type and pen number.

DUMP DEVICE IS [EXPANDED] — specifies which device receives the data when DUMP GRAPHICS or DUMP ALPHA is executed. EXPANDED allows a 2-for-1 expansion along each axis, and rotates the resulting image 90°.

DUMP GRAPHICS — copies the contents of graphics memory onto the device currently specified by DUMP DEVICE IS.

FRAME — draws a frame around the current clipping area using the current pen number and line type.

GCLEAR — clears a CRT graphics display or sends a paper advance command to a plotter.

GINIT — resets all global graphics parameters to their power-on values.

GLOAD — allows you to load the contents of an integer array into graphics R/W memory for display.

GRAPHICS ON/OFF — turns the graphics display on the ALPHA CRT on/off.

GRID — draws a full grid pattern.

GSTORE — allows the copying of graphics R/W memory contents in coded form into an integer array.

IDRAW — draws a line from current pen position to a position specified as an X,Y increment from the current position of the pen.

IMOVE — moves to a position specified as an X,Y increment from the current position of the pen without drawing.

IPLOT — incremental plot is similar to IDRAW and IMOVE. The pen control is determined by a value following the X and Y coordinates.

LABEL — outputs data in text form on graphics devices.

LABEL USING — outputs data in text form on graphics devices using the specified IMAGE.

LDIR — determines the angle at which labeling statements draw the characters.

LINE TYPE — selects a line type and repeat length for lines, labels, frames, axes and grids.

LORG — specifies the origin position of the label relative to the current pen position.

MOVE — update the current position of the pen to the specified X,Y coordinate position.

PDIR — specifies rotation for IPLOT, RPLOT, RECTANGLE, POLYGON and POLYLINE.

PEN — selects the pen used by the plotter. PEN -1 erases lines on the CRT.

PENUP — lifts the physical pen from the plotting surface.

PIVOT — specifies a rotation of axes which is applied to all subsequent lines drawn by DRAW and IDRAW statements.

PLOT — moves or draws to the specified X,Y coordinate. Pen control is specified following the X,Y coordinates.

PLOTTER IS — selects a plotter or file to receive the HPGL plotting commands.

RATIO — returns a value equal to the ratio of the physical dimensions of the current PLOTTER IS device's hard clip limits.

RPLOT — relative plot moves or draws to the X,Y position relative to the last absolute pen position. Pen control follows the X,Y values.

SHOW — isotropically defines the plotting units mapped on the VIEWPORT area.

SYSTEM\$ — extended to include "PLOTTER IS" and "GRAPHICS INPUT IS".

VIEWPORT — specifies an area in device units onto which WINDOW or SHOW statements are mapped.

WINDOW — specifies the minimum and maximum values for the plotting area specified by VIEWPORT.

Graphics Extensions Option

AREA COLOR — specifies a fill color using the HSL color model. Used with statements requiring area fill.

AREA INTENSITY — specifies the fill color using the RGB color model. Used with statements requiring area fill.

AREA PEN — specifies a pen number to use as the fill color. This is used with statements requiring area fill.

DIGITIZE — determines the X and Y coordinates of the current graphics input device's locator when the digitize button (or stylus) is pressed.

EDGE — used in the IPLOT, PLOT, POLYGON, RECTANGLE, RPLOT and SYMBOL statements to specify that the lines making up the edges of an area are to be drawn using the current line type and pen.

FILL — used in the IPLOT, PLOT, POLYGON, RECTANGLE, RPLOT and SYMBOL statements to specify that the figure drawn is to be filled with the current fill color.

GESCAPE — is used for making a request to a graphics device driver. This is a device and driver-dependent command.

GRAPHICS INPUT IS — defines the device for graphics input.

IPLOT — extended to allow incremental plotting by specifying a sequence of commands in an array.

PLOT — extended to allow a sequence of plot commands in an array.

POLYGON — outputs a closed regular polygon or part of a regular polygon.

POLYLINE — outputs an unfilled regular polygon or part of a regular polygon.

READ LOCATOR — determines the X and Y coordinates of the current graphics input device's locator without requiring the digitize button (or stylus) to be pressed.

RECTANGLE — draws a rectangle with dimensions specified as displacements from the current position.

RPLOT — extended to allow relative plotting by specifying a sequence of commands in an array.

SET ECHO — sets the current PLOTTER IS device's locator position to the specified position.

SET LOCATOR — sets the locator position on the current GRAPHICS INPUT IS device.

SET PEN...COLOR — defines one or more entries in the color map using the HSL color model.

SET PEN...INTENSITY — defines one or more entries in the color map using the RGB color model.

SYMBOL — allows labelling with user-defined symbols.

TRACK..IS ON/OFF — enables the GRAPHICS INPUT IS device's locator to be tracked by the current graphics output device's cursor (or pen) position.

WHERE — returns the current pen position in X and Y variables. It also can return information on whether the pen is up, down, inside or outside the viewport and hard clip boundaries.

I/O Option

ABORT — resets the interface functions for HB-IB interface.

ASSIGN @ — extended to support the following attributes: BYTE or WORD, CONVERT IN or OUT, EOL or EOL OFF, PARITY OFF or ODD or EVEN or ZERO or ONE and RETURN.

BREAK — causes a Break sequence to be sent on the RS-232 or Datacomm interface.

BYTE — defines byte (8-bit) type transfer for an interface as opposed to word type transfer (used with ASSIGN @).
CLEAR — allows the active controller to put HP-IB devices into a defined device-dependent state.
CMD — is used with the SEND statement to send numeric or string expressions over HP-IB with ATN true.
CONVERT IN or OUT — used in the ASSIGN @ statement to define conversion tables for ENTER (IN) or OUTPUT (OUT). The conversion table can be set up to convert the original character to its replacement by INDEXing into the table or by original/replacement PAIRS in the table.
CONVERT IN/OUT OFF — causes code conversion set up by CONVERT IN/OUT to be discontinued.
DATA — is used with the send statement to send numeric or string expressions over HP-IB with ATN false if the computer is the active controller and is addressed to talk.
DISABLE INTR — sends a word to the interrupt-enable register of the specified interface, disabling all interrupts from that interface.
ENABLE INTR — enables the specified interface to generate an interrupt which can cause end-of-statement branches. Also allows the setup of Powerfail interrupts (see ON INTR.)
END — outputs on EOI to an HP-IB device when used in an OUTPUT or SEND list.
EOL — defines End-Of-Line sequence to be sent with PRINT and OUTPUT.
EOL - [END,DELAY] — allows terminating I/O attributes when used with the ASSIGN statement.
EOL OFF — Resets EOL to Carriage Return/ Line Feed.
IMAGE — extended to include several new format specifiers.
LISTEN — is used with the SEND statement to send listen addresses.
LOCAL — returns all specified devices to their local state.
LOCAL LOCKOUT — sends the LLO (local lockout) message, preventing an operator from returning the device to control by its front (local) panel.
MLA — is used with the SEND statement to send the HP-IB interface card's listen address.
MTA — is used with the SEND statement to send the HP-IB interface card's talk address.
OFF/ON INTR — disables/ enables an event-controlled branch to occur when an interface card (enabled by ENABLE INTR) requests an interrupt. Priority may also be specified. With the Powerfail option this statement allows branching to an interrupt service routine if power should fail.
OFF SIGNAL — disables ON SIGNAL software interrupts. (See ON SIGNAL).
ON SIGNAL — sets up and enables an event branch when the SIGNAL statement is executed.
PARITY — used with ASSIGN @ to specify parity type.
PASS CONTROL — used to pass active control on the HP-IB to another controller on the bus.
PPOLL — returns a byte representing the 8 status-bit messages of those devices specified. Device responds to parallel poll.

PPOLL CONFIGURE — programs the logical sense and data bus line on which the specified device responds to a parallel poll.
PPOLL RESPONSE — stores the specified value in the parallel poll response register of the HB-IB interface.
PPOLL UNCONFIGURE — disables the parallel poll response of the specified devices.
PRINT — extended to support WIDTH and EOL sequence.
PRINTALL IS — extended to support WIDTH and EOL sequence.
PRINTER IS — extended to support WIDTH and EOL sequence.
REMOTE — places HB-IB devices having remote/local capabilities into the remote state of operation.
REQUEST — used to request service from the active controller on HP-IB.
RESET — resets an interface, file or buffer.
RETURN — when used with the ASSIGN @ statement, returns a value indicating the outcome of an assign statement.
SEC — is used with the send statement to send secondary commands and addresses over HP-IB.
SEND — sends control information and data to an HP-IB interface.
SIGNAL — generates a software interrupt, used with ON SIGNAL.
SPOLL — returns an integer whose low order byte contains the serial poll response from the addressed device; used with HP-IB.
TALK — is used with the send statement to address a device to talk on the HP-IB.
TRIGGER — initiates device-dependent action from either a selected device or all devices addressed to listen on the HP-IB.
UNL — is used with the SEND command to send the bus unlisten message (UNL) on the HP-IB.
UNT — is used with the SEND command to send the bus untalk message (UNT) on the HP-IB.
WIDTH — used in the PRINTER IS and PRINTALL IS command to specify the printer width.
WIDTH OFF — used in the ASSIGN @, PRINTER IS, and PRINTALL IS statements to set the printer width to infinite.
WORD — used with ASSIGN @ to define word (16-bit) type of interface transfer.

Keyboard Extensions Option

EDIT KEY — allows "edit-key" mode to be entered for defining the special function keys as typing aids.
LIST KEY — lists the definition of the special function keys.
LOAD KEY — loads the definitions of the special function keys from the specified mass storage file.
RE-STORE KEY — stores special function keys into a previously used file name.
SCRATCH KEY — deletes special function key definitions.
STORE KEY — stores the special function key definitions in the specified file.

Lexical Order Option

LEXICAL ORDER IS — defines the collating order used by all string relational operators and the UPC\$ and LWC\$ functions. Six language tables are present: ASCII, STANDARD, FRENCH, GERMAN, SPANISH, and SWEDISH. User-defined tables are also supported.

SYSTEM\$ — extended to include "KEYBOARD LANGUAGE" and "LEXICAL ORDER IS".

Mass Storage Option

CAT — extended to catalog the subprograms and binaries in a PROG type file and the binaries in a BIN type file. New control options have also been added (see COUNT, NO HEADER, SELECT and SKIP).

CAT TO — extended to catalog a mass storage directory to a string array.

CHECKREAD OFF — disables verification of data written to a mass storage file.

CHECKREAD ON — enables verification of data written to a mass storage file.

COUNT — used in the CAT statement to return the number of items listed.

NO HEADER — used with the CAT statement to disable header information in a catalog listing.

PRINT LABEL — writes a string expression to the volume label of a media.

READ LABEL — reads the volume label of a media to a string variable.

SELECT — used with the CAT statement to select entries that begin with the specified characters.

SKIP — used with the CAT statement to skip the specified number of files.

Matrix Option

BASE — returns the lower bound for the specified dimension of an array.

DET — returns the determinant of the specified matrix, or of the last matrix inverted.

DOT — returns the inner (dot) product of the two specified vectors.

MAT +, -, ., /, <, ≤, =, < >, >, ≥ — perform the specified array math, element by element, on the specified numeric arrays (. is for element by element multiply).

MAT* — matrix multiplication.

MAT...= — array assignment or initialization.

MAT CSUM — computes a column sum on the specified matrix.

MAT IDN — sets a square matrix to the identity matrix: the main diagonal becomes all 1's, all other elements become 0's.

MAT INV — produces the inverse of a square matrix.

MAT REORDER — used to rearrange the data in an array according to an associated pointer vector.

MAT RSUM — computes the row sum of each row in the specified matrix.

MAT SORT — used to reorder an array or set up a reorder vector according to specified keys in either ascending or descending order.

MAT TRN — computes the transpose of the specified matrix.

MAX — returns the largest value in a list of values.

MIN — returns the smallest value in a list of values.

RANK — returns the number of dimensions in an array.

REDIM — Redimension — changes the subscript bounds of a list of arrays.

SIZE — returns the number of elements in the specified dimension of an array.

SUM — returns the sum of all elements in an array.

Program Development Option

CHANGE — searches the program for the specified string and replaces it with the specified replacement string.

COPYLINES — copies lines of a program from one location to be inserted at another location in the program.

FIND — searches a program for the specified string.

INDENT — indents your program to reflect the structure based on the statements used.

LOADSUB FROM — provides a library feature by loading all undefined subprograms or functions from a mass storage file.

MOVELINES — moves lines from one location in a program to another.

SECURE — makes program lines unlistable.

TRACE ALL — allows the tracing of program flow and variable assignments while a program is running.

TRACE OFF — disables all tracing activity enabled by TRACE ALL or TRACE PAUSE.

TRACE PAUSE — causes program execution to pause before executing the specified line.

Shared Resource Management Option

CREATE DIR — creates a directory file.

LOCK...CONDITIONAL — prevents other Shared Resource workstations from accessing a shared file which is currently in use.

UNLOCK — removes exclusive access to a file which was protected by the LOCK statement.

The following BASIC Mass Storage Statements are expanded to encompass the use of Shared Resource Management mass storage files:

ASSIGN	LOADSUB...FROM
CAT	MASS STORAGE IS
CAT PROG	PROTECT DELETE
CAT TO	PROTECT MANAGER
CONTROL	PROTECT READ
COPY	PROTECT WRITE
CREATE ASCII	PURGE
CREATE BDAT	RENAME
GET	RE-SAVE
LOAD	RE-STORE
LOAD BIN	RE-STORE KEY
LOAD KEY	SAVE
LOADSUB	STORE
LOADSUB ALL FROM	TRANSFER
LOADSUB FROM - command only	

Transfer Option

ABORTIO @ — causes early termination of a transfer to an I/O path.

BUFFER — used in the DIM, COM, INTEGER, REAL, SUB, DEF FN and ASSIGN @ statements to define the buffer area to hold data for the TRANSFER statement.



CONT — used in the TRANSFER statement to indicate that the transfer is to continue until an end-of-transfer condition is encountered.

CONTROL — extended for new device registers and control.

COUNT — used in the TRANSFER statement to terminate a transfer when the specified number of bytes have been transferred.

DELIM — used with the TRANSFER statement to assign a delimiter for terminating an inbound transfer.

END — used with the TRANSFER statement to specify an interface-dependent message for terminating an inbound transfer (such as EOI for HP-IB).

ENTER — extended to allow entering data from a buffer.

EOR — used in the TRANSFER statement to specify the End-Of-Record delimiter.

ERRL - Error Line — extended for TRANSFER.

ERRN - Error Number — returns the number of most recent error; extended for TRANSFER.

OFF EOR @ — cancels on EOR @ event branching.

OFF EOT @ — cancels ON EOT @ event branching.

ON EOR @ — sets up and enables an event branch whenever a record terminator is encountered in a TRANSFER operation.

ON EOT @ — sets up and enables an event branch when the last byte is encountered in a TRANSFER operation.

REAL — extended to allow for BUFFER specification.

RECORDS — used in the TRANSFER statement to specify the number of records to be transferred.

SCRATCH — erases the program in memory; extended to provide for immediate termination of TRANSFER operations.

SCRATCH A — erases all variables, programs and common binaries; extended to provide for immediate termination of TRANSFER operations.

SCRATCH C — erases common area; extended to provide for immediate termination of TRANSFER operations.

TRANSFER — provides for unformatted DMA, interrupt and fast handshake buffer transfers.

WAIT — used in the TRANSFER statement to cause the transfer to execute in serial mode.

WAIT FOR EOR @ — suspends overlapped program execution until a record boundary is reached in a transfer operation.

WAIT FOR EOT @ — suspends overlapped program execution until completion of a transfer operation.

Interface Card Drivers

BCD Driver

Required for the HP 98623A BCD interface.

Datacomm Driver

Required for the HP 98628A Datacomm or HP 98629A Shared Resource Management (SRM) interface. If it is for use with the SRM interface, the Shared Resource Management option is also required.

GPIO Driver

Required for the HP 98622A GPIO interface.

High-Speed Disc Interface Driver

Required for the HP 98625A or HP 98625B High-speed Disc interface.

HP-IB Driver

Required for the internal (built-in) HP-IB interface and for the HP 98624A HP-IB interface; not required for the internal (built-in) disc drives.

Serial Driver

Required for the HP 98626A or HP 98644A Asynchronous Serial interface.

Mass Storage Device Drivers

Bubble Memory Driver

Required for the HP 98259A Magnetic Bubble Memory card.

CS/80 Disc Driver

Required for CS/80 disc drives including subset/CS/80 drives (e.g., HP 9122). The HP-IB driver or High-speed Disc Interface driver is also required.

EPROM Driver

Required for the HP 98253A EPROM Development Kit or HP 98255A EPROM card.

HP 9885 Driver

Required for the HP 9885* Disc drive. The drive is connected to the HP 98622A GPIO interface. The interface requires the GPIO driver.

Small Disc Driver

Required for non-CS/80 external disc drives. The HP-IB or High-speed Disc Interface Driver is also required.

CRT Drivers

Alpha Plane CRT Driver

Required for non-bit-mapped display. This option is automatically loaded when the BASIC system is booted. (If a bit-mapped display is being used, the Bit-mapped Display CRT driver is required instead).

Bit-mapped Display CRT Driver

Required for bit-mapped display. This option is automatically loaded when the BASIC system is booted. (If a non-bit-mapped display is being used, the Alpha Plane CRT driver is required).

* Not supported on all models.

Series 200 and 300 BASIC 4.0 Supported Interfaces and Peripherals

For a list of interfaces and peripherals supported by BASIC 4.0, please see the Series 200 Configuration Information and Order Guide (Part Number 09800-90020) and the Series 300 Configuration Information and Order Guide (Part Number 98561-90020).

BASIC Language Configurations

Three separate products comprise the BASIC 4.0 language offerings for Series 200 and 300 products:

- HP 98613B RAM-based BASIC 4.0 Language System contains BASIC 4.0 plus BASIC Utilities Library on flexible discs.
- HP 98613R Right-to-Reproduce BASIC Language System flexible discs (manuals provided). Option 100 is a delete manuals option for the HP 98613R product.
- HP 98613B Option 001 is an upgrade from a previous version at a reduced price. (Order must include order number from previous version of BASIC).

Utilities

BASIC Utilities Library

Provides media and system management utilities, programming aids, and the loader utility. The loader utility allows unattended loading of BASIC 4.0 system (including a STORED SYSTEM) from a mass storage device when there is more than one system file on that device (an external disc or SRM). The BASIC Utilities Library is shipped with every BASIC 4.0 Language System.

Compiled Subprogram Utilities HP 98613-11X40*

Converts modules compiled (or assembled) in Pascal (or Assembly language) for use as BASIC subprograms. The Compiled Subprogram Utilities must be purchased separately. Compiled subprograms contain the following capabilities and restrictions:

Capabilities

- Ability to call subprograms written in Pascal or Assembly from BASIC.
- Parameter passing from BASIC to the subprogram.
- Code is linkable with subprogram libraries. There is no code size restriction and code is relocatable.
- Compiled subprograms have access to BASIC common space.
- Pascal global variables space is allowed so Pascal global variables can be used.

* The "X" shown in the part numbers specifies the media on which the software is supplied. It can have the following values:

- 2 for use of 5¼-inch, internal flexible discs.
- 4 for use of 3½-inch, double-sided discs recorded single sided.
- 5 for use of 3½-inch, double-sided discs recorded double sided.

- A fixed amount of HEAP space can be specified by the user.
- Simple READ(LN) and WRITE(LN) to the PRINTER IS device can be done.

Restrictions

- Pascal device, file and graphics I/O are not supported.
- Not all Series 200 and 300 Pascal language capabilities are supported. Procedure variables and interrupt service routines are not available.
- The new CSUB 4.0 Utility is required to use these capabilities with BASIC 4.0.

HP 98603A ROM-based BASIC 4.0

BASIC 4.0 is available in ROM for use in applications where a floppy-based operating system would be undesirable. For these applications, such as production test and factory data acquisition and control, ROM makes the system faster to load, easier to install and run and more able to withstand the effects of dust, grit and vibration. There are, however, some performance differences between ROM and RAM systems which should be noted. With BASIC in ROM on a Model 310, the memory accesses to ROM are an average of 26% slower than to the 1 Mbyte on-board RAM system. On a Model 320, ROM BASIC performance is generally comparable to RAM BASIC performance.

ROM-based BASIC 4.0 is fundamentally the same product as RAM-based BASIC 4.0 with the omission of the DataComm, SRM and KNB_2 binaries. These are provided separately on disc.

BASIC 3.0 and 4.0 Series 300 Compatibility

For information on the compatibility between BASIC 3.0 and 4.0 versions, see the Supplement for Series 300 Compatibility, Publication Number 5953-9567. Details on the Display Compatibility Interface can be found in the supplement.

Ordering Information

Description	Part No.
ROM-based BASIC 4.0 Language System	98603A
RAM-based BASIC 4.0 Language System	98613B
Upgrade from previous system†	Opt. 001
5¼-inch media, internal format (formally Opt. 655)	Opt. 042
3½-inch media, single-sided format (formally Opt. 630)	Opt. 044
3½-inch media, double-sided format (formally Opt. 632)	Opt. 045
<i>Note: media option must be specified when ordering 98613B.</i>	
BASIC Right to Reproduce	98613R
Upgrade from previous system†	Opt. 001
Delete manuals	Opt. 100

† The invoice number from the previous BASIC system must be included with the order when specifying Opt. 001.

