Software Product Description

PRODUCT NAME: **VAX BASIC, Version 3.4** **SPD 25.36.20**

DESCRIPTION

VAX BASIC is an interactive, shareable language processor for the VMS Operating System. VAX BASIC takes full advantage of the VAX floating point, decimal and character instructions.

VAX BASIC provides a high-performance program development environment for both applications development and timesharing by generating in-line VMS native mode instructions. It combines the power of a structured programming language with the interactivity of the BASIC environment and the convenience of easy-to-use graphics statements. VAX BASIC is also integrated with various programming productivity tools and key components of the VAX Information Architecture.

In addition to elementary BASIC features, VAX BASIC provides:

- Support for the following Programming Productivity Tools:
  - VMS Symbolic Debugger
  - VAX Performance and Coverage Analyzer
  - Support for the VAX Language-Sensitive Editor (LSE) and VAX Source Code Analyzer (SCA) to provide error diagnostics to VAX LSE, cross reference information for VAX SCA, and support for low level program design, including the processing of pseudocode and the extraction of design information from comments
- Access to the VAX Information Architecture
  - RECORD structure for user-defined data types (similar to PASCAL record types)
  - RECORD structure retrieval from the Common Data Dictionary
- Structured Programming Constructs
  - Line numbers completely optional.
  - DECLARE statement removes requirement for (%) and ($) suffixes.
  - IF...THEN...ELSE...END IF conditional blocks.
  - SELECT...CASE...END SELECT multi-way decision blocks.
  - OTHERWISE out-of-range clause for ON GOTO and ON GOSUB statements.
  - Structured exception handling (WHEN blocks) for main and subprograms.
  - Statement modifiers to control the execution of a single statement.
  - PROGRAM, SUB, FUNCTION and PICTURE statements to declare program modules.
  - END and EXIT PROGRAM statements to return a status to DCL.
- Modern Programming Language Features
  - 31 character alphanumeric statement labels.
  - 31 character variable names, allowing ($), (__) and (.).
  - Explicit declarations provide access to many VAX data types.
  - OPTION statement specifies compiler defaults within source modules.
- Program segmentation
  - SUB, FUNCTION and PICTURE subprograms as individually-compiled modules.
  - Ability to pass parameters BY VALUE, REFerence, or DESCriptor.
  - Up to 254 actual arguments per call on external modules.
  - Ability to invoke EXTERNAL function procedures from BASIC.
  - Recursive CALL/function invocation.
  - Ability to invoke all VMS System Service and Run-Time Library routines.
  - Subprograms and function programs written in other VMS native mode languages can be invoked from VAX BASIC.
  - VAX BASIC program modules can be invoked by other VMS languages.
  - Ability to pass optional arguments to non-BASIC procedures.
  - Single and multi-line user-defined functions using DEF.
  - COMMON and MAP statements for creating static storage areas for communication between program modules.

- Graphics
  - Statements modeled after ANSI BASIC Graphics.
  - Implemented using DEC GKS for VMS to provide device independence.
  - Graphic output primitives include points, lines, areas and text.
  - Graphic input types include points, menu choice, value selection, and text.
  - Ability to adjust the range of coordinate values to suit the application.
  - SET and ASK statements to specify and retrieve graphic attribute values.
  - Graphic procedures (PICTURE subprograms) for building complex objects.
  - Ability to apply transformations to PICTURE procedures.
- Full access to VAX Record Management Services (VAX RMS)
  - Sequential I/O
  - Relative I/O
  - Multi-key Indexed I/O operations, including support for integer (WORD, LONGWORD and QUADWORD), string, segmented string keys, packed decimal keys and descending keys
  - Random access to sequential fixed files
  - Virtual Arrays (arrays mapped onto disk structures)
  - Record File Address (RFA) access for direct access to records
- Extended Report Formatting Capabilities
  - Suppression of zero fields.
  - Zero fill, blank fill, or asterisk fill numeric fields.
  - Commas in large numeric value.
  - CR (credit) or DR (debit) indicators.
  - Floating currency symbol for numeric fields.
  - Currency and decimal symbols can be changed for foreign usage.
  - FORMAT$ function accepts full PRINT USING editing syntax.
- Implicit or Explicit storage declarations
  - Specification of data types permitted on COMMON, DECLARE, DEF, DIMENSION, EXTERNAL, FUNCTION, MAP, PICTURE, RECORD and SUB statements.
  - Default data allocation rules can be specified with DCL qualifiers, BASIC commands, or by the OPTION statement in program text.
  - By default all declarations are implicit, however, the option TYPE= EXPLICIT can be used to require explicit declaration of all variables.
  - Default constant types can be specified with the OPTION CONSTANT TYPE statement.
  - Integer data type allows:

    BYTE (8 bit) with range of -128 to +127

    WORD (16 bit) with range of -32768 to +32767

    LONG (32 bit) with range of -2147483648 to +2147483647
  - Real data type allows:

    SINGLE (6 digits) with range of .29x10**-38 to 1.70x10**38

    DOUBLE (16 digits) with range of .29x10**-38 to 1.70x10**38

    GFLOAT (15 digits) with range of .56x10**308 to .90x10**308

    HFLOAT (33 digits) with range of .84x10**-4932 to .59x10**4932
    - Packed DECIMAL type supports up to 31 digits.
    - STRING data type, allowing both static (in MAP or COMMON statements) and dynamic lengths.
    - Creation of user-defined named constants through DECLARE CONSTANT.
    - Dynamic record definition and variable allocation via MAP DYNAMIC.
- BASIC Programming Support Environment
  - RUN command for immediate execution of BASIC programs.
  - EDIT command can invoke a user-selected editor directly.
  - HELP for on-line assistance.
  - SEQUENCE command for generating line numbers.
  - RESEQUENCE command for renumbering program lines.
  - Direct execution of BASIC statements (immediate mode).
  - Direct execution of DEC Command Language (DCL) statements.
  - Optional Line-by-Line syntax checking.
  - Dynamic linking (LOAD) of separately-compiled BASIC modules for use with the RUN command.
  - User-created libraries can be searched automatically when using RUN.

- Compile-Time Directives
  - Text inclusion through %INCLUDE, %INCLUDE %FROM %CDD and %INCLUDE from a text library
  - Conditional compilation (%IF)
  - Listing and cross-reference output control (%NOLIST, %LIST, %CROSS)
- EXTERNAL statement - allows access to global variables, functions, and constants, and allows data typing of parameters to aid in minimizing run-time mismatches.
- Language Subsets and Subset Flaggers
  - BASIC-PLUS-2 Subset Flagger for cross-system development
  - Declining Feature Flagger for program maintenance/conversion
  - Qualifier for ANSI Minimal BASIC conforming program execution
- Multi-line Statements and Multi-statement Lines
- Extensive array handling capabilities
  - Each array may have up to 32 dimensions.
  - Each dimension may specify both a lower and upper bound.
  - Array bounds can be specified at compile-time or run-time.
  - Matrix handling statements allow manipulation of matrices, including matrix multiplication.
- Compatibility with key RSTS/E BASIC-PLUS and BASIC-PLUS-2 features including:
  - ON ERROR GOTO exception handling
  - FIELD Statement
  - CVT and SWAP% functions
  - Virtual arrays
  - Selected nonprivileged SYS calls
  - Statement modifiers for conditionals and loops

VAX BASIC uses the full printable ASCII character set, and 8-bit character codes within constants and I/O operations.

Standard Conformance

ANSI Minimal BASIC Validated, February 1989

HARDWARE REQUIREMENTS

VAX, MicroVAX, VAXserver or VAXstation configuration as specified in the System Support Addendum (SSA 25.36.20-x).

SOFTWARE REQUIREMENTS

For Systems Using Terminals: (No DECwindows Interface):

- VMS Operating System

For Workstations Running VWS:

- VMS Operating System
- VMS Workstation Software

For Workstations Running DECwindows:

- VMS Operating System (and necessary components of VMS DECwindows)

VAX BASIC supports a DECwindows compiler interface for compile qualifier selection on workstations. This interface is accessible from FileView.

Refer to the System Support Addendum (SSA 25.36.20-x) for availability and required versions of prerequisite/optional software and for information regarding components of VMS DECwindows.

ORDERING INFORMATION

Software Licenses: QL-095A*-**
Software Media: QA-095A*-**
Software Documentation: QA-095AA-GZ
Software Product Services: QT-095A*-**

* Denotes variant fields. For additional information on available licenses, services and media, refer to the appropriate price book.

SOFTWARE LICENSING

This software is furnished under the licensing provisions of Digital Equipment Corporation's Standard Terms and Condition.

For more information about Digital's licensing terms and policies, contact your local Digital office.

LICENSE MANAGEMENT FACILITY SUPPORT

This layered product supports the VMS License Management Facility.

License units for this product are allocated on a CPU-capacity basis.

For more information on the License Management Facility, refer to the VMS Operating System Software Product Description (SPD 25.01.xx) or the License Management Facility manual of the VMS Operating System documentation set.

For more information about Digital's licensing terms and policies, contact your local Digital office.

SOFTWARE PRODUCT SERVICES

A variety of service options are available. For more information, contact your local Digital office.

SOFTWARE WARRANTY

Warranty for this software product is provided by Digital with the purchase of a license for the product as defined in the Software Warranty Addendum of this SPD.