



Software Product Description

Product Name: VSI BASIC for OpenVMS

DO-DBASPD-01A

DESCRIPTION

This Software Product Description describes VSI BASIC Version 1.8 for OpenVMS.

VSI BASIC is a shareable language processor for the OpenVMS operating system. It can be invoked as a compiler and the resulting modules can be linked and run with standard OpenVMS commands. VSI BASIC provides high performance for application development by generating inline native mode instructions. VSI BASIC is also integrated with various programming productivity tools such as the OpenVMS Debugger, the Language-Sensitive Editor (LSE), the Source Code Analyzer (SCA) and the Performance and Coverage Analyzer.

VSI BASIC includes three IEEE floating-point data types (SFLOAT, TFLOAT, and XFLOAT) and a QUAD integer data type.

FEATURES

VSI BASIC is integrated into the Common Language Environment. This integration provides VSI BASIC users with:

- Support for OpenVMS inter-language calling standard
- Access to all OpenVMS system services
- Access to the facilities of the OpenVMS Debugger
- Callable interfaces to the OpenVMS Run-Time Library
- Support for the Language-Sensitive Editor
- Structured programming constructs
 - Line numbers completely optional.
 - DECLARE statement that 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 sub-programs.
 - Statement modifiers to control the execution of a single statement.
 - PROGRAM, SUB, and FUNCTION statements to declare program modules.
 - END and EXIT PROGRAM statements to return a status to DCL.

VSI BASIC for OpenVMS

- Modern programming language features
 - Uses the full printable ASCII character set and 8-bit character codes within constants and I/O operations.
 - 31 character alphanumeric statement labels.
 - 31 character variable names, allowing (\$), (_) and (.).
 - RECORD structure for user-defined data types (similar to PASCAL record types), including RECORD structure retrieval from the Common Data Dictionary.
 - OPTION statement specifying compiler defaults within source modules.
- Program segmentation
 - SUB and FUNCTION 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 VSI BASIC.
 - Recursive CALL/function invocation.
 - Ability to invoke OpenVMS System Service and Run-Time Library routines.
 - Ability to invoke subprograms and function programs written in other OpenVMS native mode languages.
 - VSI BASIC program modules invoked by other OpenVMS languages.
 - Ability to pass optional arguments to non-BASIC procedures.
 - Single and multiple line user-defined functions using DEF.
 - COMMON and MAP statements for creating static storage areas for communication between program modules.
- Full access to OpenVMS Record Management Services (RMS)
 - Sequential I/O
 - Relative I/O
 - Multiple 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 values.
 - CR (credit) or DR (debit) indicators.
 - Floating currency symbol for numeric fields.
 - Currency and decimal symbols changed for foreign usage.
 - FORMAT\$ function accepting full PRINT USING editing syntax.

VSI BASIC for OpenVMS

- Implicit or explicit storage declaration
 - Specification of data types permitted on COMMON, DECLARE, DEF, DIMENSION, EXTERNAL, FUNCTION, MAP, RECORD and SUB statements.
 - Default data allocation rules specified with DCL qualifiers, VSI BASIC commands, or by the OPTION statement in program text.
 - By default all declarations implicit, however, the OPTION TYPE = EXPLICIT can be used to require explicit declaration of all variables.
 - Default constant types specified with the OPTION CONSTANT TYPE statement.
 - INTEGER data type including:
 - BYTE (8 bit)
 - WORD (16 bit)
 - LONG (32 bit)
 - QUAD (64 bit)
 - REAL data type including:
 - SINGLE (6 digits)- I64 BASIC uses SFLOAT for computations
 - DOUBLE (16 digits)- Alpha BASIC uses GFLOAT for computations. I64 BASIC uses TFLOAT for computations
 - GFLOAT (15 digits)
 - SFLOAT (6 digits)-an IEEE data type
 - TFLOAT (15 digits)-an IEEE data type
 - XFLOAT (33 digits)-an IEEE data type
 - Packed DECIMAL type supporting 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.
 - Compile-time directives
 - Text inclusion through %INCLUDE, %INCLUDE %FROM %CDD and %INCLUDE from a text library
 - Conditional compilation (%IF)
 - Listing and output control (%NOLIST, %LIST)
 - Cross-reference output control (%CROSS)
 - Text replacement (%DEFINE)
 - EXTERNAL statement allowing access to global variables, functions, and constants, and allowing data typing of parameters to aid in minimizing run-time mismatches.
 - Motif bindings support
 - Multiple line statements and multiple statement lines

VSI BASIC for OpenVMS

- 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
- VSI BASIC Translator for OpenVMS Alpha only

HARDWARE AND SOFTWARE REQUIREMENTS

VSI supports VSI BASIC on HPE systems running these VSI operating systems:

- Any HPE Integrity system supported on VSI V8.4-2, VSI V8.4-2L1, or later.
- Any HPE AlphaServer system supported on VSI V8.4-2L1 or later.

DISK SPACE REQUIREMENTS

For Integrity Systems:

- Disk space required for installation: 53,500 blocks (27.4 MB)
- Disk space required for installation: 45,500 blocks (23.0 MB)

For Alpha systems:

- Disk space required for installation: 32,500 blocks (16.6 MB)
- Disk space required for use (permanent): 24,500 blocks (12.5 MB)

These counts refer to the disk space required on the system disk. The sizes are approximate; actual sizes may vary depending on the user's system environment, configuration, and software options.

OPTIONAL HARDWARE

Floating-point intensive applications should be run on configurations with the appropriate hardware support for the floating-point data types being used.

OPTIONAL SOFTWARE

Refer to the appropriate Software Product Description for details and requirements on the products listed below.

- Language-Sensitive Editor component of the Language-Sensitive Editor/Source Code Analyzer (LSE/SCA)
- Performance and Coverage Analyzer (PCA)

Required to fully use the /ANALYSIS_DATA qualifier:

- Source Code Analyzer component of Language-Sensitive Editor/Source Code Analyzer

VSI BASIC for OpenVMS

Language-Sensitive Editor/Source Code Analyzer, Performance and Coverage Analyzer are available as part of DECset for OpenVMS Systems.

To use %INCLUDE %FROM %CDD or the %REPORT %DEPENDENCY directives and the /DEPENDENCY_DATA qualifier:

- Oracle CDD/Repository[TM]

CLUSTER ENVIRONMENT

This layered product is fully supported when installed on any valid and licensed OpenVMS Cluster configuration without restrictions.

ORDERING INFORMATION

VSI BASIC for OpenVMS licenses are available as electronic licenses (E-LTU) or physical licenses (P-LTU). Contact your VSI sales office for more information.

For VSI BASIC on HPE Integrity Systems:

- VSI BASIC for VMS I64 Concurrent E-LTU SL-LIBA0E-17V
- VSI BASIC for VMS I64 Concurrent P-LTU SL-LIBA0P-17V

For VSI BASIC on HPE AlphaServer Systems:

- VSI BASIC is included in the ALPHA-LP license

SOFTWARE LICENSING

This software is furnished under the licensing provisions of VMS Software, Inc. Standard Terms and Conditions. A software license is required in order to use VSI BASIC software. For I64, it is offered with Concurrent Use licenses. Version update licenses are not available for the I64 platform. Rights to use future revisions of VSI BASIC are available only through a Support Agreement or through a new license purchase. For more information about OpenVMS license terms and policies, contact your local VSI sales office. Information is also available at the following website:
<http://vmssoftware.com/services>

LICENSE MANAGEMENT FACILITY SUPPORT

VSI BASIC for OpenVMS supports the OpenVMS License Management Facility (LMF). For more information about the License Management Facility, refer to the *VSI OpenVMS License Management Utility Manual*.

SOFTWARE PRODUCT SERVICES

A variety of service options are available from VSI. For more information, contact your local VSI account representative or distributor. Information is also available at the following website: <http://vmssoftware.com/services>

SOFTWARE WARRANTY

This software product is provided by VSI with a 90-day conformance warranty in accordance with the VSI warranty terms applicable to the license purchase.

Copyright © 2017 VMS Software, Inc., Bolton Massachusetts, USA

Confidential computer software. Valid license from VSI required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for VSI products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. VSI shall not be liable for technical or editorial errors or omissions contained herein.