Software Product Description

PRODUCT NAME: VAX Language-Sensitive Editor/Source Code Analyzer, Version 3.1 **SPD 26.59.11**

DESCRIPTION

The VAX Language-Sensitive Editor/Source Code Analyzer is an integrated environment that provides software professionals with editing, cross-reference, and static analysis capabilitites. It is a multi-language, multi-module, and multi-window tool that is useful for program design, development, and maintenance.

The two main components of this product are the Language-Sensitive Editor and the Source Code Analyzer. The Language-Sensitive Editor component provides the following capabilities: source code templates, the ability to do low level program design by embedding pseudocode in source code, the ability to view source code at various levels of detail by replacing a sequence of source lines with a single overview line, and access to on-line language-specific help. Together, this support enables new and experienced programmers to develop programs faster through VAX language-specific construct completion and error detection and correction facilities.

The Source Code Analyzer component aids developers in understanding the complexities of software systems. Since it allows a developer to analyze an entire system as opposed to individual modules, and helps the developer understand unfamiliar systems, the Source Code Analyzer component is extremely useful during the implementation and maintenance phases of a project. The Source Code Analyzer component provides source code navigation, cross-reference, and static analysis for compile-time source code information generated by supported VAX languages. Additionally, the Analyzer provides design cross-reference, analysis, and reports for pseudocode and specialized comment processing information generated by supported VAX languages.

The Language-Sensitive Editor component supports the DECwindows software environment, and provides either a DECwindows or a character cell terminal interface. All the Source Code Analyzer commands are available from within an editing session.

The VAX Language-Sensitive Editor/Source Code Analyzer is included in the VAXset Software Engineering Tools Package (SPD 27.07.xx). This package provides simplified ordering and maintenance of various tools. See SPD 27.07.xx for more information on VAXset.

The VAX Language-Sensitive Editor/Source Code Analyzer works in concert with supported VAX languages and the VAX Symbolic Debugger to provide a highly interactive, on-line environment that facilitates the EDIT-COMPILE-DEBUG-NAVIGATE-ANALYZE portion of the program development cycle. Within a single editing session, users can write, edit, compile, review compile-time errors, correct errors, navigate through, and analyze their source code.

The Language-Sensitive Editor component can be invoked directly from the VAX Symbolic Debugger to correct source code errors found during a debugging session.

Users can customize the environment by tailoring and expanding upon the features and structures provided by the Language-Sensitive Editor component. User-defined environments can be saved for future use.

"Language-Sensitive" Features

Source Code Templates for Supported VAX Languages

For each supported VAX language, the Language-Sensitive Editor component provides a set of source code templates. These templates are formatted language constructs that provide keywords, punctuation, and placeholders. Templates are inserted into the editing buffer by successive expansions of tokens and placeholders. Placeholders represent positions in the source code where the user must provide additional program text or choose from indicated syntactic options. Tokens are keywords, or function names, that the user can type into the editing buffer and expand into templates for corresponding source constructs.

Support for Pseudocode Entry

The Language-Sensitive Editor component supports the use of pseudocode for low level program design. Users can mix free text with programming language syntax by enclosing the text in pseudocode delimiters. Programs in this design state may be processed by supporting compilers to produce .ANA files for the Source Code Analyzer component. Users can also convert pseudocode entries into program comments.

Support for Viewing Code at Various Levels

The Language-Sensitive Editor component also includes a code elision feature which is a facility for viewing source code at various levels of detail. A sequence of source lines may be hidden and replaced by a single overview line using the COLLAPSE command. The overview line consists of text taken from the first hidden source line. This text is enclosed in pseudocode delimiters. The EXPAND command restores the lines hidden by a COLLAPSE command. The FOCUS command allows users to collapse the entire buffer except for the nested groups that contain the current line.

VAX language-specific on-line help for placeholders and key language concepts is accessible from the Editor.

Support For Vector Software Development

The Language-Sensitive Editor component supports vector software development. When used in conjunction with the VAX FORTRAN High-Performance Option, the Editor will report source code dependencies in vectorizable arrays and suggest where to add ASSERT statements to the VAX FORTRAN source code to minimize source code dependencies.

Source Code Analysis Capabilities

The Source Code Analyzer component's query capability allows the user to search and display specific source code symbol, file and module information based on certain attributes and their relationships.

The Source Code Analyzer component can help answer some commonly asked source code questions such as:

— Where is a particular global variable declared, initialized, modified, or referenced?

— In which modules are there references to a particular symbol name?

— Are my calling arguments consistent?

— Where is a certain routine called?

VAX Language Interfaces

The Language-Sensitive Editor component interfaces to supported VAX languages (refer to List 1 for supported VAX languages) to provide the capability for on-line, interactive program editing, compilation, compile time error reporting, and compile time error correction within a single editing session. Supported VAX languages provide a /DIAGNOSTICS qualifier that instructs the compiler to generate a file of compile-time diagnostic information. This file is used by the Editor in compilation error review mode.

The Source Code Analyzer component interfaces to supported VAX languages (refer to List 1 for supported VAX languages) for source code navigation, cross- reference, and static analysis. Supported VAX languages provide an /ANALYSIS qualifier that instructs the compiler to generate a file of compile-time source code information. This file is used by the Source Code Analyzer component for source code cross-reference, navigation and analysis. In addition, supported VAX languages (see List 4) provide a /DESIGN qualifier that instructs the compiler to generate pseudocode information. This information is used by the Source Code Analyzer component to provide design cross reference, analysis, and reports.

User Diagnostic File Interface

The Language-Sensitive Editor component has an ASCII format diagnostic file interface to be used with compilers and tools other than those supplied by Digital. The file is used to communicate diagnostic messages to the Language-Sensitive Editor component and allows the user to display the diagnostic messages in the REVIEW buffer and issue a GOTO SOURCE command to display the corresponding source in another window.

The ASCII format of this file provides an easy to use Digital Command Language (DCL) style interface.

Features include:

- COMPILE command to perform compilations without leaving the Editor, and, optionally, to review compilation errors upon compile completion. The user may specify DCL qualifiers such as /DEBUG and /LIBRARY when invoking the compiler from the Language-Sensitive Editor component. The compilation may be performed in a BATCH job.

- REVIEW mode to review compilation errors upon compile completion. The Language-Sensitive Editor component displays the compilation errors in one window, with the corresponding source code displayed in a second window. For easy error correction, there is GOTO SOURCE command to position the cursor at the point in the source code where the compiler detected the error.

VAX DEC/CMS (Code Management System) Interface

The Language-Sensitive Editor component provides an interface to the VAX DEC/CMS (Code Management System) [SPD 25.52.xx] to simplify program development and source file management. All VAX DEC/CMS functions are available from within the Editor.

VAX DEC/CMS can be utilized to store program source files. These files can be moved from the CMS library, placed in the current editing buffer, modified, and returned to the CMS library without leaving the Editor.

VAX Symbolic Debugger Interface

The Language-Sensitive Editor component can be invoked from the VAX Symbolic Debugger providing the ability to make source code corrections as they are found during a debugging session.

Features include:

- User notification if the file, invoked by the Editor, is a different version than that displayed in the VAX Symbolic Debugger
- Ability to specify the file and line number from which to start the editing session with the default being the current source displayed in the VAX Symbolic Debugger
- User choice of terminating activity directly from the editing session or returning to the debugging session

User Interface

The Language-Sensitive Editor component offers an easy-to-use, flexible user interface designed to meet varying user needs.

Features include:

- Default KED/EDT keypad layout
- Alternate VAXTPU EVE keypad layout
- Multiple windows, multiple buffers
- Choice of editing mode: OVERSTRIKE or INSERT
- Ability to bind any command (user-defined or those supplied by the Language-Sensitive Editor component), or sequence of keystrokes (a "learn" sequence) to a key.
- On-line HELP facility for the Language-Sensitive Editor component
- Interface to VAX DEC/CMS. The Editor can use VAX DEC/CMS protocol to access files stored in the current CMS library.
- Source File Management features include defining a searchlist for locating source files and marking source directories as read-only to prevent accidental modifications.

Language-Sensitive Editor User-Tailorable Features

Language-Sensitive Editor component users can tailor the editing environment to address unique programming needs. These user-defined environments can be saved for future editing sessions.

Users can provide a definition of a language, memo header, or other textual templates for the Editor. To address specific coding conventions, users can add to or change the definitions supplied for the supported languages.

VAXTPU (Text Processing Utility)

For more unique editing requirements, the Language-Sensitive Editor component provides commands to access VAXTPU, a text processing utility which is part of the VMS Operating System. VAXTPU has an easy-to-use, high-level procedural language which allows users to write functions incremental to those provided by the Language-Sensitive Editor component. The VAXTPU language provides for looping and conditionals to allow users to perform more powerful editing tasks.

VAXTPU Language statements include:

- The assignment statement
- Procedural statements (PROCEDURE - ENDPROCEDURE)
- Looping statements (LOOP - ENDLOOP)
- Conditional statements (IF - THEN - ELSE - ENDIF)
- Case statements (CASE - ENDCASE)
- Error statements (ON_ERROR - ENDON_ERROR)

Language-Sensitive Editor Component Commands

Users control the editing environment and tap the Language-Sensitive Editor component's knowledge of VAX languages with a rich set of commands. Commands used most often can be easily bound to keys to enable quick and efficient programming. All commands are available through "command mode".

Command features include:

- EXPAND/UNEXPAND, ERASE/UNERASE PLACEHOLDER, commands for language construct completion
- CHANGE WINDOW_MODE, NEXT/PREVIOUS WINDOW, SET SCREEN, SHIFT, REFRESH ONE WINDOW, TWO WINDOWS, OTHER WINDOW, SPLIT WINDOW, ENLARGE WINDOW, SHRINK WINDOW, DELETE WINDOW screen manipulation commands
- COMPILE command to invoke the appropriate VAX language compiler
- REVIEW, END REVIEW, NEXT ERROR, PREVIOUS ERROR, GOTO SOURCE commands for compile-time error review and correction
- DEFINE/DELETE [LANGUAGE, TOKEN, PLACEHOLDER, ALIAS, COMMAND, KEY, PACKAGE, PARAMETER, ROUTINE, ADJUSTMENT, KEYWORD, TAG], EXTRACT, MODIFY LANGUAGE, SET MODE commands for tailoring the user environment
- SAVE ENVIRONMENT command for writing user-defined language environment to a file for future editing session use
- SAVE SECTION command for writing user-defined editing environment to a file for future editing session use
- DO command to execute an Editor command (or string of commands) or VAXTPU program statements
- CALL command for calling a VAXTPU procedure
- SPAWN and ATTACH commands for suspending the editing session to switch to a subprocess running the DCL command interpreter
- GOTO command to position the cursor at the indicated option (options include BOTTOM, TOP, BUFFER, FILE, MARK, PAGE, LINE, WORD, CHARACTER, PLACEHOLDER, SCREEN) as well as CANCEL MARK, CHANGE DIRECTION commands for cursor positioning
- SEARCH command for locating specified characters or strings (includes wildcard option)
- SUBSTITUTE command for replacing the text of one string with another
- SHOW command to display characteristics and settings of several options (including ALIAS, BUFFER, COMMAND, DIRECTORY, KEY, LANGUAGE, MARK, MODE, PACKAGE, PARAMETER, PLACEHOLDER, ROUTINE, SCREEN, SEARCH, SOURCE_DIRECTORY, TOKEN, VERSION, ADJUSTMENT, KEYWORD, TAG)
- Text manipulation commands include CANCEL SELECT_MARK, CAPITALIZE WORD, CHANGE [CASE, DIRECTION, INDENTATION, TEXT_ENTRY_MODE], CUT, ENTER [LINE, SPACE, SPECIAL, TAB, TEXT], ERASE [CHARACTER, LINE, PLACEHOLDER, WORD], FILL, LOWERCASE WORD, PASTE, SET [FORWARD, INSERT, OVERSTRIKE, REVERSE, SELECT_MARK, WRAP, LEFT_MARGIN, RIGHT_MARGIN], SUBSTITUTE, TAB, UNERASE SELECTION, UNEXPAND, UNTAB, UPPERCASE WORD.
- Pseudocode manipulation commands include ENTER [PSEUDOCODE, COMMENT], UNDO ENTER COMMENT.
- Commands to manipulate overviews include VIEW SOURCE, FOCUS, COLLAPSE, EXPAND.
- CHANGE INDENTATION, ENTER TAB, SET INDENTATION, SET TAB_INCREMENT, TAB, UNTAB commands for source code indentation control
- DELETE BUFFER, SET [MODIFY, NOMODIFY, DIRECTORY, OUTPUT_FILE, SOURCE_DIRECTORY, OVERVIEW, NOOVERVIEW], WRITE, RESERVE, UNRESERVE, REPLACE, CLOSE, READ, INCLUDE commands for manipulation of files, editing buffers, and interfacing to VAX DEC/CMS
- CHECK LANGUAGE command which finds likely errors in user created language definitions.
- FILL and ALIGN commands for source code comment formatting
- All VAX DEC/CMS commands

Source Code Analyzer Commands

Query Capabilities

FIND command provides interactive cross-reference of symbol and source file information. It provides the following capabilities:

- Locate symbols and occurrences of symbols.
- Limit the query to a specified set of names or partial names where wildcards are permitted.
- Limit the query based on symbol selection attributes.
- Limit the query based on occurrence selection attributes.
- Limit the query based on domain selection attributes.
- Locate symbols and occurrences based on various relationships to other symbols or occurrences.

INSPECT command provides interactive cross-module checking of routines and common blocks for consistency. It provides the following capabilities:

- Consistency of types
- Check for unused, incorrectly used and implicitly declared variables
- Tailor the extent to which INSPECT checks for errors.

REPORT command provides customizable generation of REPORTS, including the generation of Section 4 of the DOD-STD-2167A Software Design Document.

Library Management

- CREATE LIBRARY command initializes a Source Code Analyzer library.
- LOAD command loads the compiler-generated information into a Source Code Analyzer library.
- SET LIBRARY command selects the libraries to be used during the current analyzer session. The library may be on a remote node.
- SHOW LIBRARY command displays the directory specification of the current active library.
- REORGANIZE command optimizes the size and organization of the library.
- VERIFY command checks the integrity of the data in a library.
- SHOW MODULE command displays current module information including module name, identifier, and compilation date.
- DELETE MODULE command deletes specified module of source data from library.

Query Navigation (when used with the Language-Sensitive Editor component)

The user can navigate through source code information and directly edit the source code.

NEXT and PREVIOUS commands move to the appropriate option displayed in the query buffer as follows:

- SYMBOL specifies a symbol or file associated with the current name.
- OCCURRENCE specifies an occurrence of the current item associated with the current name.

GOTO SOURCE command displays the source code that corresponds to the current symbol occurrence.

GOTO DECLARATION command displays the source code that corresponds to the declaration of the specified symbol.

Query Management

Allows the user to maintain all previously entered queries, to reuse them, or to use them in new queries.

- NEXT, PREVIOUS and GOTO QUERY commands allow users to select different queries.
- SHOW QUERY command displays information on all queries in the current session.
- Delete query command deletes the specified query.

Generating Design Reports

In addition to getting information directly from Source Code Analyzer component queries, users can produce a variety of reports based on their design in the Source Code Analyzer component database.

Typically, reports cover all or a designated part of the user's Source Code Analyzer component database and present information in a structured, organized way. The REPORT command allows users to generate a report type provided by the Source Code Analyzer component or to create a customized report.

The following reports are provided by the Source Code Analyzer component:

- HELP - A VMS Help file generated from the user's design or code.
- PACKAGE - An LSE package definition.
- INTERNALS - A general report that formats a user's entire design in an organized manner.
- 2167A_DESIGN - A report that produces a document that meets the Section 4 requirements of the U.S. Defense Department's DOD-STD-2167A Software Design Document.

Note: Both the Language-Sensitive Editor and Source Code Analyzer components of the VAX Language Sensitive Editor/Source Code Analyzer V3.1 must be on the user's system to generate reports.

Source Code Analyzer Callable Interface

A set of routines allows the user to use the Source Code Analyzer component from within independent application programs.

Although full Source Code Analyzer component functionality is provided by its integration with the Language-Sensitive Editor component, including the editing of the corresponding source code, most Source Code Analyzer component features are available directly from the DCL command level interface.

HARDWARE REQUIREMENTS

VAX, MicroVAX, VAXstation or VAXserver configuration as specified in the System Support Addendum (SSA 26.59.11-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)

Note: For full use of the Language-Sensitive Editor component features described in this SPD, at least one of the VAX languages in List 1 or List 2 is required. For full use of the Source Code Analyzer component features described in this SPD, at least one of the VAX languages in List 1, or VAX SCAN from List 2, and the Language-Sensitive Editor component are required. (The Language-Sensitive Editor component is not required to initially install the Source Code Analyzer component.)

Note: Certain features of this product are not available when using language pre-processors such as those supplied for VAX Rdb/VMS and VAX DBMS. Consult your local Digital representative for additional information.

List 1

The following list specifies VAX languages which are fully supported by the VAX Language-Sensitive Editor/Source Code Analyzer. Templates for those languages are included with the Language-Sensitive Editor component.

- VAX Ada
- VAX BASIC
- VAX BLISS-32 Implementation Language
- VAX C
- VAX COBOL
- VAX FORTRAN
- VAX MACRO
- VAX PASCAL
- VAX PL/I

List 2

The following list specifies VAX languages which are fully supported by the Language-Sensitive Editor component. Templates for these languages are included with the product and not the Language-Sensitive Editor component.

- VAX CDD/Plus
- VAX DIBOL
- VAX DOCUMENT
- VAX Rdb/VMS
- VAX SCAN
- VAXELN PASCAL

List 3

The following list specifies VAX products which have partial support for the Language-Sensitive Editor component (Refer to product SPD for more details). Templates for these products are included with the product and not the Language-Sensitive Editor component.

- VAX ACMS
- VAX DATATRIEVE
- DECwindows UIL
- VAX TDMS

List 4

The following list specifies VAX products which process pseudocode entry information for the Source Code Analyzer component.

- VAX Ada
- VAX BASIC
- VAX BLISS-32 Implementation Language
- VAX C
- VAX COBOL
- VAX FORTRAN
- VAX PASCAL

Refer to the System Software Addendum for availability and required versions of prerequisite/optional software (SSA 26.59.11-x) and for information regarding components of VMS DECwindows.

ORDERING INFORMATION

Software Licenses: QL-057A*-**
Software Media: QA-057A*-**
Software Documentation: QA-057AA-GZ
Software Product Services: QT-057A*-**

* 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 Conditions.

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.

™ The DIGITAL Logo, VMS, DECwindows, VAX, MicroVAX, VAXstation, VAXserver and VAXset are trademarks of Digital Equipment Corporation.