Compaq DECset for OpenVMS

Compaq Language-Sensitive Editor for OpenVMS Version 4.7 Release Notes

August 2000

These release notes for the Compaq Language-Sensitive Editor for OpenVMS Version 4.7 software contain important general information, new and changed features, current restrictions, and documentation notes.

Operating System Version:	Compaq OpenVMS VAX Operating System, Versions 6.2, 7.1, 7.2 Compaq OpenVMS Alpha Operating System, Versions 6.2, 7.1, 7.2-1 Compaq DECwindows Motif for OpenVMS, Versions 1.2-4, 1.2-5
Software Version:	Compaq DECset for OpenVMS, Version 12.4 Compaq Language-Sensitive Editor for OpenVMS, Version 4.7

Compaq Computer Corporation
Houston, Texas

© 2000 Compaq Computer Corporation
© Electronic Data Systems Corporation 2000

COMPAQ, VAX, VMS, the Compaq logo, and the DIGITAL logo Registered in U.S. Patent and Trademark Office.

DEC Ada, DECforms, DECnet, DECset, DECwindows, eXcursion, OpenVMS, and PATHWORKS are trademarks of Compaq Information Technologies Group, L.P.

Microsoft, Visual C++, Windows, Windows 95, Windows 98, Windows 2000, and Windows NT are trademarks of Microsoft Corporation.

Motif is a registered trademark of The Open Group.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc.

All other product names mentioned herein may be the trademarks or registered trademarks of their respective companies.

Confidential computer software. Valid license from Compaq, EDS, or authorized sublicensor 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.

Neither Compaq nor EDS shall not be liable for technical or editorial errors or omissions contained herein.

The information in this publication is subject to change without notice and is provided "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK ARISING OUT OF THE USE OF THIS INFORMATION REMAINS WITH RECIPIENT. IN NO EVENT SHALL COMPAQ OR EDS BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL, PUNITIVE, OR OTHER DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, OR LOSS OF BUSINESS INFORMATION), EVEN IF COMPAQ OR EDS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE FOREGOING SHALL APPLY REGARDLESS OF THE NEGLIGENCE OR OTHER FAULT OF EITHER PARTY AND REGARDLESS OF WHETHER SUCH LIABILITY SOUNDS IN CONTRACT, NEGLIGENCE, TORT, OR ANY OTHER THEORY OF LEGAL LIABILITY, AND NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

The limited warranties for Compaq and EDS products are exclusively set forth in the documentation accompanying such products. Nothing herein should be construed as constituting a further or additional warranty.

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

Preface v

1 General Information

1.1 Introduction 1–1
1.2 Associated Documents 1–1
1.3 Installation Requirements 1–2
1.4 Compatibility Issues 1–2

2 New Features

2.1 Box Commands 2–1
2.2 Command to Delete Overview Information 2–3
2.3 COBOL Language Template 2–3
2.4 Example Language Extensions 2–3
2.5 TPU Pattern Style 2–3
2.5.1 User Interface 2–4
2.5.2 Partial Pattern Assignment Variables 2–5
2.5.3 New Line 2–6
2.5.4 Errors 2–6
2.5.5 Global Variables 2–7
2.5.6 Pattern Variables 2–7
2.5.7 Use for Developing DTM User Filters 2–7

3 Problems Fixed

4 Known Problems

4.1 General 4–1
4.2 Documentation 4–6

5 Restrictions

6 Reporting Problems and Suggestions

Preface

These release notes characterize the Version 4.7 release of the *Compaq Language-Sensitive Editor for OpenVMS (LSE)* software. These notes contain information that either supplements or supersedes the information in the Compaq DECset for OpenVMS documentation set.

Intended Audience

These release notes are intended for users and system managers who need information about LSE on either of the Compaq OpenVMS Operating System platforms (VAX or Alpha).

Document Structure

These release notes are organized as follows.

- Chapter 1 contains general information about this release.
- Chapter 2 contains a description of the new features.
- Chapter 3 describes all problems resolved by the current release.
- Chapter 4 describes all outstanding issues in the current release.
- Chapter 5 describes all restrictions.
- Chapter 6 describes how to report problems.

Sending Compaq Your Comments

We welcome your comments on these release notes or on any other pieces of documentation. See Chapter 6 for details on how to contact Compaq with your comments.

Conventions

The following conventions are used in these notes.

Convention	Description
$	A dollar sign ($) represents the OpenVMS DCL system prompt.
Ctrl/*x*	The key combination Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key, for example, Ctrl/Y or Ctrl/Z or a pointing device button.
boldface text	Boldface text represents the introduction of a new term.
`monospace boldface text`	Boldface monospace text represents user input in interactive examples.
italic text	Italic text represents book titles, parameters, arguments, and information that can vary in system messages (for example, Internal error *number*).
UPPERCASE	Uppercase indicates the name of a command, routine, the name of a file, the name of a file protection code, or the abbreviation of a system privilege.
lowercase	Lowercase in examples indicates that you are to substitute a word or value of your choice.

1
General Information

The information in these release notes pertain to the Compaq Language-Sensitive Editor for OpenVMS (LSE), Version 4.7 software. Compaq suggests that you upgrade your environment incrementally with each new release, but there is no technical requirement to do so. It is recommended that you review the release notes for each version because they capture important release-oriented information and advise you of changes between versions.

1.1 Introduction

LSE Version 4.7 is a minor release.

1.2 Associated Documents

The following documents contain information related to LSE. Note however, that for the Compaq DECset for OpenVMS Version 12.4 release, only the software product descriptions, installation guide, and component product release notes have been updated.

The documents associated with LSE are:

- *Guide to DIGITAL Language-Sensitive Editor for OpenVMS Systems*
- *DIGITAL Language-Sensitive Editor/Source Code Analyzer for OpenVMS Reference Manual*

Also applicable are the general DECset documents:

- *Compaq DECset for OpenVMS Installation Guide*
- *Compaq DECset for OpenVMS Alpha Software Product Description*
- *Compaq DECset for OpenVMS VAX Software Product Description*
- *Guide to Detailed Program Design for OpenVMS Systems*
- *Using DECset for OpenVMS Systems*
- *Guide to DIGITAL Source Code Analyzer for OpenVMS Systems*

In addition to the base LSE documentation, the DECset engineering team has also provided the *Compaq DECset for OpenVMS Cookbook for an Integrated Product Development Environment*. This document comprises a series of technical articles that describe how each of the component applications can be used in different programming environments. This document is available in PostScript (.PS), HTML (.HTML), and ASCII text (.TXT) formats from the following location:

```
SYS$SYSROOT:[SYSHLP.EXAMPLES.DECSET]DECSET_COOKBOOK.*"
```

1.3 Installation Requirements

For a new installation of LSE the following is required:

OpenVMS Alpha Systems

Global Pages	10000
Global Sections	15
Disk Blocks	45000

OpenVMS VAX Systems

Global Pages	6000
Global Sections	15
Disk Blocks	40000

1.4 Compatibility Issues

In versions of LSE prior to V4.3 the UNDO command is implemented to reverse the effect of the last ENTER COMMENT command. This effect can still be achieved if the user specifies the following format:

```
UNDO ENTER [COMMENT]
```

2
New Features

The following enhancements were added to LSE Version 4.7.

2.1 Box Commands

Commands that operate on a selected box have been added.

The commands have OpenVMS and portable command language forms and they are also available from a new top level menu named Box.

BOX commands operate on a selected range. The standard methods of selecting a range are used. The top left corner of the box is given by the first character in the selection and the bottom right corner is given by the last character in the selection.

If, for the example shown below, a selection that includes all the uppercase letters is made, then for a BOX command the selected box is the area containing the letter B.

```
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeBBBBBBBBBBBBBBBBBIIIIIIIIII
IIIIIIIIBBBBBBBBBBBBBBBBBIIIIIIIIII
IIIIIIIIBBBBBBBBBBBBBBBBBIIIIIIIIII
IIIIIIIIBBBBBBBBBBBBBBBBBeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
```

If the BOX DRAW command is issued for the selection described above the result is:

```
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeee+----------------+IIIIIIIIII
IIIIIIII|BBBBBBBBBBBBBBBB|IIIIIIIIII
IIIIIIII|BBBBBBBBBBBBBBBB|IIIIIIIIII
IIIIIIII+----------------+eeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
```

Box operations are performed by the following commands:

OpenVMS Command	Portable Command
BOX COPY	BOX COPY
BOX CUT	BOX CUT
BOX CUT/PAD	BOX CUT PAD
BOX PASTE	BOX PASTE
BOX DRAW	BOX DRAW
BOX PASTE/OVERSTRIKE	BOX PASTE OVERSTRIKE
BOX LOWERCASE	BOX LOWERCASE
BOX UPPERCASE	BOX UPPERCASE

The BOX COPY command copies the currently selected box to the default location (the DECwindows clipboard or the paste buffer).

The BOX CUT command moves the currently selected box to the default location.

The BOX CUT/PAD command moves the currently selected box to the default location and the box is replaced by spaces.

The BOX PASTE command copies the contents of the default location to a box with its top left hand corner at the current position.

The BOX PASTE/OVERSTRIKE command copies the contents of the default location to a box with its top left hand corner at the current position. The contents are copied in overstrike mode.

The BOX DRAW command draws a box in overstrike mode. The box is drawn using the plus sign (+) for the corners, the vertical bar (|) for the sides and a hyphen (-) for the top and bottom.

The BOX LOWERCASE command changes the case of the text in the selected box to lowercase.

The BOX UPPERCASE command changes the case of the text in the selected box to uppercase.

The following restrictions apply:

- A box operation is not allowed if there is overview information in the records that make up the box.

- If a box operation is performed for which the total size of the records containing the box before or after the operation is greater than 65535 characters then undo information is reset.
- Auto erase is switched off during a box paste operation.
- Tabs in the records containing the box are converted to spaces and the records may be extended with spaces.

2.2 Command to Delete Overview Information

This command gets rid of all the overview lines in the current buffer and makes all the real lines in the buffer visible.

OpenVMS	Equivalent Portable
DELETE OVERVIEW	DELETE BUFFER OVERVIEW

The portable command clashes with the portable command to delete a buffer with name OVERVIEW. The problem can be avoided by enclosing OVERVIEW in quotes when it is the name of a buffer.

2.3 COBOL Language Template

A new version of the COBOL Language Template has been added to LSE.

2.4 Example Language Extensions

Language template files for Java (JAVA.LSE) and Perl (PERL.LSE) have been added to the examples directory (LSE$EXAMPLE).

2.5 TPU Pattern Style

A new pattern style TPU has been added to the existing pattern styles, VMS and ULTRIX.

For full details of TPU patterns see the *DEC Text Processing Utility Reference Manual*.

Pattern styles are used in:

- SEARCH/PATTERN command (VMS command language)
- SUBSTITUTE/PATTERN command (VMS command language)
- PATTERN SEARCH command (Portable command language)
- PATTERN SUBSTITUTE command (Portable command language)

- PATTERN EXACT SUBSTITUTE command (Portable command language)
- Search dialog box (Search / Search ...)
- Substitute dialog box (Search / Substitute ...)

The new TPU pattern style is used in a similar way to the existing styles. The main advantages of the new style are:

- Direct access to the powerful TPU pattern facility (the existing pattern styles use the facility indirectly).
- The ability to make a substitution that is a function of the pattern found.
- The ability to develop *Compaq Digital Test Manager for OpenVMS (DTM)* user defined filters interactively.

Most of the examples in the following sections are given using the VMS command language and assume that the search options are set to the TPU pattern style and case insensitive searching. The search direction is assumed to be forward and the cursor is assumed to be positioned before the example text. The search and replace parameters in the VMS command language examples can be used unchanged in the equivalent Portable command language commands.

2.5.1 User Interface

The TPU pattern style is selected in one of three ways:

```
SET SEARCH PATTERN=TPU (vms command language     )

SET SEARCH PATTERN TPU (portable command language)

Select TPU pattern on the Search Attributes window
(Options/Search Attributes...)
```

The only other user interface change is the form of the search and replace string parameters when the TPU pattern style is selected for a pattern search or substitution.

The search string parameter is a TPU expression that must evaluate to a TPU pattern and the replace string parameter is a TPU expression that must evaluate to a TPU string.

The following examples are given in both command languages. The first two examples search for abc or def and the last two examples substitute all occurrences of abc or def by ghi:

```
SEARCH/PATTERN  "'abc' | 'def'"
PATTERN SEARCH  "'abc' | 'def'"

SUBSTITUTE/PATTERN/ALL   "'abc' | 'def'"  "'ghi'"
PATTERN SUBSTITUTE  "'abc' | 'def'"  "'ghi'" ALL
```

In the above examples 'abc', 'def' and 'ghi' are TPU strings and | is the TPU pattern alternation operator.

The outermost quotes in the above example must be omitted if the parameters are prompted for or if a dialog box is used.

2.5.2 Partial Pattern Assignment Variables

Partial pattern assignment variables allow a substitution to be a function of the found pattern.

For example, the following command replaces a date of the form yyyy/mm/dd with one of the form dd/mm/yyyy:

```
SUBSTITUTE/PATTERN -
   "(_year@_v1)+'/'+(_month@_v2)+'/'+(_day@_v3)" -
   "str(_v3)+'/'+str(_v2)+'/'+ str(_v1)"

when applied to:  1998/04/21  generates:  21/04/1998
```

In the above example _year, _month and _day are TPU variables holding patterns that match the year, month and day parts of a date, for details of how to set up these variables, see Section 2.5.6.

In addition @ is the TPU partial pattern assignment operator, and _v1, _v2 and _v3 are partial pattern assignment variables that are set to the found year, date and day.

A partial pattern assignment variable holds a TPU range and when used in the replacement string must be converted to a string using the TPU procedure STR.

For example, the following command will prefix any lines that start with any three characters from ABCDEFGHI with XYZ_ :

```
SUBSTITUTE/PATTERN/ALL -
   "LINE_BEGIN + (ANY('ABCDEFGHI',3)@_v1)" -
   "'XYZ_'+ str(_v1)"

when applied to:  abc        generates:  XYZ_abc
                  012                     012
                  defghi                  XYZ_defghi
```

In the above example LINE_BEGIN is a TPU keyword that matches the beginning of a line and ANY is a TPU pattern procedure that matches a specified number of characters from a specified set of characters.

2.5.3 New Line

A new line will be generated for each line feed character in the replacement string, a line feed character can be introduced by means of the TPU procedure ASCII with the value 10 as a parameter.

For example, to replace any numbers at the end of lines with the string 'xxx' (a line feed is necessary because the search pattern includes the end of the line):

```
SUBSTITUTE/PATTERN/ALL -
   "_n + LINE_END" -
   "'xxx' + ASCII(10)"

when applied to:  123 456    generates:  123 xxx
                  789                    xxx
```

In the above example LINE_END is a TPU keyword that matches the end of a line and _n is TPU variable holding a pattern that matches a number.

When a partial pattern assignment variable is converted to a string by the TPU procedure STR an optional second parameter can be set to ASCII(10) to cause any end of lines in the range described by the variable to be converted to line feed characters (without the parameter they are represented by the null string). For example:

```
SUBSTITUTE/PATTERN/ALL -
   "(LINE_BEGIN + _n + LINE_END + _n + LINE_END)@_v1" -
   "STR(_v1, ASCII(10)) + STR(_v1, ASCII(10))"

when applied to:  123        generates:  123
                  456                    456
                                         123
                                         456
```

Carriage return characters adjacent to line feed characters in the replacement string are ignored.

2.5.4 Errors

The search and replace strings are TPU expressions which have to be evaluated and may generate various TPU compilation / evaluation error messages.

Two new error messages have been added for invalid search or replace parameters:

```
Error in search pattern
Error in replacement string
```

These messages will normally be preceded by various TPU error messages. For example, the search string "'aaa' + bbb" would result in the following error messages:

```
Undefined procedure call BBB
Operand combination STRING + INTEGER unsupported
Error in search pattern
```

2.5.5 Global Variables

Partial pattern assignment variables and pattern variables (such as _year in an earlier example) need to be global and must not clash with any TPU global variables used by LSE. This can be achieved by starting any such variable names with an underscore character.

2.5.6 Pattern Variables

Any complicated search or substitution is likely to need various pattern variables to have already been set up. This can be achieved in various ways.

The definitions can be setup by issuing DO/TPU commands, for example:

```
DO/TPU "_digits:='0123456789'"
DO/TPU "_digit:=any(_digits)"
DO/TPU "_year:=any(_digits,4)"

DO/TPU "_month:=any('01',1)+_digit"
DO/TPU "_day:=any('0123',1)+_digit"
DO/TPU "_n:=span(_digits)"
```

The file LSE$PATTERNS.TPU in the LSE$EXAMPLE directory contains some examples of patterns which can be added to LSE by means of the following commands:

```
GOTO FILE LSE$EXAMPLE:LSE$PATTERNS.TPU
EXTEND *
DO/TPU "LSE$PATTERNS_MODULE_INIT"
```

2.5.7 Use for Developing DTM User Filters

The user defined filters global replace feature introduced in DTM Version 4.0 can be simulated using the SUBSTITUTE/PATTERN/ALL command. This allows DTM user defined filters to be developed interactively using LSE.

For example, to replace any numbers at the end of lines with the string 'xxx':

```
global_replace(
   _n + LINE_END,
   'xxx' + ASCII(10),
   NO_EXACT,
   OFF,
   ON);
```

The LSE equivalent (assuming that the current search attributes are equivalent to NO_EXACT) is:

```
SUBSTITUTE/PATTERN/ALL -
   "_n + LINE_END" -
   "'xxx' + ASCII(10)"
```

The LSE equivalent of the pattern to replace parameter (first parameter of the global_replace routine) is the same except that the parameter has to be in quotes.

The LSE equivalent of the replacement string parameter (second parameter) is the same if the evaluate replacement parameter (fourth parameter) is set to ON and is the same except that the parameter has to be in quotes if the evaluate replacement parameter is set to OFF.

The LSE equivalent of the search mode parameter (third parameter) is the setting of the search options (set by the SET SEARCH command).

LSE does not have equivalents of the evaluate replacement parameter (fourth parameter) or the convert linefeeds parameter (fifth parameter). It always evaluates the replacement string parameter and it always converts linefeed characters (and ignores adjacent carriage return characters).

3
Problems Fixed

The following problems were fixed and restrictions removed in LSE Version 4.7:

- ID:2149 - Using the QUOTE command to enter the control sequence for the DO key, in DECwindows mode, no longer results in an error.
- ID:2150 - Tabs following continuation characters when using the Fortran language templates are now correctly dealt with.
- ID:2151 - The RECOVER BUFFER/ALL command now correctly recovers buffers.
- ID:2161 - CTRL/C when reading in a file now correctly cancels the operation.
- ID:2166 - The cursor is now returned to the correct window after a compilation.
- ID:2168 - The WRITE command no longer has a channel leak when writing files across the network.
- ID:2173 - The 'Directory not found' message is no longer followed by a 'Message number 00000000' message.
- ID:2181 - The default window size can now be changed by adjusting the values of Tpu.Tpu$MainWindow.Columns and Tpu.Tpu$MainWindow.Rows in LSE$DEFAULTS.DAT.
- ID:2193 - Files selected from *Compaq Code Management System for OpenVMS (CMS)* running in DECwindows mode are not placed into a LSE (running in DECwindows mode) buffer.

  When the DECwindows topShadowColor color is the same as the DECwindows background color or the monitor being used is monochrome, files reserved or fetched using the CMS DECwindows interface are not placed into the LSE DECwindows buffer. The file is, however, correctly fetched or reserved from the CMS library.

  This is a CMS problem and is fixed in CMS Version 4.1.

- ID:2199 - Errors when re-sizing LSE window in portable interface.

  In the DECwindows interface, with the Portable command language, after expanding a command token on the LSE command line, re-sizing the LSE window no longer produces an "unexpected internal error" and repeatedly resizing the window no longer results in an access violation.

  The re-sizing results in the expand operation being aborted.

- ID:2212 - LSE now starts correctly if the initial file is in a CMS library. An error message is no longer given and LSE starts with the correct buffer.
- ID:2222 - An UNDO following the expansion of a menu placeholder will now produce the correct result.
- ID:2223 - An UNDO following the replacement of a placeholder by undeleting a line will now produce the correct result.
- ID:2227 - The SHOW TOKEN/PLACEHOLDER <wildcarded name> will now show all relevant items.
- ID:2230 - LSE no longer fails with an access violation if a DO/LSE or DO/TPU command is executed when there are no buffers set up in the editor.
- ID:2236 - The LSE$CURRENT_FILE logical name is now updated after exiting from LSE even if there is an error encountered during compile/review.
- ID:2237 - An UNDO of a paste command now produces the correct result.
- ID:2246 - The SHOW KEY key-specifier command will no longer display an incorrect value when a key defined with the qualifier STATE=GOLD is redefined.
- ID:2248 - In DECwindows mode, if the first action is a PF1 and KP7 key sequence this will now correctly result in command mode being entered.
- ID:2253 - The UNDO of a soft tab conversion to a hard tab no longer produces a space.
- ID:2258 - File check when positioning in a file for SCA.

  If a file with the same name and type is already in a buffer LSE now generates a popup message if the device, node, directory or version is different from that requested by SCA.

- ID:2259 - Failure to position in a file for SCA.

  Even if LSE is positioned on the command line, the prompt or the message buffer it will now correctly position in a file when requested to do so by SCA.

- ID:2260 - The restriction that undo/redo does not work in overstrike mode is now fully applied.

  The main problems were:

  - If overstrike was selected from the status line the undo information was not reset.
  - Some undo information was still set up in overstrike mode.

- ID:2261 - UNDO/REDO commands no longer produce incorrect results in a buffer which has overview information.

  The commands that produce overview information are COLLAPSE, FOCUS and VIEW SOURCE.

  This has been made a restriction. Undo information is now reset and disabled for an operation that results in overview information being generated. A new command DELETE OVERVIEW (DELETE BUFFER OVERVIEW in portable command syntax) has been added to delete any overview information thus allowing undo processing to be reenabled.

- ID:2262 - UNDO/REDO commands now work correctly for editing operations that include positioning to a different buffer using the mouse.

  An example of the problem is shown below:

```
     Buffer 1          Buffer 2

     a                 x
     b                 y
     c                 z

  Perform the following operations:

     Position using the mouse to 'a' and insert 'A'
     Position using the mouse to 'z'
     Position using the mouse to 'c'
     Undo (via edit menu)
     Undo (via edit menu)  After first Undo:

     Aa
     b
     c

  After second Undo:

     c
```

- ID:2263 - Undo now works correctly when switched off and then back on after some intervening editing operations.

An example of the problem is shown below:

```
    a
    b
    c

Perform the following operations:

    Position to 'a' and insert 'A'
    Switch undo off by setting max undo to 0
    Position to 'c' and insert 'C'
    Switch undo on by setting max undo to 10
    Undo (via edit menu)

After Undo:

    c
```

- ID:2264 - Missing undo messages.

  The following new undo mesages have been added:

```
Undo/Redo disabled for all buffers

Undo/Redo disabled for the current buffer
   (<buffer name>)

Undo/redo reset and disabled because buffer
   set to overstrike mode

Undo/redo reset and disabled because buffer
   has overview information

Undo/redo reenabled
```

- ID:2265 - A large delete involving large records no longer fails with an ACCVIO.

  The failure would only occur for a deletion greater than 65535 characters and records greater than 500 characters.

- ID:2267 - Changes to search attributes are now applied when a change is made without also changing the search string.

- ID:2268 - A reverse direction Search/Substitute ... when positioned near the end of a buffer no longer results in an unexpected internal error message.

- ID:2270 - When using eXcursion, an operation that minimizes a LSE window no longer results in a 'Terminal page length is too small for editor' message.

- ID:2271 - UNDO now works correctly after a Search/Substitute ... operation.

An example of an undo problem is shown below:

```
abc abc abc   Position to end of line
              Setup for a reverse substitute of abc
              by ABC and then use the combination of
              Find Next followed by Replace and
              Find Next 3 times.

ABC ABC ABC   Undo the substitutions.

abc abcabc    A space character is lost.
```

- ID:2272 - A Search/Substitute ... Apply or OK operation no longer moves the cursor when the search string cannot be found.
- ID:2274 - A redo of a case matched substitution will now work correctly.
- ID:2275 - A case matched pattern substitution across lines will now work correctly.

  An example of the problem is shown below:

```
Aaa           Use VMS pattern matching to substitute
BBB ccc       the pattern aaa\>bbb ccc by XXX YYY ZZZ

XXX yyy zzz   Incorrect result
```

- ID:2278 - A Search/Substitute ... Replace and Find Next operation now works correctly when:

  1. Positioned on an occurrence of the search string at the beginning of the buffer.
  2. Positioned on an occurrence of a pattern search string that spans more than one line.

- ID:2280 - A fault that caused mouse operations for the Alpha version of LSE to randomly fail with an ACCVIO has been corrected.

4
Known Problems

4.1 General

The following known problems exist in LSE Version 4.7:

- ID:2155 - Internal error compiling a *Compaq DECforms for OpenVMS* program after a *Compaq Ada* program.

  LSE will crash with an internal error when attempting to follow the compilation of an Ada program with the compilation of DECforms source. The source of the two programs appears to be unimportant.

  The workaround is to compile the DECforms source first.

- ID:2156 - SCA LOAD and ANALYZE command parameters do not default to current buffer.

  The file-spec[,..] parameter cannot be defaulted, the HELP information incorrectly says the default is the current buffer.

- ID:2158 - Cannot define an ALT key sequence as the Gold key.

  LSE cannot accept an ALT key sequence as the Gold key. When an attempt is made, LSE accepts the key definition without errors, but does not recognize the ALT key sequence as the Gold key (this definition does however disable PF1 as the Gold key), however if you show the key definitions the newly defined key has the correct definition.

  Different results are apparent depending on whether the DECwindows interface to LSE is used or if the character-cell mode interface to LSE is used. For example, in the DECwindows mode, ALT/G can be defined as the Gold key. LSE accepts the key definition, but ALT/G is not recognized as the Gold key. However, if SHIFT/ALT/G is pressed then LSE recognizes it as the Gold key. In the character-cell interface not even SHIFT/ALT/G is recognized. Again, LSE accepts the definition of ALT/G as the Gold key without errors.

  The workaround is to define the key via TPU:

  ```
  LSE> do/tpu "set(shift_key,key_name('g',alt_modified)
  ```

- ID:2164 - TPU$_WINDNOTMAPPED error reading TPU procedure into DECwindows LSE.

  DECwindows LSE may report the following error when trying to read in and compile a TPU procedure at startup:

  ```
  The window is not mapped to a buffer
 (TPU$_WINDNOTMAPPED).
  ```

  When this problem occurs, LSE will display an invalid window format (one that does not have a status line). The errors themselves will not be visible unless you display the contents of the message buffer. For example:

  ```
  DO
  LSE> GOTO BUFFER $MESSAGES
  ! a normal window is display for buffer $MAIN
  DO
  LSE> GOTO BUFFER $MESSAGES
  ```

  The workaround is to start LSE with the command LSE/COMMAND=DE_TAB.TPU and add the following line at the bottom of the DE_TAB.TPU procedure:

  ```
  LSE_NEW_KEY( "TAB", "TPU REM_TABS" );
  ```

  This will automatically read in the TPU procedure, compile it, and define an LSE key to run it.

- ID:2165 - DECwindows SET DEFAULT does not change default disk after OPEN FILE.

  In the LSE DECwindows interface, the SET DEFAULT command does not change the default device when issued after using the OPEN FILE command. It correctly changes the default directory but not the device. Prior to calling OPEN FILE, SET DEFAULT works as required.

  When you exit LSE your default device is changed.

  The workaround is to perform the SET DEFAULT command before opening any files in the DECwindows interface or specify the default directory as the filter specification in the open file window.

- ID:2174 - Cursor positioning across split screens.

  When splitting a file into two or more screens, the cursor correctly maintains its position when going back and forth between the windows. However, using the up or down arrows can cause the cursor to change column position to the same position in the other screen.

This problem occurs only when moving up or down or the last operation is an up/down motion. For example, moving up in the top screen in column 30, then switching to the bottom screen and trying to move up in column 10: the cursor jumps to column 30. If you first move the cursor left/right or perform any other action, the cursor stays in the column expected.

The problem only occurs when using the portable command language interface.

- ID:2198 - Cannot access help text after using ENABLE VMS INTEGRATION.

  After issuing the ENABLE VMS INTEGRATION command, a HELP FIND returns "No help available for the topic specified". HELP GOTO DECLARATION has the same problem.

  The ENABLE VMS INTEGRATION command enables two additional command prefixes VMSCMS_ and VMSSCA_.

  The error occurs because there is a FIND routine associated with the VMSSCA_ prefix for which there is no help. The FIND routine associated with the VMSSCA_ prefix hides the standard FIND routine (associated with the default prefix LSE_).

  The help for the standard FIND routine can still be displayed if part of it's second key is also given as this makes it unique and stops it being hidden (e.g. FIND OCCURRENCES).

  The real error is that there is no help for any of the VMSCMS_ or VMSSCA_ prefix commands.

- ID:2216 - SET DIRECTORY DEFAULT not propagated to current compile subprocess.

  LSE's portable command language does not propagate changes in the default directory to an already existing compile subprocess; causing the next COMPILE REVIEW command to fail with "Error opening <diagnostic-file>".

- ID:2220 - UNDO access violation changing lines using keys bound to TPU procedures.

  The fault can occur if user written TPU code is associated with a key. The fault can occur in the setting up of information required to support undo/redo processing for user written TPU code associated with a key. When the key is used the call of the associated code is not jacketed by any LSE code and can not be protected against undo/redo problems.

Four possible workarounds are:

1. Disabling undo/redo processing by means of the SET MODE UNDO (SET UNDO portable) command.
2. Setting the value of MAX_UNDO to zero for each buffer being edited; this may be done via a key definition:

   For example:

   ```
   LSE Command> DEFINE KEY "F17" "SET MAX_UNDO 0"
   ```

3. Defining all your keys using the LSE "DEFINE KEY" or "NEW KEY" commands.

   For example:

   ```
   LSE Command> DEFINE KEY
   _Key: "GOLD-Z"
   _String: "do/tpu ""<tpu-procedure-call>"""
   ```

   or:

   ```
   LSE> NEW KEY
   Key Name: GOLD-Z
   LSE Command: tpu "<tpu-procedure-call>"
   ```

4. Making the TPU code to be executed by the defined key a parameter of the routine lse_tpu. This routine protects against undo/redo problems by deleting the undo/redo information for all buffers and inhibiting the setting up of any undo/redo information during the execution of the TPU code associated with the key.

   For example:-

   ```
   define_key("lse_tpu('position(buffer_begin)')",
              key_name("x",shift_key))
   ```

- ID:2225 - A very long command line can produce inappropriate errors.

  A very long command can produce inappropriate errors such as 'Compilation aborted'.

  For example:

  ```
  LSE> NEW FILE AAAAAA:[BBBBBBBBBBBBBBBBBBBBBB....
  ```

```
Line truncated to 256 characters
;         0001      LSE_NEW_FILE("AAAAAA:[BBBB....
;                         ^
Missing quote
;         0002      <end-of-file>
One of the following symbols was expected:
;           , )
;         0002      <end-of-file>
;           ^
Premature end-of-file detected
Compilation aborted
Execution aborted
Execution aborted (TPU$_EXECUTEFAIL)
unexpected internal error in
lse$$command_undefined_key_proc at line 4922.
   Please submit an SPR.
```

- ID:2228 - LSE may fail when executing user's TPU code via a pull-down menu or a user command language.

  This is similar to ID:2220.

- ID:2229 - A spurious error message may be produced when the help command is issued for a long non-existent topic. For example:

```
LSE> HELP Nonexistent_topic

Error opening LSE$HELP as input

          No help available for the topic specified
```

- ID:2231 - If a buffer which has been reviewed is deleted using the DELETE BUFFER command the REVIEW information associated with the buffer is not cleaned up. For example, following the deletion, the END REVIEW command does not produce the error "NO REVIEW IN PROGRESS".

- ID:2243 - CTRL/C may not be correctly handled if issued while a DO command is processing a buffer.

- ID:2245 - The VMS and portable syntax commands are not consistent in their generation of brief or full information. The /FULL qualifier does not always result in a full display and the /BRIEF qualifier does not always produce a brief display.

- ID:2247 - Command mode prompt problem.

  The 'LSE>' prompt is at times improperly displayed when the 'LSE Command>' prompt for single command mode is expected.

- ID:2254 - Unavailable SCA commands.

  The SCA commands IMPORT, SAVE QUERY and @file-spec are not available from LSE.

- ID:2273 - Undo LF and CR character problem

  Undo may not work correctly when applied to the deletion of a line feed, for example:

```
a<LF>a    Using pattern substitution replace a*a
          by bbb

bbb       Undo

a         The line feed character is lost and the
a         two a's are in different lines. Any carriage
          return characters adjacent to the line feed
          character would also be lost.
```

4.2 Documentation

The following documenation problems exist for LSE Version 4.7:

- ID:2266 - Incorrect Language Definition Example

  The example in Section 4.1.1 of the *Guide to DIGITAL Language-Sensitive Editor for OpenVMS Systems* is incorrect, it contains:

```
PLSE SET LANGUAGE FILE TYPES ".MEMO .FOO" ADD memo
```

  This should be:

```
PLSE SET LANGUAGE FILE TYPES ".MEMO" ADD memo
PLSE SET LANGUAGE FILE TYPES ".FOO" ADD memo
```

  This problem replaces ID:2196 which was raised against an incorrect use of the SET LANGUAGE FILE TYPES command.

- ID:2276 - Missing documentation of LSE$$SOURCE

  Internally LSE uses the logical LSE$$SOURCE for source directories. LSE$$SOURCE is initialised from LSE$SOURCE. LSE$SOURCE is not updated when LSE$$SOURCE changes.

5
Restrictions

The following restrictions apply to LSE Version 4.7:

- LSE does not support VT500 series terminals. A VT500 series terminal should be set to emulate a supported terminal such as a VT300 series terminal.
- The UNDO operation has no effect for the LSE FOCUS command. When the FOCUS command is executed, the UNDO buffers will be reset. No previous operations can be undone.
- No UNDO support for TPU commands. For example :

  ```
  TPU "erase (current_buffer)" (Portable)
  DO/TPU "erase (current_buffer)" (OpenVMS)
  ```

  When a TPU command is executed, the undo buffers will be reset. No previous operations can be undone.
- UNDO and REDO commands are disabled in OVERSTRIKE mode.

  If the buffer is in OVERSTRIKE mode (not insert), initiating the UNDO or REDO command will have no effect. When the mode is changed to INSERT, undoing and redoing is possible again.
- UNDO and REDO commands are disabled following a command that generates overview information. They can be reenabled by deleting overview information, see Section 2.2.
- *Compaq Fortran for OpenVMS Systems* - Fortran 90 Language Support.

  Automatic generation of continuation lines is not supported.
- DECnet/OSI Phase V support.

  When using portable syntax commands, file names that start with a quote must be enclosed in quotes (single or double) with inner quotes of the same type as the enclosing quotes doubled up.

For example:

```
LSE> NEW FILE '"A&=BCD"::B.C'
LSE> NEW FILE """AAAAAAAAAAAAAAA""::B.C"
```

6
Reporting Problems and Suggestions

Depending on the nature of the request or problem, customers can call Compaq provided their software contract and warranty entitles them to telephone support.

Review the SPD and Warranty Addendum for an explanation of warranty.

If a problems is encountered during the warranty period, customers can report the problem using DSNLink. In the United States, customers can call the Compaq Customer Services Center at 1-800-354-9000.

In addition, Compaq now provides access to Web Information and Support (WIS), an Internet-based electronic services application that allows customers to submit electronic service requests, access databases, download patch kits, transfer files from Compaq, and connect to global notes conferences via the World Wide Web.

Dial-In Access (DIA) users can access WIS by:

1. Opening http://www1.support.compaq.com/wis.
2. Choosing a country code.
3. Entering DIA Access ID and password.

WIS and DIA use the same underlying account and directory on the Compaq Customer Support Center host system.

Customers can register for DIA, and receive a WIS account by contacting their Customer Support Center or local Compaq representative.

This method can also be used to make requests for additional features or modifications to existing features.

When submitting a report, please include the following information:

1. LSE version and type of OpenVMS operating system.
2. Problem summary.
3. Stack dumps or error logs (if applicable).

4. Files in machine-readable media: magnetic tape (preferred), floppy diskettes, or tape cassettes.