


```

LL          SSSSSSSS UU      UU  BBBB88886
LL          SSSSSSSS UU      UU  888888888
LL         SS          UU      UU  88          88
LL         SS          UU      UU  88          88
LL         SS          UU      UU  88          88
LL         SS          UU      UU  88          88
LL          SSSSSS    UU      UU  888888888
LL          SSSSSS    UU      UU  888888888
LL                   SS     UU      UU  88          88
LL                   SS     UU      UU  88          88
LL                   SS     UU      UU  88          88
LL                   SS     UU      UU  88          88
LLLLLLLLLLL SSSSSSSS  UUUUUUUUU  888888888
LLLLLLLLLLL SSSSSSSS  UUUUUUUUU  888888888

```

```

....
....
....
....

```

```

LL          IIIIII    SSSSSSSS
LL          IIIIII    SSSSSSSS
LL           II       SS
LL           II       SS
LL           II       SS
LL           II       SS
LL           II       SSSSSS
LL           II       SSSSSS
LL           II       SS
LL           II       SS
LL           II       SS
LL           II       SS
LLLLLLLLLLL IIIIII    SSSSSSSS
LLLLLLLLLLL IIIIII    SSSSSSSS

```



```

1 0001 0 %TITLE 'EDT$LSUB - SUBSTITUTE in one line of text'
2 0002 0 MODULE EDT$LSUB ( ! SUBSTITUTE in one line of text
3 0003 0 IDENT = 'V04-000' ! File: LSUB.BLI Edit: JBS1008
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
11 0011 1 * ALL RIGHTS RESERVED. *
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
18 0018 1 * TRANSFERRED. *
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
22 0022 1 * CORPORATION. *
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 **
32 0032 1 FACILITY: EDT -- The DEC Standard Editor
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 This module scans a line of text for a match with the
37 0037 1 search string. If found, it performs the substitution.
38 0038 1 This is used by the line mode commands SUBSTITUTE
39 0039 1 and SUBSTITUTE NEXT.
40 0040 1
41 0041 1 ENVIRONMENT: Runs at any access mode - AST reentrant
42 0042 1
43 0043 1 AUTHOR: Bob Kushlis, CREATION DATE: February 3, 1978
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1 1-001 - Original. DJS 30-JAN-1981. This module was created by
48 0048 1 extracting the routine SUBSTITUTE from the module EXEC.BLI.
49 0049 1 1-002 - DJS 16-MAR-1981. Before the SUBSTITUTE search string is
50 0050 1 compared to part of a text line, an additional test was added
51 0051 1 to ensure that there is enough text left on the line to make
52 0052 1 a match possible. Previously, it was possible for data beyond
53 0053 1 the end-of-line to be included in the string match.
54 0054 1 1-003 - Regularize headers. JBS 20-Mar-1981
55 0055 1 1-004 - Use the new message codes. JBS 06-Aug-1981
56 0056 1 1-005 - Force cursor to remain positioned less than 256 characters
57 0057 1 from the beginning of the line. Previously,, string substitution

```

EDT\$LSUB
V04-000

EDT\$LSUB - SUBSTITUTE in one line of text

E 13
16-Sep-1984 01:00:48
14-Sep-1984 12:23:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]LSUB.BLI;1

Page 2
(1)

```
.. 58      0058 1 |      could leave the cursor after the end of the line.  
.. 59      0059 1 |      STS 17-Sep-1981  
.. 60      0060 1 |      1-006 - Remove EDT$SA_STR_CMP. JBS 16-Jul-1982  
.. 61      0061 1 |      1-007 - Mark the line changed in the screen data base. JBS 01-Dec-1982  
.. 62      0062 1 |      1-008 - Change the call to EDT$MRK_LNCHG. JBS 27-Dec-1982  
.. 63      0063 1 |      --  
.. 64      0064 1 |
```

ED
VO

```
.. 66 0065 1 %SBTTL 'Declarations'  
.. 67 0066 1 :  
.. 68 0067 1 : TABLE OF CONTENTS:  
.. 69 0068 1 :  
.. 70 0069 1 :  
.. 71 0070 1 REQUIRE 'EDT$SRC:TRAROUNAM';  
.. 72 0509 1 :  
.. 73 0510 1 FORWARD ROUTINE  
.. 74 0511 1 EDT$$$SUB_NXT : NOVALUE;  
.. 75 0512 1 :  
.. 76 0513 1 :  
.. 77 0514 1 : INCLUDE FILES:  
.. 78 0515 1 :  
.. 79 0516 1 :  
.. 80 0517 1 REQUIRE 'EDT$SRC:EDTREQ';  
.. 81 0652 1 :  
.. 82 0653 1 :  
.. 83 0654 1 : MACROS:  
.. 84 0655 1 :  
.. 85 0656 1 : NONE  
.. 86 0657 1 :  
.. 87 0658 1 : EQUATED SYMBOLS:  
.. 88 0659 1 :  
.. 89 0660 1 : NONE  
.. 90 0661 1 :  
.. 91 0662 1 : OWN STORAGE:  
.. 92 0663 1 :  
.. 93 0664 1 : NONE  
.. 94 0665 1 :  
.. 95 0666 1 : EXTERNAL REFERENCES:  
.. 96 0667 1 :  
.. 97 0668 1 : In the routine
```

! Do a substitution within one line of text

```

99 0669 1 %SBTTL 'EDT$$$SUB_NXT - substitute in one line of text'
100 0670 1
101 0671 1 GLOBAL ROUTINE EDT$$$SUB_NXT (           ! substitute in one line of text
102 0672 1     SINGLE,                               ! 1 = just do one substitution
103 0673 1     START_POS                             ! start searching here
104 0674 1     ) : NOVALUE =
105 0675 1
106 0676 1 !++
107 0677 1 !FUNCTIONAL DESCRIPTION:
108 0678 1
109 0679 1     Do substitutions in one line of text. This routine scans the current
110 0680 1     line of text, starting at the character position START_POS, looking
111 0681 1     for a string that matches the search string. If it is found, that string
112 0682 1     is replaced by the substitute string. If the SINGLE flag is on,
113 0683 1     it returns after one substitution, otherwise all matches in the line are substituted.
114 0684 1
115 0685 1 !FORMAL PARAMETERS:
116 0686 1
117 0687 1     SINGLE                               1 = just do one substitution
118 0688 1
119 0689 1     START_POS                             where to start the search
120 0690 1
121 0691 1 !IMPLICIT INPUTS:
122 0692 1
123 0693 1     EDT$$A_CUR_BUF
124 0694 1     EDT$$T_LN_BUF
125 0695 1     EDT$$G_SEA_STRLEN
126 0696 1     EDT$$T_SEA_STR
127 0697 1     EDT$$T_SUB_STR
128 0698 1     EDT$$G_SUB_STRING
129 0699 1     EDT$$A_WK_CN
130 0700 1     EDT$$G_EXE_QRYQUIT
131 0701 1     EDT$$G_EXE_SUBCNT
132 0702 1     EDT$$G_EXE_SBITS
133 0703 1     EDT$$G_EXCT_MATCH
134 0704 1
135 0705 1 !IMPLICIT OUTPUTS:
136 0706 1
137 0707 1     EDT$$G_LN_LEN
138 0708 1     EDT$$G_EXE_SUBCNT
139 0709 1
140 0710 1 !ROUTINE VALUE:
141 0711 1
142 0712 1     NONE
143 0713 1
144 0714 1 !SIDE EFFECTS:
145 0715 1
146 0716 1     NONE
147 0717 1
148 0718 1 !--
149 0719 1
150 0720 2     BEGIN
151 0721 2
152 0722 2     EXTERNAL ROUTINE
153 0723 2     EDT$$FMT_MSG,
154 0724 2     EDT$$PUT_CH : NOVALUE,
155 0725 2     EDT$$PROC_QRYQAL,

```

```

156 0726 2      EDT$SRPL_LN,
157 0727 2      EDT$STY_CURLN,
158 0728 2      EDT$$$STR_CMP,
159 0729 2      EDT$$$MRK_LNCHG : NOVALUE;
160 0730 2
161 0731 2      EXTERNAL
162 0732 2      EDT$$A_CUR_BUF : REF TBCB BLOCK,
163 0733 2      EDT$$T_LN_BUF : VECTOR [255, BYTE],
164 0734 2      EDT$$$G_LN_LEN,
165 0735 2      EDT$$$G_SEA_STRLEN,
166 0736 2      EDT$$T_SEA_STR,
167 0737 2      EDT$$T_SUB_STR,
168 0738 2      EDT$$$G_SUB_STRLEN,
169 0739 2      EDT$$A_WK_LN : REF LIN_BLOCK,
170 0740 2      EDT$$$G_EXE_ORYQUIT,
171 0741 2      EDT$$$G_EXE_SUBCNT,
172 0742 2      EDT$$$G_EXE_SBITS,
173 0743 2      EDT$$$G_EXCT_MATCH;
174 0744 2
175 0745 2      MESSAGES ((LINEXC255));
176 0746 2
177 0747 2      LOCAL
178 0748 2      OLD : REF VECTOR [255, BYTE],
179 0749 2      END_LINE,
180 0750 2      OLD_LEN,
181 0751 2      REPPPOINT,
182 0752 2      NEW_POS,
183 0753 2      CHANGED,
184 0754 2      THIS_CHAR_CHANGED;
185 0755 2
186 0756 2      :+ Get a local pointer to the line and its length;
187 0757 2      :-
188 0758 2      OLD = EDT$$A_WK_LN [LIN_TEXT];
189 0759 2      OLD_LEN = .EDT$$A_WK_LN [LIN_LENGTH];
190 0760 2
191 0761 2      :+ Initialize the length of the new line, changed flag and position of
192 0762 2      :- last SUBSTITUTE.
193 0763 2      EDT$$$G_LN_LEN = 0;
194 0764 2      CHANGED = 0;
195 0765 2      NEW_POS = 0;
196 0766 2
197 0767 2      :+ Get a pointer to the end of the old line.
198 0768 2      :-
199 0769 2      END_LINE = CH$PTR (.OLD, .EDT$$A_WK_LN [LIN_LENGTH]);
200 0770 2
201 0771 2      :+ Loop until either we finish the line or the new line exceed 255 chars.
202 0772 2      :-
203 0773 2      WHILE ((.OLD LSSA .END_LINE) AND (.EDT$$$G_LN_LEN LSS 256)) DO
204 0774 2      BEGIN
205 0775 2      THIS_CHAR_CHANGED = 0;
206 0776 2
207 0777 2      :+ Look for a match if the position is greater or equal to
208 0778 2      :- start position.
209 0779 2
210 0780 2
211 0781 2
212 0782 2

```

```

! Compare strings of equal length
! Mark the line changed in the screen data base

```

```

! No. of substitutions done.
! The options switches.
! Type of string matching

```

```

! Pointer to the original line
! Pointer to end of the line
! Length of original line
! Pointer to the replacement string
! Character position after last subs.
! Flag indicating this line was changed.
! Flag indicating this character was changed

```

```

213 0783 3
214 0784 4 IF ((.EDT$$G_LN_LEN GEQ .START_POS) AND
215 0785 4 ((.OLD + .EDT$$G_SEA_STRLEN) LEQA .END_LINE) AND
216 0786 4 (NOT .EDT$$G_EXE_QRYQUIT) AND
217 0787 4 (NOT (.SINGLE AND .CHANGED)))
218 0788 3 THEN
219 0789 3
220 0790 3 IF EDT$$STR_CMP (.OLD, EDT$$T_SEA_STR, .EDT$$G_SEA_STRLEN, .EDT$$G_EXCT_MATCH)
221 0791 3 THEN
222 0792 3
223 0793 3 IF EDT$$PROC_QRYQAL (.OLD, .END_LINE)
224 0794 3 THEN
225 0795 4 BEGIN
226 0796 4
227 0797 4 + A match was found. Set flag indicating that a change has been
228 0798 4 - made and increment the count.
229 0799 4
230 0800 4 THIS_CHAR_CHANGED = 1;
231 0801 4 CHANGED = 1;
232 0802 4 EDT$$G_EXE_SUBCNT = .EDT$$G_EXE_SUBCNT + 1;
233 0803 4
234 0804 4 + Mark the line as changed in the screen data base.
235 0805 4 -
236 0806 4 EDT$$MRK_LNCHG (SCR_EDIT_MODIFY, .EDT$$G_LN_LEN);
237 0807 4
238 0808 4 + Get a pointer to the replacement string.
239 0809 4 -
240 0810 4 REPPPOINT = EDT$$T_SUB_STR;
241 0811 4
242 0812 4 + Move the replacement string into the new line buffer.
243 0813 4 -
244 0814 4
245 0815 4 INCR J FROM 1 TO .EDT$$G_SUB_STRLEN DO
246 0816 4 EDT$$PUT_CH (CH$RCHAR_A (REPPPOINT));
247 0817 4
248 0818 4 NEW_POS = .EDT$$G_LN_LEN;
249 0819 4
250 0820 4 + Bump the old line pointer by the length of the search string.
251 0821 4 -
252 0822 4 OLD = CH$PLUS (.OLD, .EDT$$G_SEA_STRLEN);
253 0823 3 END;
254 0824 3
255 0825 4 IF (NOT .THIS_CHAR_CHANGED)
256 0826 3 THEN
257 0827 4 BEGIN
258 0828 4
259 0829 4 + No match, just copy the character and continue.
260 0830 4 -
261 0831 4 EDT$$PUT_CH (CH$RCHAR_A (OLD));
262 0832 3 END;
263 0833 3
264 0834 2 END;
265 0835 2
266 0836 2
267 0837 2 + Check to see if any substitutions have been done.
268 0838 2 -
269 0839 2

```



```

: 270 0840 2 IF .CHANGED
: 271 0841 2 THEN
: 272 0842 2 BEGIN
: 273 0843 2
: 274 0844 2 | Check for overflow of the newly constructed line.
: 275 0845 2 |
: 276 0846 2 |
: 277 0847 2 | IF (.EDT$$G_LN_LEN GTR 255)
: 278 0848 2 | THEN
: 279 0849 2 | BEGIN
: 280 0850 2 | EDT$$FMT MSG (EDT$ LINEXC255);
: 281 0851 2 | EDT$$G_LN_LEN = 255;
: 282 0852 2 | END;
: 283 0853 2 |
: 284 0854 2 |
: 285 0855 2 | Also need to check if new cursor position would be after 255 characters.
: 286 0856 2 |
: 287 0857 2 |
: 288 0858 2 | IF (.NEW_POS GTR 255) THEN NEW_POS = 255;
: 289 0859 2 |
: 290 0860 2 |
: 291 0861 2 | Replace the line in the buffer, and update the position to
: 292 0862 2 | the position after the last substitution.
: 293 0863 2 |
: 294 0864 2 | EDT$$RPL LN (EDT$$T LN_BUF, .EDT$$G LN_LEN);
: 295 0865 2 | EDT$$A_COR_BUF [TBCB_CHAR_POS] = .NEW_POS;
: 296 0866 2 |
: 297 0867 2 | Unless /NOTYPE was specified, display the updated line.
: 298 0868 2 |
: 299 0869 2 |
: 300 0870 2 | IF ( NOT .EDT$$G_EXE_SBITS<OPB_NOTYP>) THEN EDT$$TY_CURLN ( )
: 301 0871 2 |
: 302 0872 2 | END;
: 303 0873 2 |
: 304 0874 1 END;

```

! of routine EDT\$\$\$SUB_NXT

```

.TITLE EDT$LSUB EDT$LSUB - SUBSTITUTE in one line of t
      ext
.IDENT \V04-000\

.EXTRN EDT$$FMT MSG, EDT$$PUT_CH
.EXTRN EDT$$PROC QRYQAL
.EXTRN EDT$$RPL_CN, EDT$$TY_CURLN
.EXTRN EDT$$STR_CMP, EDT$$MRK_LNCHG
.EXTRN EDT$$A_COR_BUF, EDT$$T_LN_BUF
.EXTRN EDT$$G_LN_LEN, EDT$$G_SEA_STRLEN
.EXTRN EDT$$T_SEA_STR, EDT$$T_SUB_STR
.EXTRN EDT$$G_SUB_STRLEN
.EXTRN EDT$$A_WK_CN, EDT$$G_EXE_QRYQUIT
.EXTRN EDT$$G_EXE_SUBCNT
.EXTRN EDT$$G_EXE_SBITS
.EXTRN EDT$$G_EXCT_MATCH
.EXTRN EDT$ LINEXC255

.PSECT _EDT$CODE, NOWRT, SHR, PIC, 2

```

	OFFC	00000	.ENTRY	EDT\$\$\$SUB_NXT, Save R2,R3,R4,R5,R6,R7,R8,R9,-;	0671
			MOVAB	R10, R11	
5B	00000000G	00 9E 00002	MOVAB	EDT\$\$\$SEA_STRLEN, R11	
5A	00000000G	00 9E 00009	MOVAB	EDT\$\$\$LN_LEN, R10	
50	00000000G	00 D0 00010	MOVL	EDT\$\$\$WK_LEN, R0	0759
52	07	A0 9E 00017	MOVAB	7(R0), OLD	
51		60 9A 0001B	MOVZBL	(R0), OLD_LEN	0760
		6A D4 0001E	CLRL	EDT\$\$\$LN_LEN	0765
		55 7C 00020	CLRQ	NEW_POS	0767
57		60 9A 00022	MOVZBL	(R0), END_LINE	0771
57		52 C) 00025	ADDL2	OLD, END_LINE	
57		52 D1 00028 1\$:	CMP	OLD, END_LINE	0776
		03 1F 0002B	BLSSU	3\$	
		009D 31 0002D 2\$:	BRW	9\$	
00000100	8F	6A D1 00030 3\$:	CMP	EDT\$\$\$LN_LEN, #256	
		F4 18 00037	BGEQ	2\$	
		58 D4 00039	CLRL	THIS_CHAR_CHANGED	0778
08	AC	6A D1 0003B	CMP	EDT\$\$\$LN_LEN, START_POS	0784
		7C 19 0003F	BLSS	7\$	
50	52	6B C1 00041	ADDL3	EDT\$\$\$SEA_STRLEN, OLD, R0	0785
	57	50 D1 00045	CMP	R0, END_LINE	
		73 1A 00048	BGTRU	7\$	
	6C 00000000G	00 E8 0004A	BLBS	EDT\$\$\$EXE_QRYQUIT, 7\$	0786
	03 04	AC E9 00051	BLBC	SINGLE, 4\$	0787
	65	56 E8 00055	BLBS	CHANGED, 7\$	
	00000000G	00 DD 00058 4\$:	PUSHL	EDT\$\$\$EXCT_MATCH	0790
		6B DD 0005E	PUSHL	EDT\$\$\$SEA_STRLEN	
	00000000G	00 9F 00060	PUSHAB	EDT\$\$\$SEA_STR	
		52 DD 00066	PUSHL	OLD	
00000000G	00	04 FB 00068	CALLS	#4, EDT\$\$\$STR_CMP	
	4B	50 E9 0006F	BLBC	R0, 7\$	
	0084	8F BB 00072	PUSHR	#^M<R2,R7>	0793
00000000G	00	02 FB 00076	CALLS	#2, EDT\$\$\$PROC_QRYQAL	
	3D	50 E9 0007D	BLBC	R0, 7\$	
	58	01 D0 00080	MOVL	#1, THIS_CHAR_CHANGED	0800
	56	01 D0 00083	MOVL	#1, CHANGED	0801
	00000000G	00 D6 00086	INCL	EDT\$\$\$EXE_SUBCNT	0802
		6A DD 0008C	PUSHL	EDT\$\$\$LN_LEN	0806
		01 DD 0008E	PUSHL	#1	
00000000G	00	02 FB 00090	CALLS	#2, EDT\$\$\$MRK_LNCHG	
	59 00000000G	00 9E 00097	MOVAB	EDT\$\$\$SUB_STR, REPOINT	0810
	54 00000000G	00 D0 0009E	MOVL	EDT\$\$\$SUB_STRLEN, R4	0815
		53 D4 000A5	CLRL	J	
		0A 11 000A7	BRB	6\$	
	7E	89 9A 000A9 5\$:	MOVZBL	(REPOINT)+, -(SP)	0816
00000000G	00	01 FB 000AC	CALLS	#1, EDT\$\$\$PUT_CH	
F2	53	54 F3 000B3 6\$:	AOBLEQ	R4, J, 5\$	
	55	6A D0 000B7	MOVL	EDT\$\$\$LN_LEN, NEW_POS	0818
	52	6B C0 000BA	ADDL2	EDT\$\$\$SEA_STRLEN, OLD	0822
	0A	58 E8 000BD 7\$:	BLBS	THIS_CHAR_CHANGED, 8\$	0825
	7E	82 9A 000C0	MOVZBL	(OLD)+, -(SP)	0831
00000000G	00	01 FB 000C3	CALLS	#1, EDT\$\$\$PUT_CH	
		FF5B 31 000CA 8\$:	BRW	1\$	0776
	50	56 E9 000CD 9\$:	BLBC	CHANGED, 12\$	0840
000000FF	8F	6A D1 000D0	CMP	EDT\$\$\$LN_LEN, #255	0847
		11 15 000D7	BLEQ	10\$	
	00000000G	8F DD 000D9	PUSHL	#EDT\$_LINEXC255	0850

EDT\$LSUB
V04-000

EDT\$LSUB - SUBSTITUTE in one line of text
EDT\$\$\$SUB_NXT - substitute in one line of text

L 13
16-Sep-1984 01:00:48
14-Sep-1984 12:23:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]LSUB.BLI;1

Page 9
(3)

ED
VO

00000000G	00		01	FB	000DF	CALLS	#1, EDT\$\$\$FMT_MSG	:		
	6A	FF	8F	9A	000E6	MOVZBL	#255, EDT\$\$G_LN_LEN	:	0851	
000000FF	8F		55	D1	000EA	10\$:	CMPL	NEW_POS, #255	:	0858
			04	15	000F1		BLEQ	11\$:	
	55	FF	8F	9A	000F3		MOVZBL	#255, NEW_POS	:	
			6A	DD	000F7	11\$:	PUSHL	EDT\$\$G_LN_LEN	:	0864
00000000G	00	00000000G	00	9F	000F9		PUSHAB	EDT\$\$T_LN_BUF	:	
	50	00000000G	00	FB	000FF		CALLS	#2, EDT\$\$RPL_LN	:	
	0C		00	D0	00106		MOVL	EDT\$\$A_CUR_BUF, R0	:	0865
07 00000000G	00		55	B0	0010D		MOVW	NEW_POS, 12(R0)	:	
00000000G	00		03	E0	00111		BBS	#3, EDT\$\$G_EXE_SBITS, 12\$:	0870
			00	FB	00119		CALLS	#0, EDT\$\$TY_CURLN	:	
			04	00120	12\$:		RET		:	0874

: Routine Size: 289 bytes, Routine Base: _EDT\$CODE + 0000

: 305 0875 1
: 306 0876 1 !<BLF/PAGE>

.....

```

EDT$LSUB          EDT$LSUB - SUBSTITUTE in one line of text      M 13
V04-000          EDT$$$SUB_NXT - substitute in one line of text  16-Sep-1984 01:00:48  VAX-11 Bliss-32 V4.0-742
: 308            0877 1 END                                       14-Sep-1984 12:23:42  DISK$VMSMASTER:[EDT.SRC]LSUB.BLI;1  Page 10
: 309            0878 1                                           (4)
: 310            0879 0 ELUDOM                                     . of module EDT$LSUB

```

PSECT SUMMARY

```

Name          Bytes          Attributes
_EDT$CODE     289 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

```

Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded		
_\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	35	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	7	00:00.1

COMMAND QUALIFIERS

```

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS$:LSUB/OBJ=OBJ$:LSUB MSRC$:LSUB.BLI/UPDATE=(ENH$:LSUB)

```

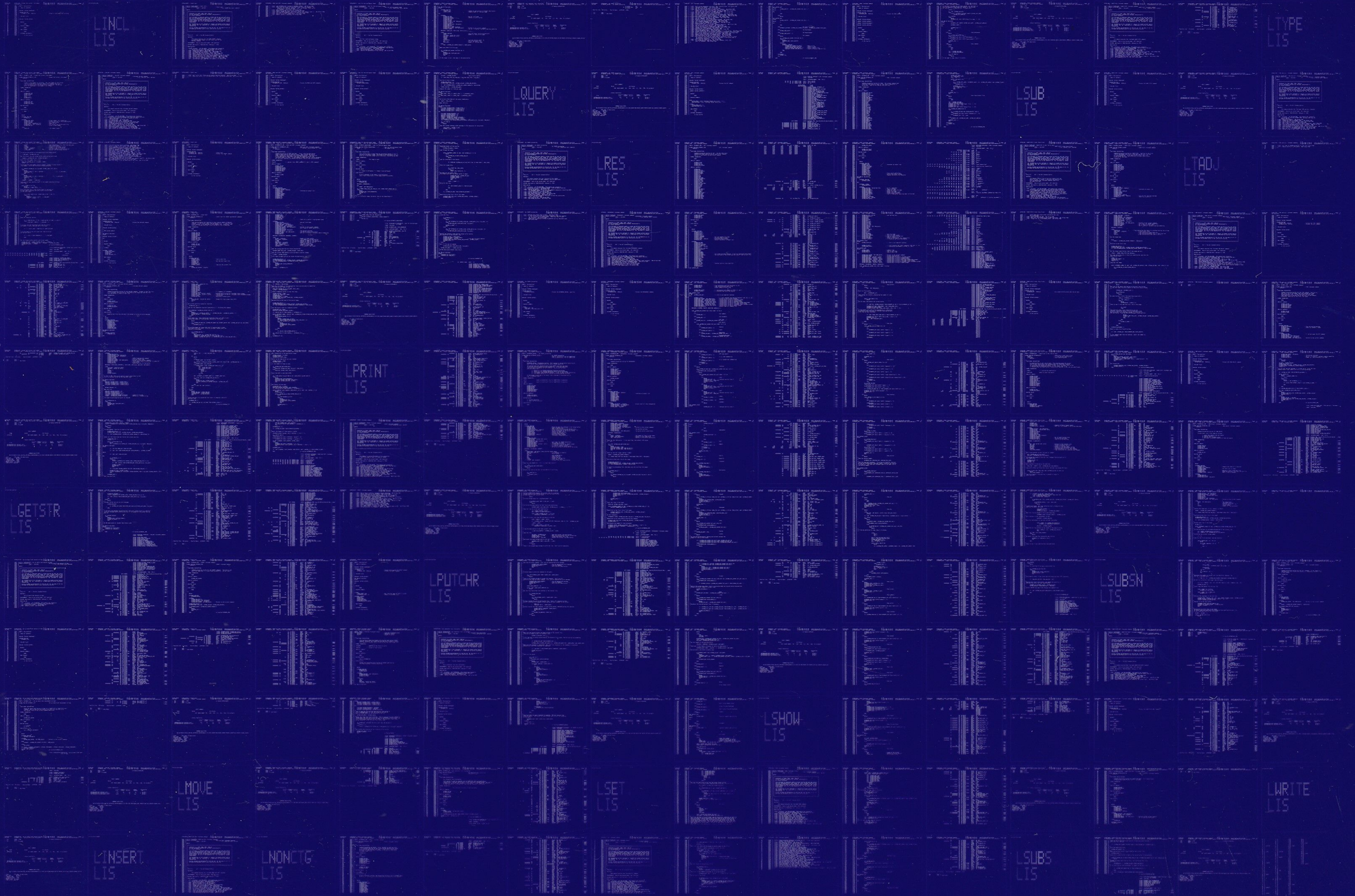
```

: Size:          289 code + 0 data bytes
: Run Time:      00:17.4
: Elapsed Time: 00:22.7
: Lines/CPU Min: 3031
: Lexemes/CPU-Min: 8879
: Memory Used: 120 pages
: Compilation Complete

```

0136 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



LNCL LIS												LTYP LIS
						LQRY LIS				L SUB LIS		
						LRES LIS						LTADJ LIS
				LPRINT LIS								
LGETSTR LIS												
						LPUTCHR LIS				L SUBSN LIS		
								LSHOW LIS				
		LMOVE LIS				LSET LIS						LWRITE LIS
LINSERT LIS		LNONCTG LIS								L SUBS LIS		