


```

CCCCCCCC HH   HH   MM   MM   IIIIII NN   NN   SSSSSSS SSSSSSS TTTTTTTTT RRRRRRR
CCCCCCCC HH   HH   MM   MM   IIIIII NN   NN   SSSSSSS SSSSSSS TTTTTTTTT RRRRRRR
CC        HH   HH   MMMM MMMM   II   NN   NN   SS   SS   SS   SS   TT   TT   RR   RR
CC        HH   HH   MMMM MMMM   II   NN   NN   SS   SS   SS   SS   TT   TT   RR   RR
CC        HH   HH   MM   MM   MM   II   NNNN NN   SS   SS   SS   SS   TT   TT   RR   RR
CC        HH   HH   MM   MM   MM   II   NNNN NN   SS   SS   SS   SS   TT   TT   RR   RR
CC        HHHHHHHHHH MM   MM   II   NN   NN   NN   SSSSSS SSSSSS   TT   TT   RRRRRRR
CC        HHHHHHHHHH MM   MM   II   NN   NN   NN   SSSSSS SSSSSS   TT   TT   RRRRRRR
CC        HH   HH   MM   MM   MM   II   NN   NN   NN   SS   SS   SS   SS   TT   TT   RR   RR
CC        HH   HH   MM   MM   MM   II   NN   NN   NN   SS   SS   SS   SS   TT   TT   RR   RR
CC        HH   HH   MM   MM   MM   II   NN   NN   NN   SS   SS   SS   SS   TT   TT   RR   RR
CC        HH   HH   MM   MM   MM   II   NN   NN   NN   SS   SS   SS   SS   TT   TT   RR   RR
CCCCCCCC HH   HH   MM   MM   IIIIII NN   NN   SSSSSSS SSSSSSS TTTTTTTTT RRRRRRR
CCCCCCCC HH   HH   MM   MM   IIIIII NN   NN   SSSSSSS SSSSSSS TTTTTTTTT RRRRRRR

```

```

LL        IIIIII SSSSSSS
LL        IIIIII SSSSSSS
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
LLLLLLLLL IIIIII SSSSSSS
LLLLLLLLL IIIIII SSSSSSS

```

...

...

...

...

...

```

1 0001 0 %TITLE 'EDT$CHMINSSTR - insert characters'
2 0002 0 MODULE EDT$CHMINSSTR ( ! Insert characters
3 0003 0 IDENT = 'V04-000' ! File: CHMINSSTR.BLI Edit: JBS1006
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 does a (change mode) insertion of a string of
37 0037 1 characters which may include carriage returns.
38 0038 1
39 0039 1 ENVIRONMENT: Runs at any access mode - AST reentrant
40 0040 1
41 0041 1 AUTHOR: Bob Kushlis, CREATION DATE: Unknown
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 1-001 - Original. DJS 04-Feb-1981. This module was created by
46 0046 1 extracting the routine INSERT_STRING from the module CHANGE.BLI.
47 0047 1 1-002 - Regularize headers. JBS 03-Mar-1981
48 0048 1 1-003 - Change SPLIT_LINE to EDT$$$SPLT LNINS. JBS 30-Mar-1981
49 0049 1 1-004 - New screen update logic. JBS T3-Sep-1982
50 0050 1 1-005 - Add parameter to split line routine. SMB 16-Nov-1982
51 0051 1 1-006 - Do special handling for trailing carriage returns. JBS 28-Dec-1982
52 0052 1 --
53 0053 1

```

```
.. 55      0054 1 %SBTTL 'Declarations'  
.. 56      0055 1  
.. 57      0056 1 : TABLE OF CONTENTS:  
.. 58      0057 1 :  
.. 59      0058 1  
.. 60      0059 1 REQUIRE 'EDT$SRC:TRAROUNAM';  
.. 61      0498 1  
.. 62      0499 1 FORWARD ROUTINE  
.. 63      0500 1     EDT$$INS_CHS;  
.. 64      0501 1  
.. 65      0502 1 :  
.. 66      0503 1 : INCLUDE FILES:  
.. 67      0504 1 :  
.. 68      0505 1  
.. 69      0506 1 REQUIRE 'EDT$SRC:EDTREQ';  
.. 70      0641 1  
.. 71      0642 1 :  
.. 72      0643 1 : MACROS:  
.. 73      0644 1 :  
.. 74      0645 1 :     NONE  
.. 75      0646 1 :  
.. 76      0647 1 : EQUATED SYMBOLS:  
.. 77      0648 1 :  
.. 78      0649 1 :     NONE  
.. 79      0650 1 :  
.. 80      0651 1 : OWN STORAGE:  
.. 81      0652 1 :  
.. 82      0653 1 :     NONE  
.. 83      0654 1 :  
.. 84      0655 1 : EXTERNAL REFERENCES:  
.. 85      0656 1 :  
.. 86      0657 1 :     In the routine
```

! Insert a string of characters which may include carriage returns

```

88 0658 1 %SBTTL 'EDT$$INS_CHS - insert characters'
89 0659 1
90 0660 1 GLOBAL ROUTINE EDT$$INS_CHS (          ! Insert characters
91 0661 1     S,                                ! address of string to insert
92 0662 1     L,                                ! length of that string
93 0663 1     ) =
94 0664 1
95 0665 1 !++
96 0666 1 ! FUNCTIONAL DESCRIPTION:
97 0667 1
98 0668 1     Insert a string of characters which may include carriage returns.
99 0669 1     Carriage returns are treated as line terminators, having the effect of
100 0670 1     breaking the current line of text into two.
101 0671 1
102 0672 1 ! FORMAL PARAMETERS:
103 0673 1
104 0674 1     S                a pointer to the character string to insert.
105 0675 1
106 0676 1     L                the length of the string.
107 0677 1
108 0678 1 ! IMPLICIT INPUTS:
109 0679 1
110 0680 1     NONE
111 0681 1
112 0682 1 ! IMPLICIT OUTPUTS:
113 0683 1
114 0684 1     NONE
115 0685 1
116 0686 1 ! ROUTINE VALUE:
117 0687 1
118 0688 1     A value of 0 is returned if a line exceeded 255 characters during the
119 0689 1     insert (further insertions were lost).
120 0690 1
121 0691 1 ! SIDE EFFECTS:
122 0692 1
123 0693 1     NONE
124 0694 1
125 0695 1 !--
126 0696 1
127 0697 2     BEGIN
128 0698 2
129 0699 2     EXTERNAL ROUTINE
130 0700 2         EDT$$INS_STR,                ! Insert a string of characters at the current position
131 0701 2         EDT$$SPLY_LNINS : NOVALUE,    ! Split a line of text at the current cursor position
132 0702 2         EDT$$CS_RIGHT,              ! Move right a character
133 0703 2         EDT$$CS_LEFT;               ! Move left a character
134 0704 2
135 0705 2     LOCAL
136 0706 2         END_STRING,
137 0707 2         POINT,
138 0708 2         MY_S,
139 0709 2         MY_L;
140 0710 2
141 0711 2     MY_L = .L;
142 0712 2 !+
143 0713 2 ! Initialize a pointer to the current character and one to the end of the
144 0714 2 ! string.

```

B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
[
\
]
^
_
`
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
{|}~

```
145 0715 2 :-  
146 0716 2 POINT = .S;  
147 0717 2 END_STRING = CH$PLUS (.S, .MY_L);  
148 0718 2 :-  
149 0719 2 :-+ If we have trailing carriage returns do the line splitting before the insertion  
150 0720 2 :- to improve screen repainting.  
151 0721 2 :-  
152 0722 2 :-  
153 0723 2 IF ((.MY_L GTR 1) AND (CH$RCHAR (CH$PLUS (.POINT, .MY_L - 1)) EQL ASC_K_CR))  
154 0724 2 THEN  
155 0725 2 BEGIN  
156 0726 2  
157 0727 2 LOCAL  
158 0728 2 LINES_INSERTED,  
159 0729 2 STATUS;  
160 0730 2  
161 0731 2 :-+  
162 0732 2 :- Insert enough empty lines to account for the carriage returns.  
163 0733 2 :- Keep backing up so our position doesn't change.  
164 0734 2 :-  
165 0735 2 LINES_INSERTED ;  
166 0736 2  
167 0737 2 WHILE ((.MY_L GTR 1) AND (CH$RCHAR (CH$PLUS (.POINT, .MY_L - 1)) EQL ASC_K_CR)) DO  
168 0738 2 BEGIN  
169 0739 2 EDT$$SPLT LNINS (1);  
170 0740 2 EDT$$CS_LEFT ();  
171 0741 2 MY_L = .MY_L - 1;  
172 0742 2 LINES_INSERTED = .LINES_INSERTED + 1;  
173 0743 2 END;  
174 0744 2  
175 0745 2 :-+  
176 0746 2 :- Insert the text without the trailing carriage returns.  
177 0747 2 :-  
178 0748 2 STATUS = EDT$$INS_CHS (.POINT, .MY_L);  
179 0749 2 :-+  
180 0750 2 :- Now move forward over those blank lines.  
181 0751 2 :-  
182 0752 2 :-  
183 0753 2 DECR I FROM .LINES_INSERTED - 1 TO 0 DO  
184 0754 2 EDT$$CS_RIGHT (I);  
185 0755 2  
186 0756 2 :-+  
187 0757 2 :- We are done with the insert.  
188 0758 2 :-  
189 0759 2 RETURN (.STATUS);  
190 0760 2 END;  
191 0761 2  
192 0762 2 :-+  
193 0763 2 :- Either this is a string of length one or there is no  
194 0764 2 :- trailing carriage returns. Loop until no more characters remain.  
195 0765 2 :-  
196 0766 2 :-  
197 0767 2 WHILE CH$PTR_NEQ (.END_STRING, .POINT) DO  
198 0768 2 BEGIN  
199 0769 2  
200 0770 2 IF (CH$RCHAR (.POINT) EQL ASC_K_CR)  
201 0771 2 THEN
```

```

202 0772 3 !+
203 0773 3 !- The character is a carriage return: split the line.
204 0774 3 !-
205 0775 4 BEGIN
206 0776 4 EDT$$$SPLT_LNINS (1); ! Use length optimizer
207 0777 4 POINT = CR$PLUS (.POINT, 1);
208 0778 4 END
209 0779 3 ELSE
210 0780 4 BEGIN
211 0781 4 !+
212 0782 4 !- The character is not a carriage return. Insert the string of
213 0783 4 characters up to the end or to the next carriage return.
214 0784 4 !-
215 0785 4 MY_S = .POINT;
216 0786 4
217 0787 4 WHILE (CH$PTR NEQ (.END_STRING, .POINT) AND (CH$RCHAR (.POINT) NEQ ASC_K_CR)) DO
218 0788 4 POINT = CR$PLUS (.POINT, 1);
219 0789 4
220 0790 4 IF (EDT$$INS_STR (.MY_S, CH$DIFF (.POINT, .MY_S)) EQL 0) THEN RETURN (0);
221 0791 4
222 0792 3 END;
223 0793 3
224 0794 2 END;
225 0795 2
226 0796 2 RETURN (1);
227 0797 1 END;

```

! of routine EDT\$\$INS_CHS

```

.TITLE EDT$CHMINSSTR EDT$CHMINSSTR - insert characters
.IDENT \V04-000\

.EXTRN EDT$$INS_STR, EDT$$$SPLT_LNINS
.EXTRN EDT$$CS_RIGHT, EDT$$CS_LEFT

```

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

```

                    007C 00000
56 00000000G 00 9E 00002 .ENTRY EDT$$INS_CHS, Save R2,R3,R4,R5,R6 ; 0660
53 08 AC D0 00009 MOVAB EDT$$$SPLT_LNINS, R6 ; 0711
54 04 AC D0 0000D MOVL L, MY_L ; 0716
53 04 AC C1 00011 ADDL3 S, POINT ; 0717
01 53 D1 00016 CMPL MY_L, #1 ; 0723
42 15 00019 BLEQ 5$ ;
0D FF A344 91 0001B CMPB -1(MY_L)[POINT], #13 ;
38 12 00020 BNEQ 5$ ;
52 D4 00022 CLRL LINES_INSERTED ; 0735
01 53 D1 00024 1$: CMPL MY_L, #1 ; 0737
19 15 00027 BLEQ 2$ ;
0D FF A344 91 00029 CMPB -1(MY_L)[POINT], #13 ;
12 12 0002E BNEQ 2$ ;
01 DD 00030 PUSHL #1 ; 0739
66 01 FB 00032 CALLS #1, EDT$$$SPLT_LNINS ;
00000000G 00 00 FB 00035 CALLS #0, EDT$$CS_LEFT ; 0740
53 D7 0003C DECL MY_L ; 0741
52 D6 0003E INCL LINES_INSERTED ; 0742
E2 11 00040 BRB 1$ ; 0737
53 DD 00042 2$: PUSHL MY_L ; 0748

```

		54	DD	00044	PUSHL	POINT	
	B6	AF	02	FB	00046	CALLS	#2, EDT\$\$INS_CHS
		53	50	DO	0004A	MOVL	R0, STATUS
			07	11	0004D	BRB	4\$
	000J0000G	00	00	FB	0004F	3\$: CALLS	#0, EDT\$\$CS_RIGHT
		F6	52	F4	00056	4\$: SOBGEQ	I, 3\$
		50	53	DO	00059	MOVL	STATUS, R0
				04	0005C	RET	
		54	55	D1	0005D	5\$: CMPL	END_STRING, POINT
			32	13	00060	BEQL	9\$
	0D		64	91	00062	CMPB	(POINT), #13
			09	12	00065	BNEQ	6\$
			01	DD	00067	PUSHL	#1
		66	01	FB	00069	CALLS	#1, EDT\$\$SPLT_LNINS
			54	D6	0006C	INCL	POINT
			ED	11	0006E	BRB	5\$
		52	54	DO	00070	6\$: MOVL	POINT, MY_S
		54	55	D1	00073	7\$: CMPL	END_STRING, POINT
			09	13	00076	BEQL	8\$
		0D	64	91	00078	CMPB	(POINT), #13
			04	13	0007B	BEQL	8\$
			54	D6	0007D	INCL	POINT
			F2	11	0007F	BRB	7\$
	7E		52	C3	00081	8\$: SUBL3	MY_S, POINT, -(SP)
			52	DD	00085	PUSHL	MY_S
	00000000G	00	02	FB	00087	CALLS	#2, EDT\$\$INS_STR
			50	D5	0008E	TSTL	R0
			CB	12	00090	BNEQ	5\$
			04	11	00092	BRB	10\$
		50	01	DO	00094	9\$: MOVL	#1, R0
				04	00097	RET	
			50	D4	00098	10\$: CLRL	R0
				04	0009A	RET	

; Routine Size: 155 bytes, Routine Base: _EDT\$CODE + 0000

: 228 0798 1
: 229 0799 1 !<BLF/PAGE>

EDT\$CHMINSSTR
V04-000

EDT\$CHMINSSTR - insert characters
EDT\$\$INS_CHS - insert characters

E 16
16-Sep-1984 00:02:23 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:22:37 [EDT.SRC]CHMINSSTR.BLI;1

Page 7
(4)

: 231 0800 1 END
: 232 0801 1
: 233 0802 0 ELUDOM

. of module EDT\$CHMINSSTR

PSECT SUMMARY

Name Bytes Attributes
_EDT\$CODE 155 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	2	0	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:CHMINSSTR/OBJ=OBJ\$:CHMINSSTR MSRC\$:CHMINSSTR.BLI/UPDATE=(ENHS:C
HMINSSTR)

: Size: 155 code + 0 data bytes
: Run Time: 00:13.5
: Elapsed Time: 00:16.7
: Lines/CPU Min: 3577
: Lexemes/CPU-Min: 10407
: Memory Used: 91 pages
: Compilation Complete

CHMFINENT LIS	CHMINIT LIS
CHMGOUNT LIS	CHMGINSTR LIS
CHMGSUSTR LIS	CHMINSMOD LIS
CHMEMESS LIS	CHMINSTAB LIS
CHMENTRM LIS	CHMINSCHR LIS
CHMEXVERB LIS	CHMINDATE LIS
CHMFINSTR LIS	CHMGDTR LIS
CHMGQSTR LIS	CHMLPKPD LIS
CHMINSSTR LIS	CHMKEYWRD LIS
CHMENDWRD LIS	CHMEXCOM LIS