


```

CCCCCCCCC HH      HH MM      MM      SSSSSSSS UU      UU BBBB BBBB SSSSSSSS
CCCCCCCCC HH      HH MM      MM      SSSSSSSS UU      UU BBBB BBBB SSSSSSSS
CC        HH      HH MMMM    MMMM    SS      UU      UU BB      BB SS
CC        HH      HH MMMM    MMMM    SS      UU      UU BB      BB SS
CC        HH      HH MM      MM      SS      UU      UU BB      BB SS
CC        HH      HH MM      MM      SS      UU      UU BB      BB SS
CC        HHHHHHHHHH MM      MM      SSSSSS    UU      UU BBBB BBBB SSSSSS
CC        HHHHHHHHHH MM      MM      SSSSSS    UU      UU BBBB BBBB SSSSSS
CC        HH      HH MM      MM      SS      UU      UU BB      BB SS
CC        HH      HH MM      MM      SS      UU      UU BB      BB SS
CC        HH      HH MM      MM      SS      UU      UU BB      BB SS
CC        HH      HH MM      MM      SS      UU      UU BB      BB SS
CCCCCCCCC HH      HH MM      MM      SSSSSSSS UU      UU BBBB BBBB SSSSSSSS
CCCCCCCCC HH      HH MM      MM      SSSSSSSS UU      UU BBBB BBBB SSSSSSSS
...
...
...
...

```



```
0001 0 %TITLE 'EDT$CHMSUBS - change mode SUBSTITUTE command'
0002 0 MODULE EDT$CHMSUBS ( ! Change mode SUBSTITUTE command
0003 0 IDENT = 'V04-000' ! File: CHMSUBS.BLI Edit: JBS1008
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This module executes the change mode SUBSTITUTE (S) command.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: Unknown
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 04-Feb-1981. This module was created by
0045 1 extracting the routine EDT$$SUB_CMD from module CHANGE.BLI.
0046 1 1-002 - Regularize headers. JBS 04-Mar-1981
0047 1 1-003 - Use new message codes. JBS 04-Aug-1981
0048 1 1-004 - New cursor position on backwards substitute. SMB 28-Oct-1981
0049 1 1-005 - Delete initial blank line by running through PRETTY. JBS 02-Nov-1981
0050 1 1-006 - Rework edit 1-004 to accomplish the same thing without changing
0051 1 the cursor position from V2. SMB 17-Aug-1982
0052 1 1-007 - Remove EDT$$G_LN_NO for new screen update logic. JBS 29-Sep-1982
0053 1 1-008 - Pass the correct number of lines to EDT$$DEL_TXTLN. JBS 08-Oct-1982
0054 1 --
0055 1
```


EDT\$CHMSUBS
V04-000

EDT\$CHMSUBS - change mode SUBSTITUTE command
Declarations

C 13
16-Sep-1984 00:12:59
14-Sep-1984 12:22:47

VAX-11 BLISS-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]CHMSUBS.BLI;1 Page 2 (2)

```
: 57      0056 1 %SBTTL 'Declarations'
: 58      0057 1
: 59      0058 1 | TABLE OF CONTENTS:
: 60      0059 1 |
: 61      0060 1
: 62      0061 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 63      0500 1
: 64      0501 1 FORWARD ROUTINE
: 65      0502 1     EDT$$SUB_CMD;
: 66      0503 1
: 67      0504 1 |
: 68      0505 1 | INCLUDE FILES:
: 69      0506 1 |
: 70      0507 1
: 71      0508 1 REQUIRE 'EDTSRC:EDTREQ';
: 72      0643 1
: 73      0644 1 |
: 74      0645 1 | MACROS:
: 75      0646 1
: 76      0647 1 |     NONE
: 77      0648 1 |
: 78      0649 1 | EQUATED SYMBOLS:
: 79      0650 1 |
: 80      0651 1 |     NONE
: 81      0652 1 |
: 82      0653 1 | OWN STORAGE:
: 83      0654 1 |
: 84      0655 1 |     NONE
: 85      0656 1 |
: 86      0657 1 | EXTERNAL REFERENCES:
: 87      0658 1 |
: 88      0659 1 |     In the routine
```

! Execute the SUBSTITUTE command

EDT
V04

S
R
E
L
M
C

```

90      0660 1 %SBTTL 'EDT$$$SUB_CMD - change mode SUBSTITUTE command'
91      0661 1
92      0662 1 GLOBAL ROUTINE EDT$$$SUB_CMD ! Change mode SUBSTITUTE command
93      0663 1 =
94      0664 1
95      0665 1 !++
96      0666 1 FUNCTIONAL DESCRIPTION:
97      0667 1
98      0668 1 This routine implements the SUBSTITUTE command. Just search for the
99      0669 1 search string, delete it and insert the replace string.
100     0670 1
101     0671 1 FORMAL PARAMETERS:
102     0672 1
103     0673 1 NONE
104     0674 1
105     0675 1 IMPLICIT INPUTS:
106     0676 1
107     0677 1 EDT$$G_DIR
108     0678 1 EDT$$A_LN_PTR
109     0679 1 EDT$$G_SEA_STRLEN
110     0680 1 EDT$$T_SEA_STR
111     0681 1 EDT$$G_SUB_STRLEN
112     0682 1 EDT$$T_SUB_STR
113     0683 1
114     0684 1 IMPLICIT OUTPUTS:
115     0685 1
116     0686 1 NONE
117     0687 1
118     0688 1 ROUTINE VALUE:
119     0689 1
120     0690 1 1 = substitution made, 0 = not made (control C or some error)
121     0691 1
122     0692 1 SIDE EFFECTS:
123     0693 1
124     0694 1 NONE
125     0695 1
126     0696 1 !--
127     0697 1
128     0698 2 BEGIN
129     0699 2
130     0700 2 EXTERNAL ROUTINE
131     0701 2 EDT$$DEL_TXTLN : NOVALUE, ! Delete whole lines of text
132     0702 2 EDT$$INS_CHS, ! Insert a string of characters which may include carriage returns
133     0703 2 EDT$$MSG_BELL : NOVALUE, ! Output a message to the terminal with a warning bell
134     0704 2 EDT$$STR_SEACMD, ! Search for a specific string
135     0705 2 EDT$$CS_LEFT, ! Move left a character
136     0706 2 EDT$$CS_RIGHT; ! Move right a character
137     0707 2
138     0708 2 EXTERNAL
139     0709 2 EDT$$G_DIR, ! The current direction.
140     0710 2 EDT$$A_LN_PTR, ! Current character pointer
141     0711 2 EDT$$G_SEA_STRLEN, ! Length of current search string
142     0712 2 EDT$$T_SEA_STR, ! Current search string
143     0713 2 EDT$$G_SUB_STRLEN, ! Length of subs string
144     0714 2 EDT$$T_SUB_STR, ! Current SUBSTITUTE string
145     0715 2 EDT$$A_CUR_BUF : REF TBCB_BLOCK; ! Current text buffer control block
146     0716 2

```



```
147 0717 2  MESSAGES ((STRNOTFND, BADRANGE));
148 0718 2
149 0719 2  LOCAL
150 0720 2  STS,
151 0721 2  NUM_LINES : LN_BLOCK,
152 0722 2  END_POS;
153 0723 2
154 0724 2  !+
155 0725 2  !- Search for the string.
156 0726 2  !-
157 0727 2
158 0728 2  IF (.EDT$$G_DIR EQL DIR_BACKWARD)
159 0729 2  THEN
160 0730 2  !+
161 0731 2  !- Move back one position before starting the search
162 0732 2  !-
163 0733 2  STS = EDT$$STR_SEACMD (EDT$$T_SEA_STR, .EDT$$G_SEA_STRLEN, 1, .EDT$$G_DIR)
164 0734 2  ELSE
165 0735 2  STS = EDT$$STR_SEACMD (EDT$$T_SEA_STR, .EDT$$G_SEA_STRLEN, 0, .EDT$$G_DIR);
166 0736 2
167 0737 2  IF (.STS EQL 1)
168 0738 2  THEN
169 0739 2  !+
170 0740 2  !- String was found, do the SUBSTITUTE.
171 0741 2  !-
172 0742 2  BEGIN
173 0743 2  !+
174 0744 2  !- Move to the end of the string, counting the lines involved.
175 0745 2  !-
176 0746 2  MOVELINE (EDT$$A_CUR_BUF [TBCB_CUR_LIN], NUM_LINES);
177 0747 2
178 0748 2  DECR I FROM .EDT$$G_SEA_STRLEN - 1 TO 0 DO
179 0749 2  EDT$$CS_RIGHT ();
180 0750 2
181 0751 2  SUBLINE (NUM_LINES, EDT$$A_CUR_BUF [TBCB_CUR_LIN], NUM_LINES);
182 0752 2  ADDLINE (NUMBER ONE, NUM_LINES);
183 0753 2  END_POS = .EDT$$A_LN_PTR;
184 0754 2  !+
185 0755 2  !- Move back to the beginning.
186 0756 2  !-
187 0757 2
188 0758 2  DECR I FROM .EDT$$G_SEA_STRLEN - 1 TO 0 DO
189 0759 2  EDT$$CS_LEFT ();
190 0760 2
191 0761 2  !+
192 0762 2  !- Make sure the string isn't too long.
193 0763 2  !-
194 0764 2
195 0765 4  IF ((.NUM_LINES [LN_HI] NEQU 0) OR (.NUM_LINES [LN_MD] NEQU 0))
196 0766 3  THEN
197 0767 4  BEGIN
198 0768 4  EDT$$MSG_BELL (EDT$_BADRANGE);
199 0769 4  RETURN (0);
200 0770 3  END;
201 0771 3
202 0772 3  !+
203 0773 3  !- Delete the string.
```

```
204 0774 3 !-
205 0775 3 EDT$$DEL_TXTLN (.NUM_LINES [LN_LO], .END_POS);
206 0776 3 !+
207 0777 3 Insert the replacement string.
208 0778 3 !-
209 0779 3 EDT$$INS_CHS (EDT$$T_SUB_STR, .EDT$$G_SUB_STRLEN);
210 0780 3 !+
211 0781 3 If we are moving backward, then move back across the string
212 0782 3 !-
213 0783 3
214 0784 3 IF (.EDT$$G_DIR EQL DIR_BACKWARD)
215 0785 3 THEN
216 0786 3
217 0787 3     DECR I FROM .EDT$$G_SUB_STRLEN TO 1 DO
218 0788 3         EDT$$CS_LEFT ();
219 0789 3
220 0790 3     RETURN (1);
221 0791 3     END
222 0792 3 ELSE
223 0793 3     BEGIN
224 0794 3 !+
225 0795 3 String was not found, EDT$$MSG_BELL an error.
226 0796 3 !-
227 0797 3
228 0798 3 IF (.STS EQL 0) THEN EDT$$MSG_BELL (EDT$_STRNOTFND);
229 0799 3
230 0800 3 RETURN (0);
231 0801 3 END;
232 0802 3
233 0803 1 END;
```

! of routine EDT\$\$\$SUB_CMD

```
.TITLE EDT$CHMSUBS EDT$CHMSUBS - change mode SUBSTITUT
E command
```

```
.IDENT \V04-000\
```

```
.EXTRN EDT$$DEL_TXTLN, EDT$$INS_CHS
.EXTRN EDT$$MSG_BELL, EDT$$STR_SEACMD
.EXTRN EDT$$CS_LEFT, EDT$$CS_RIGHT
.EXTRN EDT$$G_DIR, EDT$$A_LN_PTR
.EXTRN EDT$$G_SEA_STRLEN
.EXTRN EDT$$T_SEA_STR, EDT$$G_SUB_STRLEN
.EXTRN EDT$$T_SUB_STR, EDT$$A_CUR_BUF
.EXTRN EDT$_STRNOTFND, EDT$_BADRANGE
```

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

```
.ENTRY EDT$$$SUB_CMD, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0662
R10
```

```
07FC 00000
5A 00000000G 00 9E 00002
59 00000000G 00 9E 00009
58 00000000G 00 9E 00010
57 00000000G 00 9E 00017
56 00000000G 00 9E 0001E
5E 00000000G 08 C2 00025
51 00000000G 66 D0 00028
50 00000000G 67 D0 0002B
```

```
MOVAB EDT$$G_SUB_STRLEN, R10
MOVAB EDT$$CS_LEFT, R9
MOVAB EDT$$A_CUR_BUF, R8
MOVAB EDT$$G_DIR, R7
MOVAB EDT$$G_SEA_STRLEN, R6
SUBL2 #8, SP
MOVL EDT$$G_SEA_STRLEN, R1
MOVL EDT$$G_DIR, R0
```

```
0733
0728
```


		06	12	0002E	BNEQ	1\$	:	
		50	DD	00030	PUSHL	R0	:	0733
		01	DD	00032	PUSHL	#1	:	
		04	11	00034	BRB	2\$	:	
		50	DD	00036	PUSHL	R0	:	0735
		7E	D4	00038	CLRL	-(SP)	:	
		51	DD	0003A	PUSHL	R1	:	
	00000000G	00	9F	0003C	PUSHAB	EDT\$\$T_SEA_STR	:	
	01	04	FB	00042	CALLS	#4, EDT\$\$STR_SEACMD	:	
		50	D1	00049	CMPL	STS, #1	:	0737
		03	13	0004C	BEQL	3\$	:	
		008E	31	0004E	BRW	14\$	:	
	50	68	D0	00051	MOVL	EDT\$\$A_CUR_BUF, R0	:	0746
6E	06	A0	06	28	MOVC3	#6, 6(R0), NUM_LINES	:	
	52	66	D0	00059	MOVL	EDT\$\$G_SEA_STRLEN, I	:	0748
		07	11	0005C	BRB	5\$	:	
	00000000G	00	FB	0005E	CALLS	#0, EDT\$\$CS_RIGHT	:	0749
	F6	52	F4	00065	SOBGEQ	I, 4\$	:	
	51	06	AE	B0	MOVW	UPPER_WORD, SAVE	:	0751
	50	68	D0	0006C	MOVL	EDT\$\$A_CUR_BUF, R0	:	
6E	06	A0	6E	C3	SUBL3	NUM_LINES, 6(R0), NUM_LINES	:	
	52	0A	A0	D0	MOVL	10(R0), R2	:	
	52	04	AE	D9	SBWC	NUM_LINES, R2	:	
	04	AE	52	D0	MOVL	R2, NUM_LINES	:	
	06	AE	51	B0	MOVW	SAVE, UPPER_WORD	:	
			6E	D6	INCL	FIRST_LWORD	:	0752
			03	12	BNEQ	6\$	:	
		04	AE	B6	INCW	NEXT_WORD	:	
	53	00000000G	00	D0	MOVL	EDT\$\$A_LN_PTR, END_POS	:	0753
	52		66	D0	MOVL	EDT\$\$G_SEA_STRLEN, I	:	0758
			03	11	BRB	8\$	:	
	69		00	FB	CALLS	#0, EDT\$\$CS_LEFT	:	0759
	FA		52	F4	SOBGEQ	I, 7\$	:	
		04	AE	B5	TSTW	NUM_LINES+4	:	0765
		05	12	000A0	BNEQ	9\$	:	
		02	AE	B5	TSTW	NUM_LINES+2	:	
		08	13	000A5	BEQL	10\$	:	
		00000000G	8F	DD	PUSHL	#EDT\$_BADRANGE	:	0768
			3A	11	BRB	15\$	:	
			53	DD	PUSHL	END_POS	:	0775
	7E	04	AE	3C	MOVZWL	NUM_LINES, -(SP)	:	
00000000G	00		02	FB	CALLS	#2, EDT\$\$DEL_TXTLN	:	
			6A	DD	PUSHL	EDT\$\$G_SUB_STRLEN	:	0779
	00000000G	00	00	9F	PUSHAB	EDT\$\$T_SUB_STR	:	
			02	FB	CALLS	#2, EDT\$\$INS_CHS	:	
			67	D5	TSTL	EDT\$\$G_DIR	:	0784
			0C	12	BNEQ	13\$	:	
52	6A		01	C1	ADDL3	#1, EDT\$\$G_SUB_STRLEN, I	:	0787
			03	11	BRB	12\$	:	
	69		00	FB	CALLS	#0, EDT\$\$CS_LEFT	:	0788
	FA		52	F5	SOBGR	I, 11\$	:	
	50		01	D0	MOVL	#1, R0	:	0793
				04	RET		:	
			50	D5	TSTL	STS	:	0798
			0D	12	BNEQ	16\$	:	
	00000000G	00	8F	DD	PUSHL	#EDT\$_STRNOTFND	:	
			01	FB	CALLS	#1, EDT\$\$MSG_BELL	:	

EDT\$CHMSUBS
V04-000

EDT\$CHMSUBS - change mode SUBSTITUTE command
EDT\$\$SUB_CMD - change mode SUBSTITUTE command

H 13
16-Sep-1984 00:12:59
14-Sep-1984 12:22:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]CHMSUBS.BLI;1 Page 7 (3)

50 04 000F0 16\$: CLRL R0
04 000F2 RET

; 0803
;

; Routine Size: 243 bytes, Routine Base: _EDT\$CODE + 0000

; 234 0804 1
; 235 0805 1 !<BLF/PAGE>

EDT
V04

```

: 237      0806 1 END
: 238      0807 1
: 239      0808 0 ELUDOM

```

```
! of module EDT$CHMSUBS
```

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	243	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	35	9	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$:CHMSUBS/OBJ=OBJ$:CHMSUBS MSRC$:CHMSUBS.BLI/UPDATE=(ENH$:CHMSUBS
;      )

```

```
: Size:          243 code + 0 data bytes
: Run Time:      00:15.4
: Elapsed Time:  00:19.0
: Lines/CPU Min: 3146
: Lexemes/CPU-Min: 11937
: Memory Used:   117 pages
: Compilation Complete
```


0132 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

