


```

CCCCCCCC HH   HH   MM   MM   GGGGGGGG BBBB8888 UU   UU   FFFFFFFFFF
CCCCCCCC HH   HH   MM   MM   GGGGGGGG BBBB8888 UU   UU   FFFFFFFFFF
CC        HH   HH   MMMM MMMM GG        BB   BB   UU   UU   FF
CC        HH   HH   MMMM MMMM GG        BB   BB   UU   UU   FF
CC        HH   HH   MM   MM   GG        BB   BB   UU   UU   FF
CC        HH   HH   MM   MM   GG        BB   BB   UU   UU   FF
CC        HHHHHHHHHH MM   MM   GG        BBBB8888 UU   UU   FFFFFFFF
CC        HHHHHHHHHH MM   MM   GG        BBBB8888 UU   UU   FFFFFFFF
CC        HH   HH   MM   MM   GG   GGGGGG BB   BB   UU   UU   FF
CC        HH   HH   MM   MM   GG   GGGGGG BB   BB   UU   UU   FF
CC        HH   HH   MM   MM   GG        GG   GG   UU   UU   FF
CC        HH   HH   MM   MM   GG        GG   GG   UU   UU   FF
CCCCCCCC HH   HH   MM   MM   GGGGGG   BBBB8888 UUUUUUUUUU FF
CCCCCCCC HH   HH   MM   MM   GGGGGG   BBBB8888 UUUUUUUUUU FF

```

```

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

```

```

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
LLLLLLLLLL IIIIII   SSSSSSSS
LLLLLLLLLL IIIIII   SSSSSSSS

```

EC
VC

```

.
.
.
.

```

```

1 0001 0 %TITLE 'EDT$CHMGBUF - look for a buffer name'
2 0002 0 MODULE EDT$CHMGBUF ( ! Look for a buffer name
3 0003 0 IDENT = 'V04-000' ! File: CHMGBUF.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 scans the command buffer for a buffer name.
37 0037 1
38 0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Bob Kushlis, CREATION DATE: Unknown
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. DJS 04-Feb-1981. This module was created by
45 0045 1 extracting the routine EDT$$GET_BUF from the module CHANGE.BLI.
46 0046 1 1-002 - Regularize headers. JBS 02-Mar-1981
47 0047 1 1-003 - Use new message codes. JBS 04-Aug-1981
48 0048 1 1-004 - Check first to see if there are more characters on command line.
49 0049 1 STS 21-Jun-1982
50 0050 1 1-005 - Change alphabetic test. JBS 19-Jul-1982
51 0051 1 1-006 - Add conditional for VT220 support. JBS 11-Feb-1983
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$$GET_BUF : NOVALUE;  
.. 64 0501 1  
.. 65 0502 1 :  
.. 66 0503 1 : INCLUDE FILES:  
.. 67 0504 1 :  
.. 68 0505 1  
.. 69 0506 1 REQUIRE 'EDT$SRC:EDTREQ';  
.. 70 0641 1  
.. 71 0642 1 LIBRARY 'EDT$SRC:SUPPORTS';  
.. 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
```

! Look for a buffer specification for the cut, paste, and append commands

```

90 0660 1 %SBTTL 'EDT$$GET_BUF - look for a buffer name'
91 0661 1
92 0662 1 GLOBAL ROUTINE EDT$$GET_BUF ! Look for a buffer name
93 0663 1 : NOVALUE =
94 0664 1
95 0665 1 !++
96 0666 1 FUNCTIONAL DESCRIPTION:
97 0667 1
98 0668 1 This routine looks for a buffer spec for use by cut paste and append.
99 0669 1 If an '=' is seen, an explicit buffer spec is there, otherwise, use
100 0670 1 the paste buffer. The routine places the address of the buffer in
101 0671 1 EDT$$A_ALT_BUF , with a zero indicating failure.
102 0672 1
103 0673 1 FORMAL PARAMETERS:
104 0674 1
105 0675 1 NONE
106 0676 1
107 0677 1 IMPLICIT INPUTS:
108 0678 1
109 0679 1 EDT$$A_CMD_END
110 0680 1 EDT$$A_CMD_BUF
111 0681 1 EDT$$A_PST_BUF
112 0682 1 EDT$$A_CUR_BUF
113 0683 1
114 0684 1 IMPLICIT OUTPUTS:
115 0685 1
116 0686 1 EDT$$A_ALT_BUF
117 0687 1 EDT$$A_CMD_BUF
118 0688 1
119 0689 1 ROUTINE VALUE:
120 0690 1
121 0691 1 1 = buffer spec seen, 0 = no buffer spec seen
122 0692 1
123 0693 1 SIDE EFFECTS:
124 0694 1
125 0695 1 Uses EDT$$FND_BUF to create the buffer if necessary. If it fails,
126 0696 1 gives an error message.
127 0697 1
128 0698 1 --
129 0699 1
130 0700 2 BEGIN
131 0701 2
132 0702 2 EXTERNAL ROUTINE
133 0703 2 EDT$$MSG_BELL : NOVALUE, ! Output a message to the terminal with a warning bell
134 0704 2 EDT$$CNV_UPC, ! Convert string to upper case
135 0705 2 EDT$$FND_BUF, ! Search for a text buffer
136 0706 2 EDT$$RD_CURLN; ! Get the current line
137 0707 2
138 0708 2 EXTERNAL
139 0709 2 EDT$$A_CMD_END, ! End of command pointer
140 0710 2 EDT$$A_CMD_BUF, ! Command string pointer
141 0711 2 EDT$$A_ALT_BUF : REF TBCB_BLOCK, ! Alternate buffer used for cut/paste.
142 0712 2 EDT$$A_PST_BUF : REF TBCB_BLOCK, ! The paste buffer tbc
143 0713 2 EDT$$A_CUR_BUF : REF TBCB_BLOCK ! The current buffer tbc
144 0714 2
145 L 0715 2 %IF SUPPORT_VT220
146 0716 2 %THEN
```

```
0717 2
0718 2 EDT$$B_CHAR_INFO : BLOCKVECTOR [256, 1, BYTE] ! Information about characters
0719 2 %FI
0720 2
0721 2 ;
0722 2
0723 2 MESSAGES ((INVBUFNAM, INSMEM));
0724 2
0725 2 LOCAL
0726 2 SAVE_TCB,
0727 2 NEW,
0728 2 NAME_LEN,
0729 2 NAME_START,
0730 2 CHAR;
0731 2
0732 2 NAME_LEN = 0;
0733 2 CHAR = CH$RCHAR (.EDT$$A_CMD_BUF);
0734 2
0735 2 IF (.CHAR EQL %C'=') AND (.EDT$$A_CMD_BUF LSSU .EDT$$A_CMD_END)
0736 2 THEN
0737 2 BEGIN
0738 2 EDT$$A_CMD_BUF = CH$PLUS (.EDT$$A_CMD_BUF, 1);
0739 2 NAME_START = .EDT$$A_CMD_BUF;
0740 2
0741 2 WHILE CH$PTR_LSS (.EDT$$A_CMD_BUF, .EDT$$A_CMD_END) DO
0742 2 BEGIN
0743 2 CHAR = CH$RCHAR (.EDT$$A_CMD_BUF);
0744 2
0745 2 %IF SUPPORT_VT220
0746 2 %THEN
0747 2
0748 2 IF ((.EDT$$B_CHAR_INFO [.CHAR, 0, 0, 2, 0] EQL 0) ! Not alphabetic
0749 2 AND (.EDT$$B_CHAR_INFO [.CHAR, 0, 0, 8, 0] NEQ %X'FO') ! Not numeric
0750 2 AND (.CHAR NEQ %C'_') ! Not an underscore
0751 2 )
0752 2 THEN
0753 2 EXITLOOP;
0754 2
0755 2 %ELSE
0756 2
0757 2 IF (((.CHAR LSS %C'0') OR !
0758 2 ((.CHAR GTR %C'9') AND (.CHAR LSS %C'A')) OR !
0759 2 ((.CHAR GTR %C'Z') AND (.CHAR LSS %C'a')) OR !
0760 2 (.CHAR GTR %C'z')) AND (.CHAR NEQ %C'_'))
0761 2 THEN
0762 2 EXITLOOP;
0763 2
0764 2 %FI
0765 2
0766 2 NAME_LEN = .NAME_LEN + 1;
0767 2 EDT$$A_CMD_BUF = CH$PLUS (.EDT$$A_CMD_BUF, 1);
0768 2 END;
0769 2
0770 2 CHAR = CH$RCHAR (.NAME_START);
0771 2
0772 2 %IF SUPPORT_VT220
0773 2 %THEN
```

```

204 0774 3
205 0775 4 IF ((.NAME_LEN EQL 0) OR (.EDT$$B_CHAR_INFO [.CHAR, 0, 0, 2, 0] EQL 0))
206 0776 3 THEN
207 U 0777 3 %ELSE
208 U 0778 3
209 U 0779 3 IF ((.NAME_LEN EQL 0) OR
210 U 0780 3 (.CHAR LSS %C'A') OR ((.CHAR GTR %C'Z') AND (.CHAR LSS %C'a')) OR (.CHAR GTR %C'z'))
211 U 0781 3 THEN
212 U 0782 3 %FI
213 0783 3
214 0784 4 BEGIN
215 0785 4 EDT$$MSG_BELL (EDT$_INVBUFNAM);
216 0786 4 RETURN;
217 0787 4 END;
218 0788 3
219 0789 3 EDT$$CNV UPC (.NAME_START, .NAME_LEN);
220 0790 3 SAVE_TBCB = .EDT$$A_CUR_BUF;
221 0791 3
222 0792 3 IF EDT$$FND_BUF (.NAME_START, .NAME_LEN)
223 0793 3 THEN
224 0794 3 EDT$$A_ALT_BUF = .EDT$$A_CUR_BUF
225 0795 3 ELSE
226 0796 3 EDT$$MSG_BELL (EDT$_INSMEM);
227 0797 3
228 0798 3 EDT$$A_CUR_BUF = .SAVE_TBCB;
229 0799 3 EDT$$RD_CURLN ();
230 0800 3 END
231 0801 2 ELSE
232 0802 2 EDT$$A_ALT_BUF = .EDT$$A_PST_BUF;
233 0803 2
234 0804 1 END;

```

! of routine EDT\$\$GET_BUF

.TITLE EDT\$CHMGBUF EDT\$CHMGBUF - look for a buffer nam
e

.IDENT \V04-000\

.EXTRN EDT\$\$MSG_BELL, EDT\$\$CNV UPC
.EXTRN EDT\$\$FND_BUF, EDT\$\$RD_CURLN
.EXTRN EDT\$\$A_CMD_END, EDT\$\$A_CMD_BUF
.EXTRN EDT\$\$A_ALT_BUF, EDT\$\$A_PST_BUF
.EXTRN EDT\$\$A_CUR_BUF, EDT\$\$B_CHAR_INFO
.EXTRN EDT\$_INVBUFNAM, EDT\$_INSMEM

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

.ENTRY EDT\$\$GET_BUF, 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
55 00000000G 00 9E 00025
54 D4 0002C
51 65 D0 0002E
50 61 9A 00031

MOVAB EDT\$\$A_ALT_BUF, R10
MOVAB EDT\$\$MSG_BELL, R9
MOVAB EDT\$\$B_CHAR_INFO, R8
MOVAB EDT\$\$A_CMD_END, R7
MOVAB EDT\$\$A_CUR_BUF, R6
MOVAB EDT\$\$A_CMD_BUF, R5
CLRL NAME_LEN
MOVL EDT\$\$A_CMD_BUF, R1
MOVZBL (R1), CHAR

0732
0733

	3D	50	D1	00034	CMPL	CHAR, #61	0735	
		7B	12	00037	BNEQ	8\$		
	67	51	D1	00039	CMPL	R1, EDT\$\$A_CMD_END		
		76	1E	0003C	BGEQU	8\$		
		65	D6	0003E	INCL	EDT\$\$A_CMD_BUF	0738	
	52	65	D0	00040	MOVL	EDT\$\$A_CMD_BUF, NAME_START	0739	
	51	65	D0	00043	1\$: MOVL	EDT\$\$A_CMD_BUF, R1	0741	
	67	51	D1	00046	CMPL	R1, EDT\$\$A_CMD_END		
		21	1E	00049	BGEQU	3\$		
	50	61	9A	0004B	MOVZBL	(R1), CHAR	0743	
51	50	58	C1	0004E	ADDL3	R8, CHAR, R1	0748	
	03	61	93	00052	BITB	(R1), #3		
		0F	12	00055	BNEQ	2\$		
	F0	8F	61	91	00057	CMPB	(R1), #240	0749
		09	13	0005B	BEQL	2\$		
	0000005F	8F	50	D1	0005D	CMPL	CHAR, #95	0750
		06	12	00064	BNEQ	3\$		
		54	D6	00066	2\$: INCL	NAME_LEN	0766	
		65	D6	00068	INCL	EDT\$\$A_CMD_BUF	0767	
		D7	11	0006A	BRB	1\$	0741	
	50	62	9A	0006C	3\$: MOVZBL	(NAME_START), CHAR	0770	
		54	D5	0006F	TSTL	NAME_LEN	0775	
		06	13	00071	BEQL	4\$		
	03	6840	93	00073	BITB	EDT\$\$B_CHAR_INFO[CHAR], #3		
		0A	12	00077	BNEQ	5\$		
	00000000G	8F	DD	00079	4\$: PUSHL	#EDT\$ INVBUFNAM	0785	
	69	01	FB	0007F	CALLS	#1, EDT\$\$MSG_BELL		
		04	00082	RET			0784	
		14	BB	00083	5\$: PUSHR	#^M<R2,R4>	0789	
	00000000G	00	02	FB	00085	CALLS	#2, EDT\$\$CNV_UPC	
		53	66	D0	0008C	MOVL	EDT\$\$A_CUR_BUF, SAVE_TBCB	0790
		14	BB	0008F	PUSHR	#^M<R2,R4>	0792	
	00000000G	00	02	FB	00091	CALLS	#2, EDT\$\$FND_BUF	
		05	50	E9	00098	BLBC	R0, 6\$	
		6A	66	D0	0009B	MOVL	EDT\$\$A_CUR_BUF, EDT\$\$A_ALT_BUF	0794
		09	11	0009E	BRB	7\$		
	00000000G	8F	DD	000A0	6\$: PUSHL	#EDT\$ INSMEM	0796	
	69	01	FB	000A6	CALLS	#1, EDT\$\$MSG_BELL		
		53	D0	000A9	7\$: MOVL	SAVE_TBCB, EDT\$\$A_CUR_BUF	0798	
	00000000G	00	00	FB	000AC	CALLS	#0, EDT\$\$RD_CURLN	0799
		04	000B3	RET			0735	
	6A	00000000G	00	D0	000B4	8\$: MOVL	EDT\$\$A_PST_BUF, EDT\$\$A_ALT_BUF	0802
		04	000BB	RET			0804	

: Routine Size: 188 bytes, Routine Base: _EDT\$CODE + 0000

: 235 0805 1
: 236 0806 1 !<BLF/PAGE>

EDT\$CHMGBUF
V04-000

EDT\$CHMGBUF - look for a buffer name
EDT\$\$GET_BUF - look for a buffer name

B 10
15-Sep-1984 23:57:05
14-Sep-1984 12:22:31

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]CHMGBUF.BLI;1
Page 7 (4)

: 279 0807 1 END
: 239 0808 1
: 240 0809 0 ELUDOM

! of module EDT\$CHMGBUF

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	188	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	26	6	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
_\$255\$DUA28:[EDT.SRC]SUPPORTS.L32;1	2	1	50	5	00:00.1

COMMAND QUALIFIERS

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

: Size: 188 code + 0 data bytes
: Run Time: 00:14.4
: Elapsed Time: 00:17.7
: Lines/CPU Min: 3370
: Lexemes/CPU-Min: 11166
: Memory Used: 95 pages
: Compilation Complete

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