

EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT

• • •

.....

.....

1

...

.....

SPECIAL


```
0001 0 %TITLE 'EDT$UGBUFFER - get a buffer pointer'
0002 0 MODULE EDT$UGBUFFER ( ! Get a buffer pointer
0003 0 IDENT = 'V04-000' ! File: UGBUFFER.BLI Edit: JBS1005
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 Get a buffer pointer. Create a buffer if necessary.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: February 7, 1978
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 19-FEB-1981. This module was created by
0045 1 extracting routine EDT$GET_BUFPTR from module UTIL.
0046 1 1-002 - Regularized headers. JBS 25-Feb-1981
0047 1 1-003 - Fix module name. JBS 11-Mar-1981
0048 1 1-004 - Implement virtual memory deallocation TMV 6-Aug-81
0049 1 1-005 - When a buffer is first allocated we are positioned at
0050 1 its front. JBS 08-Apr-1982
0051 1 --
0052 1
```


EDTSUGBUFFER
V04-000

EDTSUGBUFFER - get a buffer pointer
Declarations

J 1
16-Sep-1984 02:00:20
14-Sep-1984 12:25:15

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

```

54      0053 1 %SBTTL 'Declarations'
55      0054 1
56      0055 1 | TABLE OF CONTENTS:
57      0056 1 |
58      0057 1 |
59      0058 1 REQUIRE 'EDTSRC:TRAROUNAM';
60      0497 1
61      0498 1 FORWARD ROUTINE
62      0499 1     EDT$$GET_BUFPTR;
63      0500 1
64      0501 1 |
65      0502 1 | INCLUDE FILES:
66      0503 1 |
67      0504 1 |
68      0505 1 REQUIRE 'EDTSRC:EDTREQ';
69      0640 1
70      0641 1 |
71      0642 1 | MACROS:
72      0643 1 |
73      0644 1 |     NONE
74      0645 1 |
75      0646 1 | EQUATED SYMBOLS:
76      0647 1 |
77      0648 1 |     NONE
78      0649 1 |
79      0650 1 | OWN STORAGE:
80      0651 1 |
81      0652 1 |     NONE
82      0653 1 |
83      0654 1 | EXTERNAL REFERENCES:
84      0655 1 |
85      0656 1 |     In the routine
```



```

: 87 0657 1 %SBTTL 'EDT$$GET_BUFPTR - get a buffer pointer'
: 88 0658 1
: 89 0659 1 GLOBAL ROUTINE EDT$$GET_BUFPTR (
: 90 0660 1     NAME,
: 91 0661 1     LEN,
: 92 0662 1     BUF_ADDR,
: 93 0663 1     NEW
: 94 0664 1     ) =
: 95 0665 1
: 96 0666 1 ++
: 97 0667 1 FUNCTIONAL DESCRIPTION:
: 98 0668 1
: 99 0669 1     This routine searches for a buffer with a specified name. If the buffer
100 0670 1     if found, then the address of the TBCB is returned. If the buffer is not
101 0671 1     found, then a buffer by that name is created.
102 0672 1
103 0673 1 FORMAL PARAMETERS:
104 0674 1
105 0675 1     NAME           Pointer to a string containing the name of the buffer.
106 0676 1
107 0677 1     LEN           Length of the name string.
108 0678 1
109 0679 1     BUF_ADDR      Address of a fullword to receive the TBCB address.
110 0680 1
111 0681 1     NEW           Address of a flag to be set if the buffer was newly created.
112 0682 1
113 0683 1 IMPLICIT INPUTS:
114 0684 1
115 0685 1     EDT$$A_BUF_LST
116 0686 1
117 0687 1 IMPLICIT OUTPUTS:
118 0688 1
119 0689 1     EDT$$A_BUF_LST
120 0690 1
121 0691 1 ROUTINE VALUE:
122 0692 1
123 0693 1     Returns zero if an attempt to create a new buffer failed, one otherwise.
124 0694 1
125 0695 1 SIDE EFFECTS:
126 0696 1
127 0697 1     May create a new buffer and link it onto the buffer list.
128 0698 1
129 0699 1 --
130 0700 1
131 0701 2 BEGIN
132 0702 2
133 0703 2 EXTERNAL ROUTINE
134 0704 2     EDT$$ALO_HEAP;
135 0705 2
136 0706 2 EXTERNAL
137 0707 2     EDT$$A_BUF_LST : REF TBCB_BLOCK,
138 0708 2     EDT$$L_LNO_ZERO : LN_BLOCK;
139 0709 2
140 0710 2 LOCAL
141 0711 2     NEW_BUF : REF TBCB_BLOCK;
142 0712 2
143 0713 2 !+
```



```
144 0714 2 ! Start with the head of the text buffer list.
145 0715 2 !-
146 0716 2 NEW_BUF = .EDT$$A_BUF_LST;
147 0717 2 !+
148 0718 2 Now link our way through the list until we either find a matching
149 0719 2 name or hit the end of the list.
150 0720 2 !-
151 0721 2
152 0722 2 WHILE (.NEW_BUF NEQ 0) DO
153 0723 2
154 0724 2     IF CH$EQL (.LEN, .NAME, .NEW_BUF [TBCB_NAME_LEN], NEW_BUF [TBCB_NAME])
155 0725 2     THEN
156 0726 2         EXITLOOP
157 0727 2     ELSE
158 0728 2         NEW_BUF = .NEW_BUF [TBCB_NEXT_BUF];
159 0729 2
160 0730 2 IF (.NEW_BUF EQL 0)
161 0731 2 THEN
162 0732 2     BEGIN
163 0733 2 !+
164 0734 2 Well, we didn't find it. Better create a new one.
165 0735 2 !-
166 0736 2
167 0737 2     IF ( NOT EDT$$ALO_HEAP (%REF (TBCB_SIZE + .LEN), NEW_BUF)) THEN RETURN (0);
168 0738 2
169 0739 2     NEW_BUF [TBCB_NAME_LEN] = .LEN;
170 0740 2 !+
171 0741 2 Move the name into the newly created text buffer.
172 0742 2 !-
173 0743 2     EDT$$CPY_MEM (.LEN, .NAME, NEW_BUF [TBCB_NAME]);
174 0744 2 !+
175 0745 2 And append it to the front of the buffer list.
176 0746 2 !-
177 0747 2     NEW_BUF [TBCB_NEXT_BUF] = .EDT$$A_BUF_LST;
178 0748 2     NEW_BUF [TBCB_PREV_BUF] = 0;
179 0749 2
180 0750 2     IF (.EDT$$A_BUF_LST NEQ 0) THEN EDT$$A_BUF_LST [TBCB_PREV_BUF] = .NEW_BUF;
181 0751 2
182 0752 2     EDT$$A_BUF_LST = .BUF_ADDR = .NEW_BUF;
183 0753 2 !+
184 0754 2 Initialize the other fields in the TBCB.
185 0755 2 !-
186 0756 2     NEW_BUF [TBCB_LINE_ADDR] = 0;           ! Pointer to current line.
187 0757 2     NEW_BUF [TBCB_CUR_BUKT] = 0;           ! Current bucket number.
188 0758 2     MOVELINE (EDT$$L [NO_ZERO, NEW_BUF [TBCB_CUR_LIN]]); ! Current line number.
189 0759 2     NEW_BUF [TBCB_CHAR_POS] = 0;           ! The character position within the line
190 0760 2     NEW_BUF [TBCB_FIRST_BUKT] = 0;         ! First bucket number.
191 0761 2     NEW_BUF [TBCB_LAST_BUKT] = 0;          ! Last bucket number.
192 0762 2     MOVELINE (EDT$$L [LNO_ZERO, NEW_BUF [TBCB_INPUT_LINE]]); ! Number of last input line.
193 0763 2     MOVELINE (EDT$$L [LNO_ZERO, NEW_BUF [TBCB_LINE_COUNT]]); ! Count of lines in buffer.
194 0764 2     NEW_BUF [TBCB_CHAR_COUNT] = 0;         ! Count of chars in buffer.
195 0765 2     NEW_BUF [TBCB_INPUT_RAB] = 0;          ! Pointer to input RAB.
196 0766 2     NEW_BUF [TBCB_IS_MAC] = 0;
197 0767 2     .NEQ = 1;
198 0768 2     END
199 0769 2 ELSE
200 0770 2     .NEW = 0;
```

EDTSUGBUFFER
V04-000

EDTSUGBUFFER - get a buffer pointer
EDT\$\$GET_BUFPTR - get a buffer pointer

M 1
16-Sep-1984 02:00:20
14-Sep-1984 12:25:15

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]UGBUFFER.BLI;1 Page 5 (3)

: 201
: 202
: 203
: 204
0771 2
0772 2
0773 2
0774 1
.BUF_ADDR = .NEW_BUF;
RETURN (1);
END;

! of routine EDT\$\$GET_BUFPTR

.TITLE EDTSUGBUFFER EDTSUGBUFFER - get a buffer pointer

.IDENT \V04-000\

.EXTRN EDT\$\$ALO_HEAP, EDT\$\$A_BUF_LST

.EXTRN EDT\$\$L_LNO_ZERO

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

.ENTRY EDT\$\$GET_BUFPTR, Save R2,R3,R4,R5,R6,R7,R8

MOVAB EDT\$\$A_BUF_LST, R8 : 0659
MOVAB EDT\$\$L_LNO_ZERO, R7
SUBL2 #8, SP
MOVL EDT\$\$A_BUF_LST, NEW_BUF : 0716
MOVL NEW_BUF, R4 : 0722
BEQL 2\$:
MOVZBL 44(R4), R0 : 0724
CMPC5 LEN, @NAME, #0, R0, 45(R4)
BEQL 2\$
MOVL 38(R4), NEW_BUF : 0728
BRB 1\$: 0724
TSTL NEW_BUF : 0730
BNEQ 4\$:
PUSHAB NEW_BUF : 0737
ADDL3 #45, LEN, 4(SP)
PUSHAB 4(SP)
CALLS #2, EDT\$\$ALO_HEAP
BLBC R0, 6\$:
MOVL NEW_BUF, R6 : 0739
MOVB LEN, 44(R6)
MOVC3 LEN, @NAME, 45(R6) : 0743
MOVL EDT\$\$A_BUF_LST, R0 : 0747
MOVL R0, 38(R6)
CLRL 34(R6) : 0748
TSTL R0 : 0750
BEQL 3\$:
MOVL R6, 34(R0)
MOVL R6, @BUF_ADDR : 0752
MOVL R6, EDT\$\$A_BUF_LST :
CLRL (R6) : 0756
CLRW 4(R6) : 0757
MOVC3 #6, EDT\$\$L_LNO_ZERO, 6(R6) : 0758
CLRL 12(R6) : 0759
CLRW 16(R6) : 0761
MOVC3 #6, EDT\$\$L_LNO_ZERO, 18(R6) : 0762
MOVC3 #6, EDT\$\$L_LNO_ZERO, 24(R6) : 0763
CLRL 30(R6) : 0764
CLRW 42(R6) : 0765
MOVL #1, @NEW : 0767
BRB 5\$: 0730

50 00 04 58 00000000G 00 01FC 00000
57 00000000G 00 9E 00002
5E 08 C2 00010
04 AE 68 D0 00013
54 04 AE D0 00017 1\$:
16 13 0001B
50 2C A4 9A 0001D
04 BC 08 AC 2D 00021
2D A4 00028
07 13 0002A
04 AE 26 A4 D0 0002C
E4 11 00031
04 AE D5 00033 2\$:
65 12 00036
04 AE 9F 00038
2D C1 0003B
04 AE 9F 00041
00 02 FB 00044
58 50 E9 0004B
56 04 AE D0 0004E
2C A6 08 AC 90 00052
2D A6 04 BC 08 AC 28 00057
50 68 D0 0005E
26 A6 50 D0 00061
22 A6 D4 00065
50 D5 00068
04 13 0006A
22 A0 56 D0 0006C
0C BC 56 D0 00070 3\$:
68 56 D0 00074
66 D4 00077
04 A6 B4 00079
06 28 0007C
0C A6 D4 00081
10 A6 B4 00084
06 28 00087
12 A6 06 28 0008C
18 A6 06 28 00091
1E A6 D4 00094
2A A6 B4 00097
10 BC 01 D0 00097
03 11 0009B

EDTSUGBUFFER
V04-000

EDTSUGBUFFER - get a buffer pointer
EDT\$GET_BUFPTR - get a buffer pointer

N 1
16-Sep-1984 02:00:20
14-Sep-1984 12:25:15

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]UGBUFFER.BLI;1 Page 6 (3)

0C	BC	10	BC	D4	0009D	4\$:	CLRL	@NEW	:	0770
	50	04	AE	D0	000A0	5\$:	MOVL	NEW_BUF, @BUF_ADDR	:	0772
			01	D0	000A5		MOVL	#1, -R0	:	0773
				04	000A8		RET		:	
			50	D4	000A9	6\$:	CLRL	R0	:	0774
				04	000AB		RET		:	

; Routine Size: 172 bytes, Routine Base: _EDT\$CODE + 0000

; 205 0775 1
; 206 0776 1 !<BLF/PAGE>

EDT\$UGBUFFER
V04-000

EDT\$UGBUFFER - get a buffer pointer
EDT\$\$GET_BUFPTR - get a buffer pointer

B 2
16-Sep-1984 02:00:20
14-Sep-1984 12:25:15

VAX-11 Bliss-32 V4.0-742
DISK\$VMMASTER:[EDT.SRC]UGBUFFER.BLI;1 (4)

Page 7

: 208
: 209
: 210
0777 1 END
0778 1
0779 0 ELUDOM

! of module EDT\$UGBUFFER

PSECT SUMMARY

:
: Name Bytes Attributes
: _EDT\$CODE 172 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 30 7 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\$:UGBUFFER/OBJ=OBJ\$:UGBUFFER MSRC\$:UGBUFFER.BLI/UPDATE=(ENH\$:UGBU
: FFER)

: Size: 172 code + 0 data bytes
: Run Time: 00:14.4
: Elapsed Time: 00:17.1
: Lines/CPU Min: 3245
: Lexemes/CPU-Min: 11595
: Memory Used: 102 pages
: Compilation Complete

0141 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

