


```

UU      UU  FFFFFFFF  BBBB BBBB  UU      UU  FFFFFFFF  FFFFFFFF  EEEEEEE  RRRRRRR
UU      UU  FFFFFFFF  BBBB BBBB  UU      UU  FFFFFFFF  FFFFFFFF  EEEEEEE  RRRRRRR
UU      UU  FF          BB      BB  UU      UU  FF          FF          FF          RR          RR
UU      UU  FF          BB      BB  UU      UU  FF          FF          FF          RR          RR
UU      UU  FF          BB      BB  UU      UU  FF          FF          FF          RR          RR
UU      UU  FF          BB      BB  UU      UU  FF          FF          FF          RR          RR
UU      UU  FFFFFFFF  BBBB BBBB  UU      UU  FFFFFFFF  FFFFFFFF  EEEEEEE  RRRRRRR
UU      UU  FFFFFFFF  BBBB BBBB  UU      UU  FFFFFFFF  FFFFFFFF  EEEEEEE  RRRRRRR
UU      UU  FF          BB      BB  UU      UU  FF          FF          FF          RR          RR
UU      UU  FF          BB      BB  UU      UU  FF          FF          FF          RR          RR
UU      UU  FF          BB      BB  UU      UU  FF          FF          FF          RR          RR
UU      UU  FF          BB      BB  UU      UU  FF          FF          FF          RR          RR
UUUUUUUU  FF          BBBB BBBB  UUUUUUUU  FF          FF          EEEEEEE  RR          RR
UUUUUUUU  FF          BBBB BBBB  UUUUUUUU  FF          FF          EEEEEEE  RR          RR

```

```

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
LLLLLLLL  IIIIII  SSSSSSS
LLLLLLLL  IIIIII  SSSSSSS

```

```

1 0001 0 %TITLE 'EDT$UFBUFFER - find a buffer'
2 0002 0 MODULE EDT$UFBUFFER ( ' Find a buffer
3 0003 0 IDENT = 'V04-000' ! File: UFBUFFER.BLI Edit: JBS1005
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 Find a buffer, given its name. A buffer is created if necessary.
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: February 7, 1978
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. DJS 19-FEB-1981. This module was created by
45 0045 1 extracting routine EDT$$FND_BUF from module UTIL.
46 0046 1 1-002 - Regularized headers. JBS 25-Feb-1981
47 0047 1 1-003 - Fix module name. JBS 11-Mar-1981
48 0048 1 1-004 - Initialize more of the tbc. STS 10-Mar-1982
49 0049 1 1-005 - Improve the appearance of the listing. JBS 17-Jun-1983
50 0050 1 --
51 0051 1

```

EDT\$UFBUFFER
V04-000

EDT\$UFBUFFER - find a buffer
Declarations

G 16
16-Sep-1984 01:59:18
14-Sep-1984 12:25:13

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

```

: 53 0052 1 %SBTTL 'Declarations'
: 54 0053 1 |
: 55 0054 1 | TABLE OF CONTENTS:
: 56 0055 1 |
: 57 0056 1 |
: 58 0057 1 REQUIRE 'EDT$SRC:TRAROUNAM';
: 59 0496 1 |
: 60 0497 1 FORWARD ROUTINE
: 61 0498 1 EDT$SFND_BUF;
: 62 0499 1 |
: 63 0500 1 |
: 64 0501 1 | INCLUDE FILES:
: 65 0502 1 |
: 66 0503 1 |
: 67 0504 1 REQUIRE 'EDT$SRC:EDTREQ';
: 68 0639 1 |
: 69 0640 1 |
: 70 0641 1 | MACROS:
: 71 0642 1 |
: 72 0643 1 | NONE
: 73 0644 1 |
: 74 0645 1 | EQUATED SYMBOLS:
: 75 0646 1 |
: 76 0647 1 | NONE
: 77 0648 1 |
: 78 0649 1 | OWN STORAGE:
: 79 0650 1 |
: 80 0651 1 | NONE
: 81 0652 1 |
: 82 0653 1 | EXTERNAL REFERENCES:
: 83 0654 1 |
: 84 0655 1 | In the routine
```

```
86 0656 1 %SBTTL 'EDT$$FND_BUF - find a buffer'
87 0657 1
88 0658 1 GLOBAL ROUTINE EDT$$FND_BUF (           : Find a buffer
89 0659 1     NAME,                               : pointer to the buffer name
90 0660 1     LEN                                 : length of that name
91 0661 1     ) =
92 0662 1
93 0663 1
94 0664 1 :++
95 0665 1 :
96 0666 1 :     This routine searches for a buffer with a specified name and positions
97 0667 1 :     into it.  If a buffer by that name does not exist, then it is created.
98 0668 1 :     The global variable EDT$$A_PRV_BUF is set to point to the TBCB
99 0669 1 :     of the buffer we were originally positioned in.
100 0670 1 :
101 0671 1 FORMAL PARAMETERS:
102 0672 1
103 0673 1     NAME                               Pointer to a string containing the name of the buffer.
104 0674 1
105 0675 1     LEN                               Length of the name string.
106 0676 1
107 0677 1 IMPLICIT INPUTS:
108 0678 1
109 0679 1     EDT$$A_CUR_BUF
110 0680 1
111 0681 1 IMPLICIT OUTPUTS:
112 0682 1
113 0683 1     EDT$$A_CUR_BUF
114 0684 1     EDT$$A_PRV_BUF
115 0685 1
116 0686 1 ROUTINE VALUE
117 0687 1
118 0688 1     Returns 1 if the buffer was found or created, 0 if an attempt to create a
119 0689 1     new buffer failed.
120 0690 1
121 0691 1 SIDE EFFECTS:
122 0692 1
123 0693 1     May create a new buffer.
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$$GET_BUF_PTR,
131 0701 2     EDT$$NEW_BUF;
132 0702 2
133 0703 2 EXTERNAL
134 0704 2     EDT$$L_LNO_ZERO : LN_BLOCK,
135 0705 2     EDT$$A_CUR_BUF : REF TBCB_BLOCK,      ! Pointer to current buffer TBCB
136 0706 2     EDT$$A_PRV_BUF : REF TBCB_BLOCK;    ! Pointer to previous buffer TBCB
137 0707 2
138 0708 2 LOCAL
139 0709 2     BUF_ADDR : REF TBCB_BLOCK,
140 0710 2     NEW;
141 0711 2
142 0712 2 !+
```

```

: 143 0713 2 ! Look for the buffer.
: 144 0714 2 !-
: 145 0715 2
: 146 0716 2     IF EDT$$GET_BUF_PTR (.NAME, .LEN, BUF_ADDR, NEW)
: 147 0717 2     THEN
: 148 0718 2         BEGIN
: 149 0719 2
: 150 0720 2     !- Position to the new buffer.
: 151 0721 2
: 152 0722 2         EDT$$A_CUR_BUF = .EDT$$A_CUR_BUF;
: 153 0723 2         EDT$$A_CUR_BUF = .BUF_ADDR;
: 154 0724 2
: 155 0725 2     !- If the buffer was new, then open it up.
: 156 0726 2
: 157 0727 2
: 158 0728 2     IF (.NEW NEQ 0)
: 159 0729 2     THEN
: 160 0730 2         BEGIN
: 161 0731 2             BUF_ADDR [TCB_INPUT_RAB] = 0;
: 162 0732 2             MOVE LINE (EDT$$LNO_ZERO, BUF_ADDR [TCB_LINE_COUNT]);
: 163 0733 2             EDT$$NEW_BUF ();
: 164 0734 2         END;
: 165 0735 2
: 166 0736 2     RETURN (1);
: 167 0737 2     END
: 168 0738 2 ELSE
: 169 0739 2     RETURN (0);
: 170 0740 2
: 171 0741 1     END;

```

! of routine EDTSSFND_BUF

.TITLE EDTSUFBUFFER EDTSUFBUFFER - find a buffer
.IDENT \V04-000\

.EXTRN EDT\$\$GET_BUF_PTR
.EXTRN EDT\$\$NEW_BUF, EDT\$\$LNO_ZERO
.EXTRN EDT\$\$A_CUR_BUF, EDT\$\$A_PRV_BUF

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

.ENTRY EDTSSFND_BUF, Save R2,R3,R4,R5,R6

56	00000000G	00	9E	00002	MOVAB	EDT\$\$A_CUR_BUF, R6	:	0658
5E		08	C2	00009	SUBL2	#8, SP	:	
		5E	DD	0000C	PUSHL	SP	:	0716
		08	AE	9F	0000E	PUSHAB	BUF_ADDR	:
	7E	04	AC	7D	00011	MOVQ	NAME, -(SP)	:
	00000000G	00	04	FB	00015	CALLS	#4, EDT\$\$GET_BUF_PTR	:
		29	50	E9	0001C	BLBC	R0, 2\$:
	00000000G	00	66	D0	0001F	MOVL	EDT\$\$A_CUR_BUF, EDT\$\$A_PRV_BUF	:
		50	04	AE	D0	00026	MOVL	BUF_ADDR, R0
		66	50	D0	0002A	MOVL	R0, EDT\$\$A_CUR_BUF	:
			6E	D5	0002D	TSTL	NEW	:
			13	13	0002F	BEQL	1\$:
			2A	A0	94	00031	CLRB	42(R0)
18	A0	00000000G	00	06	28	00034	MOVCS	#6, EDT\$\$LNO_ZERO, 24(R0)
		00000000G	00	00	FB	0003D	CALLS	#0, EDT\$\$NEW_BUF
			50	01	D0	00044	MOVL	#1, R0

EDTSUFBUFFER
V04-000

EDTSUFBUFFER - find a buffer
EDTSSFND_BUF - find a buffer

J 16
16-Sep-1984 01:59:18
14-Sep-1984 12:25:13

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

50 04 00047 RET
04 00048 2s: CLRL R0
04 0004A RET

:
:
: 0741

: Routine Size: 75 bytes, Routine Base: _EDT\$CODE + 0000

: 172 0742 1
: 173 0743 1 !<BLF/PAGE>

EDTSUFBUFFER
V04-000

EDTSUFBUFFER - find a buffer
EDTSSFND_BUF - find a buffer

K 16
16-Sep-1984 01:59:18
14-Sep-1984 12:25:13

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

Page 6

: 175 0744 1 END
: 176 0745 1
: 177 0746 0 ELUDOM

! of module EDTSUFBUFFER

PSECT SUMMARY

Name Bytes Attributes
_EDT\$CODE 75 MOVEC,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\$:UFBUFFER/OBJ=OBJ\$:UFBUFFER MSRC\$:UFBUFFER.BLI/UPDATE=(ENH\$:UFBU
FFER)

: Size: 75 code + 0 data bytes
: Run Time: 00:10.6
: Elapsed Time: 00:14.4
: Lines/CPU Min: 4206
: Lexemes/CPU-Min: 11808
: Memory Used: 70 pages
: Compilation Complete

