

EEEEEEEEE	DDDDDDDDDD	TTTTTTTTTT
EEEEEEEEE	DDDDDDDDDD	TTTTTTTTTT
EEEEEEEEE	DDDDDDDDDD	TTTTTTTTTT
EEE	DDD	TTT
EEEEEEEEE	DDDDDDDDDD	TTT
EEEEEEEEE	DDDDDDDDDD	TTT
EEEEEEEEE	DDDDDDDDDD	TTT

FILEID**MACCAL

E 5

MM MM AAAAAA CCCCCCCC CCCCCCCC AAAAAA LL
MM MM AAAAAA CCCCCCCC CCCCCCCC AAAAAA LL
MM MM AA AA CC CC AA AA LL
MM MM AA AA CC CC AA AA LL
MM MM AA AA CC CC AA AA LL
MM MM AA AA CC CC AA AA LL
MM MM AA AA CC CC AA AA LL
MM MM AAAAAAAAAA CC CC AAAAAAAAAA LL
MM MM AAAAAAAAAA CC CC AAAAAAAAAA LL
MM MM AA AA CC CC AA AA LL
MM MM AA AA CC CC AA AA LL
MM MM AA AA CCCCCCCC CCCCCCCC AA AA LLLLLLLL
MM MM AA AA CCCCCCCC CCCCCCCC AA AA LLLLLLLL

LL IIIIII SSSSSSS
LL IIIIII SSSSSSS
LL II SS
LLIIIIII SSSSSSS
LLIIIIII SSSSSSS

ED
VO
:
:
:

```
1 0001 0 %TITLE 'EDT$MACCAL - macro call'  
2 0002 0 MODULE EDT$MACCAL (  
3 0003 0 IDENT = 'V04-000'  
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 Macro call  
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: 6-AUG-1979  
41 0041 1  
42 0042 1 MODIFIED BY:  
43 0043 1  
44 0044 1 1-001 - Original. DJS 18-FEB-1981. This module was created by  
45 0045 1 extracting routine EDT$SMAC_CALL from module EDTCTR.  
46 0046 1 1-002 - Regularize headers and fix file and module name. JBS 04-Mar-1981  
47 0047 1 1-003 - Remove L LINE. JBS 01-Oct-1981  
48 0048 1 1-004 - Replace [ LINE. TMV 7-Dec-81  
49 0049 1 1-005 - Add an entry point so that EDT$SEXECMD_NOOVERLAY can  
50 0050 1 get this module back into memory after a macro has been executed. JBS 10-Mar-1982  
51 0051 1 1-006 - Save and restore the command line over a macro call, so another  
52 0052 1 command can be after the macro name. JBS 03-Jun-1982  
53 0053 1 1-007 - See if a control C was found and reset command buffer. STS 16-Jul-1982  
54 0054 1 1-008 - Improve the appearance of the listing. JBS 14-Jun-1983  
55 0055 1 --  
56 0056 1
```

```
58      0057 1 %SBTTL 'Declarations'  
59      0058 1  
60      0059 1 TABLE OF CONTENTS:  
61      0060 1  
62      0061 1  
63      0062 1 REQUIRE 'EDTSRC:TRAROUNAM';  
64      0501 1  
65      0502 1 FORWARD ROUTINE  
66      0503 1     EDTSSMAC CALL : NOVALUE,  
67      0504 1     EDTSSLOAD_MACCAL : NOVALUE;  
68      0505 1  
69      0506 1  
70      0507 1 INCLUDE FILES:  
71      0508 1  
72      0509 1  
73      0510 1 REQUIRE 'EDTSRC:EDTREQ';  
74      0645 1  
75      0646 1  
76      0647 1 MACROS:  
77      0648 1     NONE  
78      0649 1  
79      0650 1  
80      0651 1 EQUATED SYMBOLS:  
81      0652 1  
82      0653 1     NONE  
83      0654 1  
84      0655 1 OWN STORAGE:  
85      0656 1     NONE  
86      0657 1  
87      0658 1  
88      0659 1 EXTERNAL REFERENCES:  
89      0660 1  
90      0661 1     In the routine
```

```

92      0662 1 %SBTTL 'EDT$SMAC_CALL - macro call'
93      0663 1
94      0664 1 GLOBAL ROUTINE EDT$SMAC_CALL (
95          MAC
96          ) : NOVALUE =
97      0667 1
98      0668 1 ++
99      0669 1     FUNCTIONAL DESCRIPTION:
100     0670 1
101     0671 1     Call a macro. A macro is a buffer which contains line-mode
102     0672 1     commands.
103     0673 1
104     0674 1     FORMAL PARAMETERS:
105     0675 1
106     0676 1     MAC           Address of the macro
107     0677 1
108     0678 1     IMPLICIT INPUTS:
109     0679 1
110     0680 1     EDTSSA_CUR_BUF
111     0681 1     EDTSSA_MAC_BUF
112     0682 1     EDTSSA_CMD_BUF
113     0683 1     EDTSSG_CMD_LEN
114     0684 1     EDTSST_CMD_BUF
115     0685 1     EDTSSA_CMD_END
116     0686 1
117     0687 1     IMPLICIT OUTPUTS:
118     0688 1
119     0689 1     EDTSSA_CUR_BUF
120     0690 1     EDTSSA_MAC_BUF
121     0691 1     EDTSSA_CMD_BUF
122     0692 1     EDTSSA_CMD_END
123     0693 1     EDTSSG_CC_DONE
124     0694 1
125     0695 1     ROUTINE VALUE:
126     0696 1
127     0697 1     NONE
128     0698 1
129     0699 1     SIDE EFFECTS:
130     0700 1
131     0701 1     Saves and restores the command line.
132     0702 1
133     0703 1     --
134     0704 1
135     0705 2     BEGIN
136     0706 2
137     0707 2     EXTERNAL ROUTINE
138     0708 2     EDT$SEXECMD_NOOVERLAY,
139     0709 2     EDT$SRD_CURLN : NOVALUE,
140     0710 2     EDT$STOP_BUF : NOVALUE,
141     0711 2     EDT$SALO_HEAP,
142     0712 2     EDT$SDEA_HEAP : NOVALUE,
143     0713 2     EDT$SFMT_MSG : NOVALUE;
144     0714 2
145     0715 2     EXTERNAL
146     0716 2     EDTSSG_CC_DONE,
147     0717 2     EDTSSA_CUR_BUF,
148     0718 2     EDTSSG_CMD_LEN,
```

! Same as EDT\$SEXECMD but no overlay analysis

! Allocate heap storage

! Deallocate heap storage

! Format a message

! Current tcb

```
149      0719 2      EDT$ST_CMD_BUF,  
150      0720 2      EDT$SA_MAC_BUF,  
151      0721 2      EDT$SA_CMD_BUF,  
152      0722 2      EDT$SA_CMD_END:  
153      0723 2  
154      0724 2      LOCAL  
155      0725 2      STATUS,  
156      0726 2      SAVE_TBCB,  
157      0727 2      SAVE_MACRO,  
158      0728 2      SAVE_CMD_BUF,  
159      0729 2      SAVE_CMD_END,  
160      0730 2      CMD_TEXT: REF VECTOR [, BYTE],  
161      0731 2      CMD_LENGTH;  
162      0732 2  
163      0733 2      MESSAGES ((INSMEM));  
164      0734 2      + Save the current command line.  
165      0735 2      -  
166      0736 2      SAVE_CMD_BUF = .EDT$SA_CMD_BUF;  
167      0737 2      SAVE_CMD_END = .EDT$SA_CMD_END;  
168      0738 2      CMD_LENGTH = CH$DIFF (CH$P[US (.SAVE_CMD_END, 1), .SAVE_CMD_BUF);  
169      0739 2  
170      0740 2  
171      0741 3      IF (.CMD_LENGTH GTR 0)  
172      0742 2      THEN  
173      0743 3      BEGIN  
174      0744 3  
175      0745 3      IF EDT$ALO_HEAP (CMD_LENGTH, CMD_TEXT)  
176      0746 3      THEN  
177      0747 3      CH$MOVE (.CMD_LENGTH, .SAVE_CMD_BUF, .CMD_TEXT)  
178      0748 3      ELSE  
179      0749 4      BEGIN  
180      0750 4      EDT$SFMT_MSG (EDT$INSMEM);  
181      0751 4      RETURN;  
182      0752 3      END;  
183      0753 3  
184      0754 2      END;  
185      0755 2  
186      0756 2      + Point the command processor to the macro without destroying  
187      0757 2      the current buffer or the current macro.  
188      0758 2  
189      0759 2      -  
190      0760 2      SAVE_TBCB = .EDT$SA_CUR_BUF;  
191      0761 2      EDT$SA_CUR_BUF = .MAC;  
192      0762 2      EDT$STOP_BUF ();  
193      0763 2      EDT$SA_CUR_BUF = .SAVE_TBCB;  
194      0764 2      EDT$SRD_CURLN ();  
195      0765 2      SAVE_MACRO = .EDT$SA_MAC_BUF;  
196      0766 2      EDT$SA_MAC_BUF = .MAC;  
197      0767 2      + Execute the commands in the specified buffer.  
198      0768 2  
199      0769 2      -  
200      0770 2      STATUS = EDT$SEXEC_CMD_NOOVERLAY (INP_MACRO);  
201      0771 2      +  
202      0772 2      - Restore the former macro.  
203      0773 2  
204      0774 2      EDT$SA_MAC_BUF = .SAVE_MACRO;  
205      0775 2      +
```

```

206      0776 2 ! Restore the former command line contents, if any.
207      0777 2
208      0778 2 EDTSSA_CMD_BUF = .SAVE_CMD_BUF;
209      0779 2 EDTSSA_CMD_END = .SAVE_CMD_END;
210      0780 2
211      0781 2 IF (.CMD_LENGTH GTR 0)
212      0782 2 THEN
213      0783 2 BEGIN
214      0784 2 CHSMOVE (.CMD_LENGTH, .CMD_TEXT, .SAVE_CMD_BUF);
215      0785 2 EDT$DEA_HEAP(.CMD_LENGTH, .CMD_TEXT);
216      0786 2 END;
217      0787 2
218      0788 2 IF (.STATUS EQL 2)           ! if we saw a control C
219      0789 2 THEN
220      0790 2 BEGIN
221      0791 2
222      0792 2 IF (.EDTSSA_CMD_END NEQ .EDTSSA_CMD_BUF) THEN EDT$G_CC_DONE = 1;
223      0793 2
224      0794 3 EDTSSA_CMD_BUF = EDT$ST_CMD_BUF;
225      0795 3 EDTSSA_CMD_END = .EDTSSA_CMD_BUF;
226      0796 3 CHSWCHAR (%C'!', .EDTSSA_CMD_END);
227      0797 2 END;
228      0798 2
229      0799 1 END;                      ! of routine EDT$MAC_CALL

```

.TITLE EDT\$MACCAL EDT\$MACCAL - macro call
.IDENT \V04-000\

.EXTRN EDT\$SEXECMD_NOOVERLAY
.EXTRN EDT\$RD_CURLN, EDT\$STOP_BUF
.EXTRN EDT\$ALO_HEAP, EDT\$DEA_HEAP
.EXTRN EDT\$SFMT_MSG, EDT\$G_CC_DONE
.EXTRN EDTSSA_CUR_BUF, EDT\$G_CMD_LEN
.EXTRN EDT\$ST_CMD_BUF, EDTSSA_MAC_BUF
.EXTRN EDTSSA_CMD_BUF, EDTSSA_CMD_END
.EXTRN EDT\$INSMEM

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

		OFFC 00000			
		5B 0000000G	00 9E 00002	MOVAB	EDTSSA_CUR_BUF, R11
		5A 0000000G	00 9E 00009	MOVAB	EDTSSA_CMD_BUF, R10
		59 0000000G	00 9E 00010	MOVAB	EDTSSA_CMD_END, R9
		5E	08 C2 00017	SUBL2	#8, SP
		57	6A D0 0001A	MOVL	EDTSSA_CMD_BUF, SAVE_CMD_BUF
		56	69 D0 0001D	MOVL	EDTSSA_CMD_END, SAVE_CMD_END
50	04 AE	56	57 C3 00020	SUBL3	SAVE_CMD_BUF, SAVE_CMD_END, R0
		01	A0 9E 00024	MOVAB	1(R0), CMD_LENGTH
			25 15 00029	BLEQ	2\$
			5E DD 0002B	PUSHL	SP
		08	AE 9F 0002D	PUSHAB	CMD_LENGTH
	0000000G	00	02 FB 00030	CALLS	#2, EDT\$ALO_HEAP
		08	50 E9 00037	BLBC	R0, 1\$
00 BE	67	04	AE 28 0003A	MOV3	CMD_LENGTH, (SAVE_CMD_BUF), @CMD_TEXT
		OE	11 00040	BRB	2\$

EDT\$MACCAL
V04-000EDT\$MACCAL - macro call
EDT\$MAC_CALL - macro callK 5
16-Sep-1984 01:07:09
14-Sep-1984 12:23:50
VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]MACCAL.BLI;1Page 6
(3)ED1
VO4

00000000G	00	00000000G	8F	DD	00042	1\$:	PUSHL	#EDTS_INSMEM	: 0750
			01	FB	00048		CALLS	#1, EDT\$FMT_MSG	
				04	0004F		RET		0749
	52		6B	DO	00050	2\$:	MOVL	EDTSSA_CUR_BUF, SAVE_TBCB	: 0760
00000000G	00		04	AC	00053		MOVL	MAC, EDTSSA_CUR_BUF	0761
	6B		00	FB	00057		CALLS	#0, EDT\$STOP_BUF	0762
00000000G	00		52	DO	0005E		MOVL	SAVE_TBCB, EDTSSA_CUR_BUF	0763
00000000G	00		00	FB	00061		CALLS	#0, EDT\$RD_CURLN	0764
	52	00000000G	00	DO	00068		MOVL	EDTSSA_MAC_BUF, SAVE_MACRO	0765
00000000G	00		04	AC	0006F		MOVL	MAC, EDTSSA_MAC_BUF	0766
			01	DD	00077		PUSHL	#1	0770
00000000G	00		01	FB	00079		CALLS	#1, EDT\$SEXECMD_NOOVERLAY	
	58		50	DO	00080		MOVL	RO, STATUS	
00000000G	00		52	DO	00083		MOVL	SAVE_MACRO, EDTSSA_MAC_BUF	0774
	6A		57	DO	0008A		MOVL	SAVE_CMD_BUF, EDTSSA_CMD_BUF	0778
	69		56	DO	0008D		MOVL	SAVE_CMD-END, EDTSSA_CMD-END	0779
			04	AE	D5	00090	TSTL	CMD_LENGTH	0781
			12	15	00093		BLEQ	3\$	
67	00	BE	04	AE	28	00095	MOVC3	CMD_LENGTH, @CMD_TEXT, (SAVE_CMD_BUF)	0784
			5E	DD	0009B		PUSHL	SP	0785
00000000G	00		08	AE	9F	0009D	PUSHAB	CMD_LENGTH	
	02		02	FB	000A0		CALLS	#2, EDT\$DEA_HEAP	
			58	D1	000A7	3\$:	CMPL	STATUS, #2	0788
			1C	12	000AA		BNEQ	5\$	
	6A		69	D1	000AC		CMPL	EDTSSA_CMD-END, EDTSSA_CMD_BUF	0792
00000000G	00		07	13	000AF		BEQL	4\$	
	6A	00000000G	01	DO	000B1		MOVL	#1, EDT\$G_CC_DONE	
			6A	9E	000B8	4\$:	MOVAB	EDT\$ST_CMD_BUF, EDTSSA_CMD_BUF	0794
			69	6A	DO	000BF	MOVL	EDTSSA_CMD_BUF, EDTSSA_CMD-END	0795
			50	69	DO	000C2	MOVL	EDTSSA_CMD-END, RO	0796
			60	21	90	000C5	MOVB	#33, (R0)	
			04	000C8		5\$:	RET		0799

: Routine Size: 201 bytes, Routine Base: _EDT\$CODE + 0000

: 230 0800 1

```

232      0801 1 %SBTTL 'EDT$LOAD_MACCAL - load this module into memory'
233      0802 1
234      0803 1 GLOBAL ROUTINE EDT$LOAD_MACCAL           ! Load this module into memory
235      0804 1 : NOVALUE =
236      0805 1
237      0806 1 ++
238      0807 1 FUNCTIONAL DESCRIPTION:
239      0808 1
240      0809 1 This routine has no function. It exists as an entry point so that
241      0810 1 EDT$EXE_CMD_NOOVERLAY can call this module back into memory before
242      0811 1 returning to it.
243      0812 1
244      0813 1 FORMAL PARAMETERS:
245      0814 1     NONE
246      0815 1
247      0816 1 IMPLICIT INPUTS:
248      0817 1     NONE
249      0818 1
250      0819 1 IMPLICIT OUTPUTS:
251      0820 1     NONE
252      0821 1
253      0822 1     NONE
254      0823 1
255      0824 1     NONE
256      0825 1     ROUTINE VALUE:
257      0826 1     NONE
258      0827 1
259      0828 1     NONE
260      0829 1     SIDE EFFECTS:
261      0830 1     NONE
262      0831 1
263      0832 1
264      0833 1     --
265      0834 1
266      0835 2     BEGIN
267      0836 2     0
268      0837 1     END;                                ! of routine EDT$LOAD_MACCAL

```

0000 00000
04 00002

.ENTRY EDT\$LOAD_MACCAL, Save nothing
RET

: 0803
: 0837

; Routine Size: 3 bytes, Routine Base: _EDT\$CODE + 00C9

; 269 0838 1
; 270 0839 1 !<BLF/PAGE>

EDT\$MACCAL
V04-000

EDT\$MACCAL - macro call
EDT\$LOAD_MACCAL - load this module into memory

M 5

16-Sep-1984 01:07:09
14-Sep-1984 12:23:50

VAX-11 Bliss-32 v4.0-742
DISK\$VMSMASTER:[EDT.SRC]MACCAL.BLI;1

Page 8
(5)

: 272 0840 1 END
: 273 0841 1
: 274 0842 0 ELUDOM

: of module EDT\$MACCAL

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	204	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

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

COMMAND QUALIFIERS

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

Size: 204 code + 0 data bytes
Run Time: 00:14.5
Elapsed Time: 00:18.1
Lines/CPU Min: 3496
Lexemes/CPU-Min: 9707
Memory Used: 91 pages
Compilation Complete

0137 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

