

[illegible]

EXE

Mod

ED 1

ED

ED

ED

ED

ED1

ED

ED

ED
EDED
EDSYS
LIB

L18

```

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
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```



```
0001 0 %TITLE 'EDT$KEYCOM - get a keypad command'
0002 0 MODULE EDT$KEYCOM (
0003 0 IDENT = 'V04-000'
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 keypad command.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: April 7, 1979
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 24-Feb-1981. This module was created by
0045 1 extracting routine EDT$$GET_KPADCMD from module KEYTRAN.
0046 1 1-002 - Regularize headers. JBS 09-Mar-1981
0047 1 1-003 - Add return values. JBS 02-Oct-1981
0048 1 1-004 - Check one more return value, and add a variable to call to
0049 1 EDT$$KPAD_INP. STS 21-Oct-1981
0050 1 1-005 - Set a flag if control C actually aborts something. JBS 25-May-1982
0051 1 1-006 - In NOKEYPAD mode copy control characters into the buffer. JBS 07-Jun-1982
0052 1 1-007 - Pass flag to EDT$$NXT_CMDCH so that it will accept repeat counts. STS 16-Jun-1982
0053 1 1-008 - Add reverse video to nokeypad command input. SMB 24-Jun-1982
0054 1 1-009 - New implementation of defined keys. JBS 12-Aug-1982
0055 1 1-010 - Add another control C check since a control C can occur
0056 1 in edt$$trn_kstr. STS 24-Nov-1982
0057 1 1-011 - Don't call EDT$$ERA_MSGLN unnecessarily. JBS 20-Jan-1983
```

EDTSKEYCOM
V04-000

EDTSKEYCOM - get a keypad command

H 6
16-Sep-1984 00:42:18
14-Sep-1984 12:23:21

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

: 58
: 59

0058 1 !--
0059 1

EDT
V04


```
: 61      0060 1 %SBTTL 'Declarations'
: 62      0061 1
: 63      0062 1 : TABLE OF CONTENTS:
: 64      0063 1 :
: 65      0064 1
: 66      0065 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 67      0504 1
: 68      0505 1 FORWARD ROUTINE
: 69      0506 1     EDT$$GET_KPADCMD;
: 70      0507 1
: 71      0508 1 :
: 72      0509 1 : INCLUDE FILES:
: 73      0510 1 :
: 74      0511 1
: 75      0512 1 REQUIRE 'EDTSRC:EDTREQ';
: 76      0647 1
: 77      0648 1 LIBRARY 'EDTSRC:KEYPADDEF';
: 78      0649 1
: 79      0650 1 :
: 80      0651 1 : MACROS:
: 81      0652 1 :
: 82      0653 1 :     NONE
: 83      0654 1 :
: 84      0655 1 : EQUATED SYMBOLS:
: 85      0656 1 :
: 86      0657 1 :     NONE
: 87      0658 1 :
: 88      0659 1 : OWN STORAGE:
: 89      0660 1 :
: 90      0661 1 :     NONE
: 91      0662 1 :
: 92      0663 1 : EXTERNAL REFERENCES:
: 93      0664 1 :
: 94      0665 1 :     In the routine
```

! Get a keypad command

```

: 96      0666 1 %SBTTL 'EDT$$GET_KPADCMD - get a keypad command'
: 97      0667 1
: 98      0668 1 GLOBAL ROUTINE EDT$$GET_KPADCMD                ! Get a keypad command
: 99      0669 1 =
100     0670 1
101     0671 1 ++
102     0672 1 FUNCTIONAL DESCRIPTION:
103     0673 1
104     0674 1 Get a keypad command. Characters are read and translated until a complete
105     0675 1 command is seen. The characters are handled as follows:
106     0676 1
107     0677 1     Keypad and control characters are translated, and if they end in a '.' then the
108     0678 1     command is complete.
109     0679 1
110     0680 1     Other characters are treated as insertions and always complete.
111     0681 1
112     0682 1 FORMAL PARAMETERS:
113     0683 1
114     0684 1     NONE
115     0685 1
116     0686 1 IMPLICIT INPUTS:
117     0687 1
118     0688 1     EDT$$T_CMD_BUF
119     0689 1     EDT$$A_CMD_BUF
120     0690 1     EDT$$G_KPAD
121     0691 1     EDT$$G_TIN_ECHOFLG
122     0692 1     EDT$$G_MSGFLG
123     0693 1
124     0694 1 IMPLICIT OUTPUTS:
125     0695 1
126     0696 1     EDT$$A_CMD_END
127     0697 1     EDT$$G_CC_DONE
128     0698 1
129     0699 1 ROUTINE VALUE:
130     0700 1
131     0701 1     1 = ok, 0 = control C, 2 = end of journal file
132     0702 1
133     0703 1 SIDE EFFECTS:
134     0704 1
135     0705 1     NONE
136     0706 1
137     0707 1 --
138     0708 1
139     0709 2 BEGIN
140     0710 2
141     0711 2 EXTERNAL ROUTINE
142     0712 2     EDT$$SC_REVID,
143     0713 2     EDT$$CHR_CC,
144     0714 2     EDT$$ERA_MSGLN : NOVALUE,
145     0715 2     EDT$$NXT_CMDCH,
146     0716 2     EDT$$TRN_KSTR,
147     0717 2     EDT$$PUT_CMDCH : NOVALUE,
148     0718 2     EDT$$KPAD_INP;
149     0719 2
150     0720 2 EXTERNAL
151     0721 2     EDT$$T_CMD_BUF,
152     0722 2     EDT$$A_CMD_BUF,
```

```

! Turn on reverse video for nokeypad commands
! Check for control C
! Erase the message line
! Get the next command character
! Build the command buffer
! Put a character in the command buffer
! Auxiliary call to EDT$$TI_RDCMDLN
```

```

! Command buffer
! Pointer to next char in command buffer
```



```
153 0723 2 EDT$$A_CMD_END,
154 0724 2 EDT$$G_KPAD,
155 0725 2 EDT$$G_CC_DONE,
156 0726 2 EDT$$G_TIN_ECHOFLG,
157 0727 2 EDT$$G_MSGFLG;
158 0728 2
159 0729 2 LOCAL
160 0730 2 C
161 0731 2 RES_TERM,
162 0732 2 SUCCEED,
163 0733 2 KEYS_SEEN;
164 0734 2
165 0735 2 KEYS_SEEN = 0;
166 0736 2
167 0737 2 + Loop until command is complete.
168 0738 2 -
169 0739 2
170 0740 2 WHILE 1 DO
171 0741 2 BEGIN
172 0742 2
173 0743 2 + Get the next command character. If the command was aborted, return.
174 0744 2 -
175 0745 2 SUCCEED = EDT$$NXT_CMDCH (C, 1);
176 0746 2
177 0747 2 IF (.SUCCEED NEQ 1) THEN RETURN (.SUCCEED);
178 0748 2
179 0749 2 +
180 0750 2 - Check for control C and abort the command.
181 0751 2 -
182 0752 2
183 0753 2 IF EDT$$CHK_CC ()
184 0754 2 THEN
185 0755 2 4 BEGIN
186 0756 2 4 EDT$$A_CMD_END = EDT$$T_CMD_BUF;
187 0757 2 4
188 0758 2 4 IF .KEYS_SEEN THEN EDT$$G_CC_DONE = 1;
189 0759 2 4
190 0760 2 4 RETURN (0);
191 0761 2 4 END;
192 0762 2
193 0763 2 KEYS_SEEN = 1;
194 0764 2 +
195 0765 2 - Erase the message line and make sure we are in reverse video mode
196 0766 2 - if this is a nokeypad command
197 0767 2 -
198 0768 2
199 0769 2 IF ((.EDT$$G_MSGFLG NEQ 0) OR .EDT$$G_TIN_ECHOFLG) THEN EDT$$ERA_MSGLN ();
200 0770 2
201 0771 2 IF ( NOT .EDT$$G_KPAD) THEN EDT$$SC_REVID ();
202 0772 2
203 0773 2 SELECTONEU .C OF
204 0774 2 SET
205 0775 2
206 0776 2 [0 TO 31] :
207 0777 2 +
208 0778 2 - Control character: if in keypad mode, translate it.
209 0779 2 -
```

```
210 0780 3
211 0781 3
212 0782 3 IF .EDT$$G_KPAD
213 0783 4 THEN
214 0784 4 BEGIN
215 0785 4 IF EDT$$TRN_KSTR (.C, 1, 0) THEN RETURN (1);
216 0786 4
217 0787 4 END
218 0788 3 ELSE
219 0789 4 BEGIN
220 0790 4 !+
221 0791 4 !- In nokeypad mode, copy characters into the buffer.
222 0792 4 !-
223 0793 4
224 0794 4 IF (EDT$$KPAD_INP (.C, ASC_K_CR, RES_TERM) EQL 2) THEN RETURN (2);
225 0795 4
226 0796 4 EDT$$A_CMD_END = .EDT$$A_CMD_BUF;
227 0797 4 RETURN (1);
228 0798 3 END;
229 0799 3
230 0800 3 [ASC_K_DEL, K_KPAD_BASE TO K_KEY_MAX] :
231 0801 3 !+
232 0802 3 !- DEL, keypad or function key: always translate.
233 0803 3 !-
234 0804 3
235 0805 3 IF EDT$$TRN_KSTR (.C, 1, 0) THEN RETURN (1);
236 0806 3
237 0807 3 [OTHERWISE] :
238 0808 3 !+
239 0809 3 !- Insertion character: make the command Ic^Z.
240 0810 3 !-
241 0811 3
242 0812 3 IF .EDT$$G_KPAD
243 0813 4 THEN
244 0814 4 BEGIN
245 0815 4 EDT$$PUT_CMDCH (%C'I', 0);
246 0816 4 EDT$$PUT_CMDCH (.C, 0);
247 0817 4 EDT$$PUT_CMDCH (ASC_K_CTRL_Z, 0);
248 0818 4 EDT$$A_CMD_END = .EDT$$A_CMD_BUF;
249 0819 4 RETURN (1);
250 0820 4 END
251 0821 3 ELSE
252 0822 4 BEGIN
253 0823 4 !+
254 0824 4 !- In nokeypad mode, copy characters into the buffer.
255 0825 4 !-
256 0826 4
257 0827 4 IF (EDT$$KPAD_INP (.C, ASC_K_CR, RES_TERM) EQL 2) THEN RETURN (2);
258 0828 4
259 0829 4 EDT$$A_CMD_END = .EDT$$A_CMD_BUF;
260 0830 4 RETURN (1);
261 0831 3 END;
262 0832 3
263 0833 3 TES;
264 0834 3
265 0835 3 !+
266 0836 3 !- Check for control C and abort the command.
```



```

: 267 0837 3 !-
: 268 0838 3
: 269 0839 3
: 270 0840 3
: 271 0841 4
: 272 0842 4
: 273 0843 4
: 274 0844 4
: 275 0845 4
: 276 0846 4
: 277 0847 3
: 278 0848 3
: 279 0849 2
: 280 0850 2
: 281 0851 2
: 282 0852 1

      IF EDT$$CHK_CC ( )
      THEN
        BEGIN
          EDT$$A_CMD_END = EDT$$T_CMD_BUF;

          IF .KEYS_SEEN THEN EDT$$G_CC_DONE = 1;

          RETURN (0);
        END;
      END;

      RETURN (1);
    END;
```

! of routine EDT\$\$GET_KPADCMD

```
.TITLE EDT$KEYCOM EDT$KEYCOM - get a keypad command
.IDENT \V04-000\
```

```
.EXTRN EDT$$SC REVID, EDT$$CHK_CC
.EXTRN EDT$$ERA_MSGLN, EDT$$NXT_CMDCH
.EXTRN EDT$$TRN_KSTR, EDT$$PUT_CMDCH
.EXTRN EDT$$KPAD_INP, EDT$$T_CMD_BUF
.EXTRN EDT$$A_CMD_BUF, EDT$$A_CMD_END
.EXTRN EDT$$G_KPAD, EDT$$G_CC_DONE
.EXTRN EDT$$G_TIN_ECHOFLG
.EXTRN EDT$$G_MSGFLG
```

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

				07FC 00000				
5A	00000000G	00	9E	00002	MOVAB	EDT\$\$A_CMD_BUF, R10		
59	00000000G	00	9E	00009	MOVAB	EDT\$\$KPAD_INP, R9		
58	00000000G	00	9E	00010	MOVAB	EDT\$\$CHK_CC, R8		
57	00000000G	00	9E	00017	MOVAB	EDT\$\$PUT_CMDCH, R7		
56	00000000G	00	9E	0001E	MOVAB	EDT\$\$A_CMD_END, R6		
55	00000000G	00	9E	00025	MOVAB	EDT\$\$G_KPAD, R5		
5E		08	C2	0002C	SUBL2	#8, SP		
		53	D4	0002F	CLRL	KEYS_SEEN	0735	
		01	DD	00031	PUSHL	#1	0745	
		04	AE	9F 00033	PUSHAB	C		
00000000G	00	02	FB	00036	CALLS	#2, EDT\$\$NXT_CMDCH		
	54	50	D0	0003D	MOVL	R0, SUCCEED		
	01	54	D1	00040	CMPL	SUCCEED, #1	0747	
		04	13	00043	BEQL	2\$		
	50	54	D0	00045	MOVL	SUCCEED, R0		
			04	00048	RET			
	68	00	FB	00049	CALLS	#0, EDT\$\$CHK_CC	0753	
	03	50	E9	0004C	BLBC	R0, 3\$		
		00AC	31	0004F	BRW	16\$		
	53	01	D0	00052	MOVL	#1, KEYS_SEEN	0763	
		00	D5	00055	TSTL	EDT\$\$G_MSGFLG	0769	
		07	12	0005B	BNEQ	4\$		
	07	00	E9	0005D	BLBC	EDT\$\$G_TIN_ECHOFLG, 5\$		

00000000G	00	00	FB	00064	4\$:	CALLS	#0, EDTSSERA_MSGLN		
	07	65	E8	0006B	5\$:	BLBS	EDTSSG_KPAD, 6\$	0771	
00000000G	00	00	FB	0006E		CALLS	#0, EDTSSSC_REVID		
	52	6E	D0	00075	6\$:	MOVL	C, R2	0773	
	1F	52	D1	00078		CMPL	R2, #31	0776	
		28	1A	0007B		BGTRU	10\$		
	11	65	E9	0007D		BLBC	EDTSSG_KPAD, 8\$	0781	
	7E	01	7D	00080	7\$:	MOVQ	#1, -(SP)	0785	
		52	DD	00083		PUSHL	R2		
00000000G	00	03	FB	00085		CALLS	#3, EDTSSSTRN_KSTR		
	66	50	E9	0008C		BLBC	R0, 15\$		
		12	11	0008F		BRB	9\$		
		04	AE	9F	00091	8\$:	PUSHAB	RES_TERM	0794
		0D	DD	00094		PUSHL	#13		
		52	DD	00096		PUSHL	R2		
	69	03	FB	00098		CALLS	#3, EDTSSKPAD_INP		
	02	50	D1	0009B		CMPL	R0, #2		
		4A	13	0009E		BEQL	13\$		
	66	6A	D0	000A0		MOVL	EDTSSA_CMD_BUF, EDTSSA_CMD_END	0796	
		6C	11	000A3	9\$:	BRB	17\$	0797	
0000007F	8F	52	D1	000A5	10\$:	CMPL	R2, #127	0800	
		D2	13	000AC		BEQL	7\$		
0000012C	8F	52	D1	000AE		CMPL	R2, #300		
		09	1F	000B5		BLSSU	11\$		
000003E7	8F	52	D1	000B7		CMPL	R2, #999		
		C0	1B	000BE		BLEQU	7\$		
	18	65	E9	000C0	11\$:	BLBC	EDTSSG_KPAD, 12\$	0812	
		7E	D4	000C3		CLRL	-(SP)	0815	
	7E	49	8F	9A	000C5	MOVZBL	#73, -(SP)		
	67	02	FB	000C9		CALLS	#2, EDTSSPUT_CMDCH		
		7E	D4	000CC		CLRL	-(SP)	0816	
		52	DD	000CE		PUSHL	R2		
	67	02	FB	000D0		CALLS	#2, EDTSSPUT_CMDCH		
	7E	1A	7D	000D3		MOVQ	#26, -(SP)	0817	
	67	02	FB	000D6		CALLS	#2, EDTSSPUT_CMDCH		
		13	11	000D9		BRB	14\$	0822	
		04	AE	9F	000DB	12\$:	PUSHAB	RES_TERM	0827
		0D	DD	000DE		PUSHL	#13		
		52	DD	000E0		PUSHL	R2		
	69	03	FB	000E2		CALLS	#3, EDTSSKPAD_INP		
	02	50	D1	000E5		CMPL	R0, #2		
		04	12	000E8		BNEQ	14\$		
	50	02	D0	000EA	13\$:	MOVL	#2, R0		
		04	000ED			RET			
	50	01	D0	000EE	14\$:	MOVL	#1, R0	0830	
	66	6A	D0	000F1		MOVL	EDTSSA_CMD_BUF, EDTSSA_CMD_END	0818	
		04	000F4			RET		0822	
	68	00	FB	000F5	15\$:	CALLS	#0, EDTSSCHK_CC	0839	
	03	50	E8	000F8		BLBS	R0, 16\$		
		FF33	31	000FB		BRW	1\$		
	66	00	9E	000FE	16\$:	MOVAB	EDTSSCMD_BUF, EDTSSA_CMD_END	0842	
	0D	53	E9	00105		BLBC	KEYS_SEEN, 18\$	0844	
00000000G	00	01	D0	00108		MOVL	#1, EDTSSG_CC_DONE		
		04	11	0010F		BRB	18\$	0846	
	50	01	D0	00111	17\$:	MOVL	#1, R0	0851	
		04	00114			RET			
		50	D4	00115	18\$:	CLRL	R0	0852	

EDTSKEYCOM
V04-000

EDTSKEYCOM - get a keypad command
EDT\$\$GET_KPADCMD - get a keypad command

B 7
16-Sep-1984 00:42:18
14-Sep-1984 12:23:21

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]KEYCOM.BLI;1 Page 9
(3)

04 00117

RET

; Routine Size: 280 bytes, Routine Base: _EDT\$CODE + 0000

: 283
: 284 0853 1
0854 1 !<BLF/PAGE>

EDT\$KEYCOM
V04-000

EDT\$KEYCOM - get a keypad command
EDT\$\$GET_KPADCMD - get a keypad command

C 7
16-Sep-1984 00:42:18
14-Sep-1984 12:23:21

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]KEYCOM.BLI;1 Page 10
(4)

: 286 0855 1 END
: 287 0856 1
: 288 0857 0 ELUDOM

! of module EDT\$KEYCOM

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	280	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	3	0	40	00:00.2
-\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
-\$255\$DUA28:[EDT.SRC]KEYPADDEF.L32;1	34	2	5	7	00:00.1

COMMAND QUALIFIERS

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

: Size: 280 code + 0 data bytes
: Run Time: 00:16.4
: Elapsed Time: 00:19.8
: Lines/CPU Min: 3127
: Lexemes/CPU-Min: 8405
: Memory Used: 110 pages
: Compilation Complete

0135 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY