


```

1 0001 0 XTITLE 'EDTS$KEYPUTCHR - put a character in the command buffer'
2 0002 0 MODULE EDTS$KEYPUTCHR ( ! Put a character in the command buffer
3 0003 0 IDENT = 'V04-000' ! File: KEYPUTCHR.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 Put a character in the command buffer.
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: April 7, 1979
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. DJS 24-Feb-1981. This module was created by
45 0045 1 extracting routine PUT_COM_CHAR from module KEYTRAN.
46 0046 1 1-002 - Regularize headers. JBS 10-Mar-1981
47 0047 1 1-003 - Use new message codes. JBS 04-Aug-1981
48 0048 1 1-004 - Use MSG_BELL for errors. SMB 30-Jun-1982
49 0049 1 1-005 - Improve the appearance of the listing. JBS 14-Jun-1983
50 0050 1 --
51 0051 1
    
```

**F

```
.. 53 0052 1 %SBTTL 'Declarations'  
.. 54 0053 1 |  
.. 55 0054 1 | TABLE OF CONTENTS:  
.. 56 0055 1 |  
.. 57 0056 1 |  
.. 58 0057 1 REQUIRE 'EDTSRC:TRAROUNAM';  
.. 59 0496 1 |  
.. 60 0497 1 FORWARD ROUTINE  
.. 61 0498 1 EDT$SPUT_CMDCH : NOVALUE;  
.. 62 0499 1 |  
.. 63 0500 1 |  
.. 64 0501 1 | INCLUDE FILES:  
.. 65 0502 1 |  
.. 66 0503 1 |  
.. 67 0504 1 REQUIRE 'EDTSRC: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$$PUT_CMDCH - put a character in the command buffer'
: 87 0657 1
: 88 0658 1 GLOBAL ROUTINE EDT$$PUT_CMDCH (                ! Put a character in the command buffer
: 89 0659 1     C,                                       ! Character to put
: 90 0660 1     FORCE_ECHO                               ! 1 = echo the character
: 91 0661 1     ) : NOVALUE =
: 92 0662 1
: 93 0663 1 !++
: 94 0664 1 !FUNCTIONAL DESCRIPTION:
: 95 0665 1
: 96 0666 1     This routine places a character in the command buffer at EDT$$A_CMD_BUF and
: 97 0667 1     advances EDT$$A_CMD_BUF .
: 98 0668 1
: 99 0669 1 !FORMAL PARAMETERS:
:100 0670 1
:101 0671 1     C                The character to append to the command buffer.
:102 0672 1
:103 0673 1     FORCE_ECHO       If 1, then echo the character.
:104 0674 1
:105 0675 1 !IMPLICIT INPUTS:
:106 0676 1
:107 0677 1     EDT$$T_CMD_BUF
:108 0678 1     EDT$$A_CMD_BUF
:109 0679 1     EDT$$A_CMD_END
:110 0680 1
:111 0681 1 !IMPLICIT OUTPUTS:
:112 0682 1
:113 0683 1     EDT$$A_CMD_BUF
:114 0684 1
:115 0685 1 !ROUTINE VALUE:
:116 0686 1
:117 0687 1     NONE
:118 0688 1
:119 0689 1 !SIDE EFFECTS:
:120 0690 1
:121 0691 1     NONE
:122 0692 1
:123 0693 1 !--
:124 0694 1
:125 0695 1     BEGIN
:126 0696 1
:127 0697 1     EXTERNAL ROUTINE
:128 0698 1     EDT$$MSG BELL,
:129 0699 1     EDT$$TI_ECHOCH;
:130 0700 1
:131 0701 1     EXTERNAL
:132 0702 1     EDT$$T_CMD_BUF,                ! Command buffer
:133 0703 1     EDT$$A_CMD_BUF,                ! Pointer to next char in command buffer
:134 0704 1     EDT$$A_CMD_END;              ! Pointer to end of info in command buffer
:135 0705 1
:136 0706 1     MESSAGES ((COMBUFEXH));
:137 0707 1 !+
:138 0708 1     Check to make sure there is room, if not then put out a message and
:139 0709 1     flush the entire buffer.
:140 0710 1 !-
:141 0711 1
:142 0712 1     IF CH$PTR_EQL (.EDT$$A_CMD_BUF, .EDT$$A_CMD_END)
```

```

: 143      0713      2      THEN
: 144      0714      2      BEGIN
: 145      0715      2      EDT$$MSG BELL (EDT$ COMBUFEXH);
: 146      0716      2      EDT$$A_CMD_BUF = CH$PTR (EDT$$I_CMD_BUF);
: 147      0717      2      END
: 148      0718      2      ELSE
: 149      0719      2      BEGIN
: 150      0720      2      !+
: 151      0721      2      !- Echo the character if necessary then move it to the buffer.
: 152      0722      2
: 153      0723      2
: 154      0724      2      IF .FORCE_ECHO THEN EDT$$TI_ECHOCH (.C);
: 155      0725      2
: 156      0726      2      CH$WCHAR_A (.C, EDT$$A_CMD_BUF);
: 157      0727      2      END;
: 158      0728      2
: 159      0729      2      END;

```

! of routine EDT\$\$PUT_CMDCH

.TITLE EDTSKEYPUTCHR EDTSKEYPUTCHR - put a character i
n the command

.IDENT \V04-000\

.EXTRN EDT\$\$MSG BELL, EDT\$\$TI_ECHOCH
.EXTRN EDT\$\$I_CMD_BUF, EDT\$\$A_CMD_BUF
.EXTRN EDT\$\$A_CMD_END, EDT\$ COMBUFEXH

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

			0004	0000	.ENTRY	EDT\$\$PUT_CMDCH, Save R2	: 0658
	52	00000000G	00	9E 00002	MOVAB	EDT\$\$A_CMD_BUF, R2	: 0712
00000000G	00		62	D1 00009	CMPL	EDT\$\$A_CMD_BUF, EDT\$\$A_CMD_END	: 0715
			15	12 00010	BNEQ	1\$: 0716
00000000G	00	00000000G	8F	DD 00012	PUSHL	#EDT\$ COMBUFEXH	: 0712
	62	00000000G	01	FB 00018	CALLS	#1, EDT\$\$MSG BELL	: 0712
			00	9E 0001F	MOVAB	EDT\$\$I_CMD_BUF, EDT\$\$A_CMD_BUF	: 0724
			04	00026	RET		: 0726
	0A	08	AC	E9 00027	BLBC	FORCE_ECHO, 2\$: 0729
			04	AC DD 0002B	PUSHL	C	
00000000G	00		01	FB 0002E	CALLS	#1, EDT\$\$TI ECHOCH	
	50		62	D0 00035	MOVL	EDT\$\$A_CMD_BUF, R0	
	60	04	AC	90 00038	MOVB	C, (R0)	
			62	D6 0003C	INCL	EDT\$\$A_CMD_BUF	
			04	0003E	RET		

: Routine Size: 63 bytes, Routine Base: _EDT\$CODE + 0000

```

: 160      0730      1
: 161      0731      1 !<BLF/PAGE>

```

EDT\$KEYPUTCHR
V04-000

EDT\$KEYPUTCHR - put a character in the command
EDT\$\$PUT_CMDCH - put a character in the comman

L 11
16-Sep-1984 00:45:40
14-Sep-1984 12:23:27

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]KEYPUTCHR.BLI;1

Page 5
(4)

: 163 0732 1 END
: 164 0733 1
: 165 0734 0 ELUDOM

! of module EDT\$KEYPUTCHR

PSECT SUMMARY

Name Bytes Attributes
:_EDT\$CODE 63 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	2	0	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\$:KEYPUTCHR/OBJ=OBJ\$:KEYPUTCHR MSRC\$:KEYPUTCHR.BLI/UPDATE=(ENH\$:KEYPUTCHR)

: Size: 63 code + 0 data bytes
: Run Time: 00:09.9
: Elapsed Time: 00:12.2
: Lines/CPU Min: 4471
: Lexemes/CPU-Min: 12712
: Memory Used: 61 pages
: Compilation Complete

EDT
V04

KEYPADDEF LIS	LDIVISION LIS
KEYDEFKEY LIS	KEYFMTSTR LIS
KEYCHR LIS	KEYIMMNP LIS
KEYCOM LIS	KEYPUTCHR LIS
KEYPAD LIS	LCLEAR LIS
KEYTRNCHR LIS	LCOUNT LIS
	LDEFM LIS
	LFILL LIS
	LDELETE LIS
	LFLNO LIS