

EEEEEEEEEEEEEEEE	DDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDD	TTT

EXE

MOO

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

EDI

```

PPPPPPPP  RRRRRRRR  GGGGGGGG  EEEEEEEEE  TTTTTTTTTT  CCCCCCCC  HH      HH  RRRRRRRR
PPPPPPPP  RRRRRRRR  GGGGGGGG  EEEEEEEEE  TTTTTTTTTT  CCCCCCCC  HH      HH  RRRRRRRR
PP      PP  RR      RR  GG      GG  EE      EE  TT      TT  CC      CC  HH      HH  RR      RR
PP      PP  RR      RR  GG      GG  EE      EE  TT      TT  CC      CC  HH      HH  RR      RR
PP      PP  RR      RR  GG      GG  EE      EE  TT      TT  CC      CC  HH      HH  RR      RR
PPPPPPPP  RRRRRRRR  GGGGGGGG  EEEEEEEEE  TTTTTTTTTT  CCCCCCCC  HH      HH  RRRRRRRR
PPPPPPPP  RRRRRRRR  GGGGGGGG  EEEEEEEEE  TTTTTTTTTT  CCCCCCCC  HH      HH  RRRRRRRR
PP      RR  RR      GG  GGGGGG  EE      EE  TT      TT  CC      CC  HH      HH  RR      RR
PP      RR  RR      GG  GGGGGG  EE      EE  TT      TT  CC      CC  HH      HH  RR      RR
PP      RR      RR  GG      GG  EE      EE  TT      TT  CC      CC  HH      HH  RR      RR
PP      RR      RR  GG      GG  EE      EE  TT      TT  CC      CC  HH      HH  RR      RR
PP      RR      RR  GG      GG  EE      EE  TT      TT  CC      CC  HH      HH  RR      RR
PP      RR      RR  GG      GG  EE      EE  TT      TT  CC      CC  HH      HH  RR      RR
PP      RR      RR  GG      GG  EE      EE  TT      TT  CC      CC  HH      HH  RR      RR

```

```

LL          IIIIIII  SSSSSSSS
LL          IIIIIII  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
LL          II           SS
LLLLLLLLLLL IIIIIII  SSSSSSSS
LLLLLLLLLLL IIIIIII  SSSSSSSS

```

....
....
....
....

.....

```

1 0001 0 %TITLE 'EDT$PRGETCHR - get next parse character'
2 0002 0 MODULE EDT$PRGETCHR ( ! Get next parse character
3 0003 0 IDENT = 'V04-000' ! File: PRGETCHR.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 Get the next character to be parsed.
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: December 12, 1978
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. DJS 25-Feb-1981. This module was created by
45 0045 1 extracting routine GET_CHAR from module PARSER.
46 0046 1 1-002 - Regularize headers. JBS 12-Mar-1981
47 0047 1 1-003 - Change alphabetic processing. JBS 19-Jul-1982
48 0048 1 1-004 - Don't strip diacritical marks. JBS 13-Dec-1982
49 0049 1 1-005 - Add VT220 support conditional. JBS 11-Feb-1983
50 0050 1 --
51 0051 1

```

```
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$PA_GETCH : 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 LIBRARY 'EDTSRC:SUPPORTS';  
70 0641 1  
71 0642 1 :  
72 0643 1 : MACROS:  
73 0644 1 :  
74 0645 1 : NONE  
75 0646 1 :  
76 0647 1 : EQUATED SYMBOLS:  
77 0648 1 :  
78 0649 1 : NONE  
79 0650 1 :  
80 0651 1 : OWN STORAGE:  
81 0652 1 :  
82 0653 1 : NONE  
83 0654 1 :  
84 0655 1 : EXTERNAL REFERENCES:  
85 0656 1 :  
86 0657 1 : In the routine
```

```
88 0658 1 %SBTTL 'EDT$$PA_GETCH - get the next char to parse'
89 0659 1
90 0660 1 GLOBAL ROUTINE EDT$$PA_GETCH          ! Get the next character to parse
91 0661 1 : NOVALUE =
92 0662 1
93 0663 1 ++
94 0664 1 FUNCTIONAL DESCRIPTION:
95 0665 1
96 0666 1     Get the next character from the input line.  If we are pointing at the
97 0667 1     end of the record, return a '!', otherwise, get the character pointed
98 0668 1     to by EDT$$A_CMD_BUF and bump EDT$$A_CMD_BUF.  The character
99 0669 1     is placed in the global EDT$$C_PA_CH.  Lower case alphabetic are converted
100 0670 1     to upper case.
101 0671 1
102 0672 1 FORMAL PARAMETERS:
103 0673 1
104 0674 1     NONE
105 0675 1
106 0676 1 IMPLICIT INPUTS:
107 0677 1
108 0678 1     EDT$$A_CMD_BUF
109 0679 1     EDT$$A_CMD_END
110 0680 1
111 0681 1 IMPLICIT OUTPUTS:
112 0682 1
113 0683 1     EDT$$C_PA_CH
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 2 BEGIN
126 0696 2
127 0697 2 EXTERNAL
128 0698 2     EDT$$A_CMD_BUF,          ! Pointer into command buffer.
129 0699 2     EDT$$A_CMD_END,        ! Pointer to end of current command.
130 0700 2
131 L 0701 2 %IF SUPPORT_VT220
132 0702 2 %THEN
133 0703 2     EDT$$B_CHAR_INFO : BLOCKVECTOR [256, 1, BYTE], ! table of character information
134 0704 2 %FI
135 0705 2
136 0706 2     EDT$$C_PA_CH;          ! the currently being processed character
137 0707 2
138 0708 2 LOCAL
139 0709 2     CHAR;
140 0710 2
141 0711 2 IF CH$PTR_GTR (.EDT$$A_CMD_BUF, .EDT$$A_CMD_END)
142 0712 2 THEN
143 0713 2     EDT$$C_PA_CH = %C'!'
144 0714 2 ELSE
```

```

: 145      0715      3          BEGIN
: 146      0716      3          CHAR = CH$RCHAR_A (EDT$$A_CMD_BUF);
: 147      0717      3
: 148      L 0718      4 %IF ( NOT SUPPORT_VT220)
: 149      U 0719      3 %THEN
: 150      U 0720      3          CHAR = .CHAR AND %X'7F';
: 151      U 0721      3 %FI
: 152      U 0722      3
: 153      L 0723      3 %IF SUPPORT_VT220
: 154      U 0724      3 %THEN
: 155      U 0725      3
: 156      U 0726      3          IF .EDT$$B_CHAR_INFO [.CHAR, 0, 0, 1, 0]          ! Lower case letter
: 157      U 0727      3          THEN
: 158      U 0728      3 %ELSE
: 159      U 0729      3
: 160      U 0730      3          IF ((.CHAR GEQ %C'a') AND (.CHAR LEQ %C'z'))
: 161      U 0731      3          THEN
: 162      U 0732      3 %FI
: 163      U 0733      3
: 164      U 0734      3          EDT$$C_PA_CH = .CHAR - %C'a' + %C'A'
: 165      U 0735      3          ELSE
: 166      U 0736      3          EDT$$C_PA_CH = .CHAR;
: 167      U 0737      3
: 168      U 0738      3          END;
: 169      U 0739      3
: 170      U 0740      1          END;

```

! of routine EDT\$\$PA_GETCH

```

.TITLE EDT$PRGETCHR EDT$PRGETCHR - get next parse char
       acter
.IDENT \V04-000\
.EXTRN EDT$$A_CMD_BUF, EDT$$A_CMD_END
.EXTRN EDT$$B_CHAR_INFO
.EXTRN EDT$$C_PA_CH
.PSECT _EDT$CODE,NOWRT, SHR, PIC.2

```

		000C 00000		.ENTRY EDT\$\$PA_GETCH, Save R2,R3	: 0660
	53 00000000G	00 9E 00002		MOVAB EDT\$\$A_CMD_BUF, R3	:
	52 00000000G	00 9E 00009		MOVAB EDT\$\$C_PA_CH, R2	:
	00000000G 00	63 D1 00010		CML EDT\$\$A_CMD_BUF, EDT\$\$A_CMD_END	: 0711
		04 1B 00017		BLEQU 1\$:
	62	21 D0 00019		MOVL #33, EDT\$\$C_PA_CH	: 0713
		04 0001C		RET	:
	51	63 D0 0001D 1\$:		MOVL EDT\$\$A_CMD_BUF, R1	: 0716
	50	61 9A 00020		MOVZBL (R1), CHAR	:
		63 D6 00023		INCL EDT\$\$A_CMD_BUF	:
	05 00000000G0040	00 E1 00025		BBC #0, EDT\$\$B_CHAR_INFO[CHAR], 2\$: 0726
	62	E0 A0 9E 0002E		MOVAB -32(R0), EDT\$\$C_PA_CH	: 0734
		04 00032		RET	:
	62	50 D0 00033 2\$:		MOVL CHAR, EDT\$\$C_PA_CH	: 0736
		04 00036		RET	: 0740

; Routine Size: 55 bytes. Routine Base: _EDT\$CODE + 0000

EDT\$PRGETCHR
V04-000

EDT\$PRGETCHR - get next parse character
EDT\$SPA_GETCH - get the next char to parse

E 15
16-Sep-1984 01:17:10
14-Sep-1984 12:24:09

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

ED:
VO:

: 171
: 172

0741 1
0742 1 !<BLF/PAGE>

: |
:
:

EDT\$PRGETCHR
V04-000

EDT\$PRGETCHR - get next parse character
EDT\$\$PA_GETCH - get the next char to parse

F 15
16-Sep-1984 01:17:10
14-Sep-1984 12:24:09

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]PRGETCHR.BLI;1 (4) Page 6

ED
VO

: 174
: 175
: 176

0743 1 END
0744 1
0745 0 ELUDOM

! of module EDT\$PRGETCHR

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	55	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	1	0	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
_\$255\$DUA28:[EDT.SRC]SUPPORTS.L32;1	2	1	50	5	00:00.1

COMMAND QUALIFIERS

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

: Size: 55 code + 0 data bytes
: Run Time: 00:10.1
: Elapsed Time: 00:13.5
: Lines/CPU Min: 4425
: Lexemes/CPU-Min: 12124
: Memory Used: 58 pages
: Compilation Complete

MCGETLIN LIS	MCRIGHT LIS	MCCHANGE LIS	MACCAL LIS	NOOPEN LIS	PRAPPNUM LIS	PRGETTOK LIS	PRISTOK LIS
MCDOWN LIS	MCLEFT LIS	MCBOTTOM LIS	LXPRINT LIS	MCTOP LIS	MSGTXT LIS	PAUDIT LIS	PRGETCHR LIS
LXCOM LIS	MAIN LIS	MCUP LIS					