

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

0001 0 %TITLE 'EDT$PRMACCAL - test for macro token'
0002 0 MODULE EDT$PRMACCAL ( ! Test for macro token
0003 0 IDENT = 'V04-000' ! File: PRMACCAL.BLI Edit: JBS1006
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 Test for macro token.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: December 12, 1978
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 25-Feb-1981. This module was created by
0045 1 extracting routine MACRO CALL from module PARSER.
0046 1 1-002 - Regularize headers. JBS 12-Mar-1981
0047 1 1-003 - Implement new TBCB with macro pointers inside. Remove
0048 1 references to the MAC BLOCK. TMV 6-Aug-81
0049 1 1-004 - Remove call to digit testing routine, replace with a
0050 1 test based on a table of character information. JBS 20-Jul-1982
0051 1 1-005 - Add VT220 support conditional. JBS 11-Feb-1983
0052 1 1-006 - Remove an extra dot added in edit 1-005. JBS 03-Mar-1983
0053 1 --
0054 1

```

```
: 56 0055 1 %SBTTL 'Declarations'  
: 57 0056 1  
: 58 0057 1 | TABLE OF CONTENTS:  
: 59 0058 1 |  
: 60 0059 1  
: 61 0060 1 REQUIRE 'EDTSRC:TRAROUNAM';  
: 62 0499 1  
: 63 0500 1 FORWARD ROUTINE  
: 64 0501 1 EDT$SPA_TSTMACCAL;  
: 65 0502 1  
: 66 0503 1 |  
: 67 0504 1 | INCLUDE FILES:  
: 68 0505 1 |  
: 69 0506 1  
: 70 0507 1 REQUIRE 'EDTSRC:EDTREQ';  
: 71 0642 1  
: 72 0643 1 REQUIRE EDTSRC:PARLITS';  
: 73 0927 1  
: 74 0928 1 LIBRARY 'EDTSRC:SUPPORTS';  
: 75 0929 1  
: 76 0930 1 |  
: 77 0931 1 | MACROS:  
: 78 0932 1 |  
: 79 0933 1 | NONE  
: 80 0934 1 |  
: 81 0935 1 | EQUATED SYMBOLS:  
: 82 0936 1 |  
: 83 0937 1 | NONE  
: 84 0938 1 |  
: 85 0939 1 | OWN STORAGE:  
: 86 0940 1 |  
: 87 0941 1 | NONE  
: 88 0942 1 |  
: 89 0943 1 | EXTERNAL REFERENCES:  
: 90 0944 1 |  
: 91 0945 1 | In the routine
```

```

93 0946 1 %SBTTL 'EDT$$PA_TSTMACCAL - test for macro token'
94 0947 1
95 0948 1 GLOBAL ROUTINE EDT$$PA_TSTMACCAL          ! Test for macro token
96 0949 1 =
97 0950 1
98 0951 1 !++
99 0952 1 FUNCTIONAL DESCRIPTION:
100 0953 1
101 0954 1 This routine checks to see if the current token is the name of a macro,
102 0955 1 and returns 1 if it is, 0 if not. A list of macro description records is
103 0956 1 scanned and the current token is compared against each of them.
104 0957 1
105 0958 1 FORMAL PARAMETERS:
106 0959 1
107 0960 1 NONE
108 0961 1
109 0962 1 IMPLICIT INPUTS:
110 0963 1
111 0964 1 EDT$$A_CMD_BUF
112 0965 1 EDT$$A_BUF_LST
113 0966 1 EDT$$G_PA_CURCMD
114 0967 1 EDT$$A_PA_CURTOK
115 0968 1 EDT$$G_PA_CURTOKLEN
116 0969 1 EDT$$G_PA_TOKCLASS
117 0970 1
118 0971 1 IMPLICIT OUTPUTS:
119 0972 1
120 0973 1 NONE
121 0974 1
122 0975 1 ROUTINE VALUE:
123 0976 1
124 0977 1 1 = current token is a macro
125 0978 1 0 = current token is not a macro, or the parse stack overflowed
126 0979 1
127 0980 1 SIDE EFFECTS:
128 0981 1
129 0982 1 NONE
130 0983 1
131 0984 1 --
132 0985 1
133 0986 2 BEGIN
134 0987 2
135 0988 2 EXTERNAL ROUTINE
136 0989 2 EDT$$PA_SCANTOK : NOVALUE, ! Get the next token
137 0990 2 EDT$$PA_GETCH : NOVALUE, ! Get the next character from the input line
138 0991 2 EDT$$PA_SEMRUT; ! Semantic routines referenced in the parser tables
139 0992 2
140 0993 2 EXTERNAL
141 0994 2 EDT$$A_CMD_BUF, ! Pointer into command buffer.
142 0995 2 EDT$$A_BUF_LST, ! Header for all buffers
143 0996 2 EDT$$G_PA_CURCMD : REF NODE_BLOCK,
144 0997 2 EDT$$A_PA_CURTOK, ! start of the current token
145 0998 2 EDT$$G_PA_CURTOKLEN, ! length of current token
146 0999 2
147 .7 L 1000 2 %IF SUPPORT_VT220
148 .48 1001 2 %THEN
149 1002 2 EDT$$B_CHAR_INFO : BLOCKVECTOR [256, 1, BYTE], ! Information about characters

```

```
150 1003 2 %FI
151 1004 2
152 1005 2 EDT$$G_PA_TOKCLASS; ! class of current token
153 1006 2
154 1007 2 LOCAL
155 1008 2 CP,
156 1009 2 CH,
157 1010 2 LEN,
158 1011 2 BUF : REF TBCB_BLOCK;
159 1012 2
160 1013 2 IF (.EDT$$G_PA_TOKCLASS EQL CL_NAME)
161 1014 2 THEN
162 1015 2 BEGIN
163 1016 2 LEN = .EDT$$G_PA_CURTOKLEN;
164 1017 2 CP = CH$PTR (.EDT$$A_PA_CURTOK, .EDT$$G_PA_CURTOKLEN);
165 1018 2 CH = CH$RCHAR (.CP);
166 1019 2
167 L 1020 2 %IF SUPPORT_VT220
168 1021 2 %THEN
169 1022 2
170 1023 2 WHILE (.EDT$$B_CHAR_INFO [CH, 0, 0, 8, 0] EQL %X'F0') ! Digit
171 1024 2 DO
172 U 1025 2 %ELSE
173 U 1026 2
174 U 1027 2 WHILE ((.CH GEQ %C'0') AND (.CH LEQ %C'9')) DO
175 1028 2 %FI
176 1029 2
177 1030 2 BEGIN
178 1031 2 CH = CH$A_RCHAR (CP);
179 1032 2 LEN = .LEN + 1;
180 1033 2 END;
181 1034 2
182 1035 2 BUF = .EDT$$A_BUF_LST;
183 1036 2
184 1037 2 WHILE (.BUF NEQ 0) DO
185 1038 2 BEGIN
186 1039 2
187 1040 2 IF (CH$EQL (.LEN, .EDT$$A_PA_CURTOK, .BUF [TBCB_NAME_LEN], BUF [TBCB_NAME])) AND !
188 1041 2 (.BUF [TBCB_IS_MAC] EQL T)
189 1042 2 THEN
190 1043 2 BEGIN
191 1044 2 !+
192 1045 2 ! We have found a macro corresponding to this token.
193 1046 2 !-
194 1047 2
195 1048 2 IF (EDT$SPA_SEMRUT (INI_COM, COM_MAC_CALL) EQL 0) THEN RETURN (0);
196 1049 2
197 1050 2 EDT$$G_PA_CURCMD [RANGE1] = .BUF;
198 1051 2 EDT$$A_CMD_BUF = .CP;
199 1052 2 EDT$SPA_GETCH ();
200 1053 2 EDT$SPA_SCANTOK ();
201 1054 2 RETURN (1);
202 1055 2 END;
203 1056 2
204 1057 2 !+
205 1058 2 ! Update the pointer to the next buffer which is defined as a macro.
206 1059 2 !-
```

```

: 207      1060      4          BUF = .BUF [TBCB_NEXT_BUF];
: 208      1061      4          END;
: 209      1062      4
: 210      1063      4          END;
: 211      1064      4
: 212      1065      4          RETURN (0);
: 213      1066      1          END;

```

! of routine EDT\$\$PA_TSTMACCAL

```

.TITLE EDT$PRMACCAL EDT$PRMACCAL - test for macro toke
.IDENT \V04-000\

```

```

.EXTRN EDT$$PA_SCANTOK
.EXTRN EDT$$PA_GETCH, EDT$$PA_SEMRUT
.EXTRN EDT$$A_CMD_BUF, EDT$$A_BUF_LST
.EXTRN EDT$$G_PA_CURCMD
.EXTRN EDT$$A_PA_CURTOK
.EXTRN EDT$$G_PA_CURTOKLEN
.EXTRN EDT$$B_CHAR_INFO
.EXTRN EDT$$G_PA_TOKCLASS

```

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

```

.OOFL 00000
57 00000000G 00 9E 00002
00000000G 00 D5 00009
7D 12 0000F
50 00000000G 00 D0 00011
56 50 D0 00018
55 67 50 C1 0001B
50 65 9A 0001F
FO 8F 00000000G0040 91 00022 1$:
09 12 0002B
55 55 D6 0002D
50 65 9A 0002F
56 D6 00032
EC 11 00034
54 00000000G 00 D0 00036 2$:
4F 13 0003D 3$:
51 67 D0 0003F
50 2C A4 9A 00042
50 56 2D 00046
50 2D A4 0004B
39 12 0004D
01 2B A4 91 0004F
33 12 00053
19 DD 00055
14 DD 00057
00000000G 00 02 FB 00059
50 D5 00060
2A 13 00062
50 00000000G 00 D0 00064
04 A0 54 D0 0006B
00000000G 00 55 D0 0006F
00000000G 00 00 FB 00076
00000000G 00 00 FB 0007D

.ENTRY EDT$$PA_TSTMACCAL, Save R2,R3,R4,R5,R6,R7 : 0948
MOVAB EDT$$A_PA_CURTOK, R7 : 1013
TSTL EDT$$G_PA_TOKCLASS : 1016
BNEQ 5$ : 1017
MOVL EDT$$G_PA_CURTOKLEN, R0 : 1018
MOVL R0, LEN : 1019
ADDL3 R0, EDT$$A_PA_CURTOK, CP : 1020
MOVZBL (CP), CH : 1021
CMPB EDT$$B_CHAR_INFO[CH], #240 : 1022
BNEQ 2$ : 1023
INCL CP : 1031
MOVZBL (CP), CH : 1032
INCL LEN : 1033
BRB 1$ : 1034
MOVL EDT$$A_BUF_LST, BUF : 1035
BEQL 5$ : 1037
MOVL EDT$$A_PA_CURTOK, R1 : 1040
MOVZBL 44(BUF), R0
CMPCS LEN, (R1), #0, R0, 45(BUF)
BNEQ 4$ : 1041
CMPB 43(BUF), #1 : 1042
BNEQ 4$ : 1043
PUSHL #25 : 1048
PUSHL #20
CALLS #2, EDT$$PA_SEMRUT
TSTL R0
BEQL 5$
MOVL EDT$$G_PA_CURCMD, R0 : 1050
MOVL BUF, 4(R0)
MOVL CP, EDT$$A_CMD_BUF : 1051
CALLS #0, EDT$$PA_GETCH : 1052
CALLS #0, EDT$$PA_SCANTOK : 1053

```

EDT\$PRMACCAL
V04-000

EDT\$PRMACCAL - test for macro token
EDT\$SPA_TSTMACCAL - test for macro token

N 1
16-Sep-1984 01:19:30
14-Sep-1984 12:24:10

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

Page 6
(3)

50	01	D0	00084	MOVL	#1, R0	:	1054	
		04	00087	RET		:		
54	26	A4	D0 00088	4\$:	MOVL	38(BUF), BUF	:	1060
		AF	11 0008C		BRB	3\$:	1057
		50	D4 0008E	5\$:	CLRL	R0	:	1066
		04	00090	RET		:		

: Routine Size: 145 bytes. Routine Base: _EDT\$CODE + 0000

: 214 1067 1
: 215 1068 1 !<BLF/PAGE>

EDT\$
V04

EDT\$PRMACCAL
V04-000

EDT\$PRMACCAL - test for macro token
EDT\$\$PA_TSTMACCAL - test for macro token

B 2
16-Sep-1984 01:19:30
14-Sep-1984 12:24:10

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]PRMACCAL.BLI;1 (4) Page 7

EDT
V04

: 217 1069 1 END
: 218 1070 1
: 219 1071 0 ELUDOM

! of module EDT\$PRMACCAL

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	145	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	75	19	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=LIS\$:PRMACCAL/OBJ=OBJ\$:PRMACCAL MSRC\$:PRMACCAL.BLI/UPDATE=(ENH\$:PRMACCAL)

: Size: 145 code + 0 data bytes
: Run Time: 00:16.7
: Elapsed Time: 00:20.7
: Lines/CPU Min: 3857
: Lexemes/CPU-Min: 12522
: Memory Used: 127 pages
: Compilation Complete

: R

0138 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

