


```

1 0001 0 MODULE PAS$EOF2 ( %TITLE 'Inquire if end-of-file'
2 0002 0 IDENT = '1-001' ! File: PASEOF2.B32 Edit: SBL1001
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 **
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This procedure implements the Pascal EOF procedure. It
36 0036 1 is not called directly from compiled code, but rather can
37 0037 1 be passed as a procedure by users.
38 0038 1
39 0039 1 ENVIRONMENT: User mode - AST reentrant
40 0040 1
41 0041 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 1-001 - Original. SBL 1-April-1981
46 0046 1 --

```

```
.. 48      0047 1 %SBTTL 'Declarations'  
.. 49      0048 1  
.. 50      0049 1 : PROLOGUE DEFINITIONS:  
.. 51      0050 1 :  
.. 52      0051 1 :  
.. 53      0052 1 REQUIRE 'RTLIN:PASPROLOG';           . Externals, linkages, PSECTs, structures  
.. 54      0116 1  
.. 55      0117 1 :  
.. 56      0118 1 : TABLE OF CONTENTS:  
.. 57      0119 1 :  
.. 58      0120 1  
.. 59      0121 1 FORWARD ROUTINE  
.. 60      0122 1 PASSEOF2;           ! Inquire if end-of-file  
.. 61      0123 1  
.. 62      0124 1 :  
.. 63      0125 1 : MACROS:  
.. 64      0126 1 :  
.. 65      0127 1 : NONE  
.. 66      0128 1 :  
.. 67      0129 1 : EQUATED SYMBOLS:  
.. 68      0130 1 :  
.. 69      0131 1 : NONE  
.. 70      0132 1 :  
.. 71      0133 1 : FIELDS:  
.. 72      0134 1 :  
.. 73      0135 1 : NONE  
.. 74      0136 1 :  
.. 75      0137 1 : OWN STORAGE:  
.. 76      0138 1 :  
.. 77      0139 1 : NONE  
.. 78      0140 1 :
```

```

80 0141 1 %SBTTL 'PAS$EOF2 - Inquire if end-of-file'
81 0142 1 GLOBAL ROUTINE PAS$EOF2 (           ! Inquire if end-of-file
82 0143 1 PFV: REF $PASS$PFV_FILE_VARIABLE  ! File variable
83 0144 1 ) =
84 0145 1
85 0146 1 ++
86 0147 1 FUNCTIONAL DESCRIPTION:
87 0148 1
88 0149 1     This procedure returns a boolean value TRUE if the given file
89 0150 1     is at end-of-file, FALSE otherwise.
90 0151 1
91 0152 1 CALLING SEQUENCE:
92 0153 1
93 0154 1     Eof.wv.v = PAS$EOF2 (PFV.mr.r)
94 0155 1
95 0156 1 FORMAL PARAMETERS:
96 0157 1
97 0158 1     PFV           - The Pascal File Variable (PFV) passed by reference.
98 0159 1                 The structure of the PFV is defined in PAS$PFV.REQ.
99 0160 1
100 0161 1 IMPLICIT INPUTS:
101 0162 1
102 0163 1     NONE
103 0164 1
104 0165 1 IMPLICIT OUTPUTS:
105 0166 1
106 0167 1     NONE
107 0168 1
108 0169 1 ROUTINE VALUE:
109 0170 1
110 0171 1     Boolean TRUE (1) if file is at end-of-file.
111 0172 1     Boolean FALSE (0) if file is not at end-of-file.
112 0173 1
113 0174 1 SIDE EFFECTS:
114 0175 1
115 0176 1     Resolves any lazy lookahead in progress, possibly opening the file.
116 0177 1
117 0178 1 SIGNALLED ERRORS:
118 0179 1
119 0180 1     FILNOTOPE - file not open
120 0181 1
121 0182 1 --
122 0183 1
123 0184 2 BEGIN
124 0185 2
125 0186 2 LOCAL
126 0187 2     FCB: REF $PASS$.CB_CONTROL_BLOCK, ! Control block
127 0188 2     RESULT, ! Function result
128 0189 2     PFV_ADDR: VOLATILE, ! Enable argument
129 0190 2     UNWIND_ACT: VOLATILE; ! Enable argument
130 0191 2
131 0192 2 ENABLE
132 0193 2     PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT);
133 0194 2
134 0195 2 ++
135 0196 2     Set the enable argument for the PFV address.
136 0197 2

```

137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184

```

0198 2
0199 2 PFV_ADDR = PFV [PFV$R_PFV];
0200 2
0201 2  !+
0202 2  ! Validate and lock the PFV.
0203 2  !-
0204 2
0205 2  PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
0206 2
0207 2  !+
0208 2  ! Set unwind action to unlock file.
0209 2  !-
0210 2
0211 2  UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
0212 2
0213 2  !+
0214 2  ! Call PAS$$LOOK_AHEAD to resolve lazy lookahead.
0215 2  !-
0216 2
0217 2  IF NOT .PFV [PFV$V_VALID]
0218 2  THEN
0219 2      PAS$$LOOK_AHEAD (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
0220 2
0221 2  !+
0222 2  ! Verify that file is open.
0223 2  !-
0224 2
0225 2  IF NOT .PFV [PFV$V_OPEN]      ! Not open
0226 2  THEN
0227 2      $PAS$IO_ERROR (PAS$_FILNOTOPE,0);
0228 2
0229 2  !+
0230 2  ! Determine if EOF
0231 2  !-
0232 2
0233 2  RESULT = NOT .PFV [PFV$V_DFB];
0234 2
0235 2  !+
0236 2  ! Unlock file and return.
0237 2  ! Indicate successful completion
0238 2  !-
0239 2
0240 2  FCB [FCB$L_STATUS] = 0;
0241 2  PFV [PFV$V_LOCK] = 0;
0242 2
0243 2  RETURN .RESULT;
0244 2
0245 1  END;

```

```

! End of routine PAS$EOF2

.TITLE PAS$EOF2 Inquire if end-of-file
.IDENT \1-001\

.EXTRN PAS$EOF2, PAS$$IO_HANDLER
.EXTRN PAS$$VALIDATE_PFV
.EXTRN PAS$$LOOK_AHEAD
.EXTRN PAS$$SIGNAL, PAS$K_FILNOTOPE

```

```

.PSECT _PAS$CODE,NOWRT, SHR, PIC,2
.ENTRY PAS$EOF2, Save R2,R3,R6,R7
SUBL2 #4, SP
CLRL UNWIND_ACT
CLRL PFV_ADDR
MOVAL 4$, -(FP)
MOVL PFV, R6
MOVL R6, PFV_ADDR
JSB PAS$$VACIDATE_PFV
MOVL #1, UNWIND_ACT
BLBS 6(R6), 1$
JSB PAS$$LOOK_AHEAD
BBS #5, 7(R6), 2$
CLRL -(SP)
MOVZBL #PAS$K_FILNOTOPE, -(SP)
CALLS #2, PAS$$SIGNAL
BRB 3$
EXTZV #1, #1, 6(R6), RESULT
MCOML RESULT, RESULT
CLRL -44(FCB)
BICB2 #128, 7(R6)
RET
CLRL R0
RET
.WORD Save nothing
MOVL 8(AP), R0
MOVL 4(R0), R0
PUSHAB UNWIND_ACT
PUSHAB PFV_ADDR
PUSHL #2
PUSHL SP
MOVQ 4(AP), -(SP)
CALLS #3, PAS$$IO_HANDLER
RET

```

: Routine Size: 115 bytes, Routine Base: _PAS\$CODE + 0000

```

: 185 0246 1
: 186 0247 1 !<BLF/PAGE>
: 187 0248 1 END
: 188 0249 1
: 189 0250 0 ELUDOM

```

! End of module PAS\$EOF2

PSECT SUMMARY

```

: Name Bytes Attributes
: _PAS$CODE 115 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

```

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	0	0	581	00:01.0
\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	90	21	33	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASEOF2/OBJ=OBJ\$:PASEOF2 MSRC\$:PASEOF2/UPDATE=(ENH\$:PASEOF2)

```

: Size:          115 code + 0 data bytes
: Run Time:      00:04.5
: Elapsed Time: 00:21.2
: Lines/CPU Min: 3333
: Lexemes/CPU-Min: 9320
: Memory Used:  64 pages
: Compilation Complete

```

