


```

PPPPPPPP      AAAAAA      SSSSSSSS      EEEEEEEEEEE      000000      LL      NN      NN      222222
PPPPPPPP      AAAAAA      SSSSSSSS      EEEEEEEEEEE      000000      LL      NN      NN      222222
PP      PP      AA      AA      SS      EE      00      00      LL      NN      NN      22      22
PP      PP      AA      AA      SS      EE      00      00      LL      NN      NN      22      22
PP      PP      AA      AA      SS      EE      00      00      LL      NNNN      NN      22      22
PP      PP      AA      AA      SS      EE      00      00      LL      NNNN      NN      22      22
PPPPPPPP      AA      AA      SSSSSS      EEEEEEEEEEE      00      00      LL      NN      NN      22      22
PPPPPPPP      AA      AA      SSSSSS      EEEEEEEEEEE      00      00      LL      NN      NN      22      22
PP      AAAAAAAAAA      SS      EE      00      00      LL      NN      NNNN      22      22
PP      AAAAAAAAAA      SS      EE      00      00      LL      NN      NNNN      22      22
PP      AA      AA      SS      EE      00      00      LL      NN      NN      22      22
PP      AA      AA      SS      EE      00      00      LL      NN      NN      22      22
PP      AA      AA      SSSSSSSS      EEEEEEEEEEE      000000      LLLLLLLLLL      NN      NN      2222222222      ....
PP      AA      AA      SSSSSSSS      EEEEEEEEEEE      000000      LLLLLLLLLL      NN      NN      2222222222      ....

```

```

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
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS

```

```

1 0001 0 MODULE PASSEOLN2 ( %TITLE 'Inquire if end-of-line'
2 0002 0 IDENT = '1-001' ! File: PASEOLN2.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 EOLN 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 --
47 0047 1

```

```

: 49      0048 1 %SBTTL 'Declarations'
: 50      0049 1
: 51      0050 1 : PROLOGUE DEFINITIONS:
: 52      0051 1 :
: 53      0052 1
: 54      0053 1 REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures
: 55      0117 1
: 56      0118 1
: 57      0119 1 :
: 58      0120 1 : TABLE OF CONTENTS:
: 59      0121 1 :
: 60      0122 1
: 61      0123 1 FORWARD ROUTINE
: 62      0124 1 PAS$EOLN2;                          ! Inquire if end-of-line
: 63      0125 1
: 64      0126 1 :
: 65      0127 1 : MACROS:
: 66      0128 1
: 67      0129 1 : NONE
: 68      0130 1
: 69      0131 1 : EQUATED SYMBOLS:
: 70      0132 1
: 71      0133 1 : NONE
: 72      0134 1
: 73      0135 1 : FIELDS:
: 74      0136 1
: 75      0137 1 : NONE
: 76      0138 1
: 77      0139 1 : OWN STORAGE:
: 78      0140 1
: 79      0141 1 : NONE

```

PA
Sy
PA
PA
PA
PA
PS
--
_P
Ph
--
In
Co
Pa
Pa
Sy
Pa
Sy
Ps
Cr
As
Th
24
Th
23
0
Ma
--
_S
0
Th
MA

```

: 81 0142 1 %SBTTL 'PASSEOLN2 - Inquire if end-of-file'
: 82 0143 1 GLOBAL ROUTINE PASSEOLN2 (           ! Inquire if end-of-file
: 83 0144 1     PFV: REF $PASS$PFV_FILE_VARIABLE ! File variable
: 84 0145 1     ) =
: 85 0146 1
: 86 0147 1 !++
: 87 0148 1 ! FUNCTIONAL DESCRIPTION:
: 88 0149 1
: 89 0150 1     This procedure returns a boolean value TRUE if the given file
: 90 0151 1     is at end-of-line, FALSE otherwise.
: 91 0152 1
: 92 0153 1 ! CALLING SEQUENCE:
: 93 0154 1
: 94 0155 1     Eoln.wv.v = PASSEOLN2 (PFV.mr.r)
: 95 0156 1
: 96 0157 1 ! FORMAL PARAMETERS:
: 97 0158 1
: 98 0159 1     PFV           - The Pascal File Variable (PFV) passed by reference.
: 99 0160 1                   The structure of the PFV is defined in PASPFV.REQ.
100 0161 1
101 0162 1 ! IMPLICIT INPUTS:
102 0163 1
103 0164 1     NONE
104 0165 1
105 0166 1 ! IMPLICIT OUTPUTS:
106 0167 1
107 0168 1     NONE
108 0169 1
109 0170 1 ! ROUTINE VALUE:
110 0171 1
111 0172 1     Boolean TRUE (1) if file is at end-of-file.
112 0173 1     Boolean FALSE (0) if file is not at end-of-file.
113 0174 1
114 0175 1 ! SIDE EFFECTS:
115 0176 1
116 0177 1     Resolves any lazy lookahead in progress, possibly opening the file.
117 0178 1
118 0179 1 ! SIGNALLED ERRORS:
119 0180 1
120 0181 1     FILNOTOPE - file not open
121 0182 1     FILNOTTEX - file is not a textfile
122 0183 1
123 0184 1 !--
124 0185 1
125 0186 2     BEGIN
126 0187 2
127 0188 2     LOCAL
128 0189 2         FCB: REF $PASS$FCB_CONTROL_BLOCK, ! File control block
129 0190 2         RESULT, ! Function result
130 0191 2         PFV_ADDR: VOLATILE, ! Enable argument
131 0192 2         UNWIND_ACT: VOLATILE; ! Enable argument
132 0193 2
133 0194 2     ENABLE
134 0195 2         PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT);
135 0196 2
136 0197 2     !+
137 0198 2     ! Set enable argument for PFV address.

```

```

: 138 0199 2 !-
: 139 0200
: 140 0201 PFV_ADDR = PFV [PFV$R_PFV]; ! Set enable argument
: 141 0202
: 142 0203 !+
: 143 0204 ! Validate and lock PFV.
: 144 0205 !-
: 145 0206
: 146 0207 PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
: 147 0208
: 148 0209 !+
: 149 0210 ! Set unwind action to unlock file.
: 150 0211 !-
: 151 0212
: 152 0213 UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
: 153 0214
: 154 0215 !+
: 155 0216 ! Call PAS$$LOOK_AHEAD to resolve lazy lookahead.
: 156 0217 !-
: 157 0218
: 158 0219 IF NOT .PFV [PFV$V_VALID]
: 159 0220 THEN
: 160 0221 PAS$$LOOK_AHEAD (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
: 161 0222
: 162 0223 !+
: 163 0224 ! Verify that file is open.
: 164 0225 !-
: 165 0226
: 166 0227 IF NOT .PFV [PFV$V_OPEN] ! Not open
: 167 0228 THEN
: 168 0229 $PASSIO_ERROR (PAS$_FILNOTOPE,0);
: 169 0230
: 170 0231 !+
: 171 0232 ! Verify that this is a textfile
: 172 0233 !-
: 173 0234
: 174 0235 IF NOT .FCB [FCB$V_TEXT]
: 175 0236 THEN
: 176 0237 $PASSIO_ERROR (PAS$_FILNOTTEX,0);
: 177 0238
: 178 0239 !+
: 179 0240 ! Determine if EOLN
: 180 0241 !-
: 181 0242
: 182 0243 RESULT = .PFV [PFV$V_EOLN];
: 183 0244
: 184 0245 !+
: 185 0246 ! Indicate successful completion
: 186 0247 ! Unlock file and return.
: 187 0248 !-
: 188 0249
: 189 0250 FCB [FCB$L_STATUS] = 0;
: 190 0251 PFV [PFV$V_LOCK] = 0;
: 191 0252
: 192 0253 RETURN .RESULT;
: 193 0254
: 194 0255 END;

```

! End of routine PAS\$EOLN2

.TITLE PASSEOLN2 Inquire if end-of-line
.IDENT \1-001\

.EXTRN PASSEOLN2, PASS\$IO_HANDLER
.EXTRN PASS\$VALIDATE_PFV
.EXTRN PASS\$LOOK_AHEAD
.EXTRN PASS\$SIGNAL, PASSK_FILNOTOPE
.EXTRN PASSK_FILNOTTEX

.PSECT _PASSCODE, NOWRT, SHR, PIC, 2

.ENTRY PASSEOLN2, Save R2,R3,R6,R7

SUBL2 #4, SP
CLRL UNWIND ACT
CLRL PFV_ADDR
MOVAL 6\$, -(FP)
MOVL PFV, R6
MOVL R6, PFV_ADDR
JSB PASS\$VALIDATE_PFV
MOVL #1, UNWIND_ACT
BLBS 6(R6), 1\$
JSB PASS\$LOOK_AHEAD
BBS #5, 7(R6), 2\$
CLRL -(SP)
MOVZBL #PASSK_FILNOTOPE, -(SP)
BRB 3\$
BLBS -8(FCB), 4\$
CLRL -(SP)
MOVZBL #PASSK_FILNOTTEX, -(SP)
CALLS #2, PASS\$SIGNAL
BRB 5\$
EXTZV #3, #1, 6(R6), RESULT
CLRL -4(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, PASS\$IO_HANDLER
RET

: 0143
: 0186
: 0201
: 0207
: 0213
: 0219
: 0221
: 0227
: 0229
: 0235
: 0237
: 0243
: 0250
: 0251
: 0253
: 0255
: 0186

```

                                00CC 00000
                                04 C2 00002
5E                                7E D4 00005
                                04 AE D4 00007
                                04 AE D4 00007
6D                                04 AE D4 00007
004E CF DE 0000A
56                                04 AC D0 0000F
04 AE 56 D0 00013
00000000G 00 16 00017
6E 01 D0 0001D
06 06 A6 E8 00020
00000000G 00 16 00024
08 07 A6 05 E0 0002A 1$:
                                7E D4 0002F
                                00G 8F 9A 00031
                                0A 11 00035
0F F8 A7 E8 00037 2$:
                                7E D4 0003B
                                00G 8F 9A 0003D
00000000G 00 02 FB 00041 3$:
                                0F 11 00048
50 06 A6 01 03 EF 0004A 4$:
                                D4 A7 D4 00050
                                07 A6 80 8F 8A 00053
                                04 00058
                                50 D4 00059 5$:
                                04 0005B
                                0000 0005C 6$:
50 08 AC D0 0005E
50 04 A0 D0 00062
                                F8 A0 9F 00066
                                FC A0 9F 00069
                                02 DD 0006C
                                5E DD 0006E
                                04 AC 7D 00070
00000000G 7E 00 03 FB 00074
                                04 0007B
```

; Routine Size: 124 bytes, Routine Base: _PASSCODE + 0000

: 195 0256 1
: 196 0257 1 !<BLF/PAGE>

PAS\$EOLN2
1-001

Inquire if end-of-line
PAS\$EOLN2 - Inquire if end-of-file

J 5
16-Sep-1984 01:31:43
14-Sep-1984 12:51:26

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASEOLN2.B32;1

Page 6
(4)

: 198 0258 1 END
: 199 0259 1
: 200 0260 0 ELUDOM

! End of module PAS\$EOLN2

PSECT SUMMARY

Name	Bytes	Attributes
_PAS\$CODE	124	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	00:00.9
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	91	21	33	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASEOLN2/OBJ=OBJ\$:PASEOLN2 MSRC\$:PASEOLN2/UPDATE=(ENH\$:PASEOLN2)

: Size: 124 code + 0 data bytes
: Run Time: 00:04.8
: Elapsed Time: 00:20.1
: Lines/CPU Min: 3277
: Lexemes/CPU-Min: 10966
: Memory Used: 69 pages
: Compilation Complete

