

```

PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTTTT      LLL
PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTTTT      LLL
PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTTTT      LLL
PPP                PPP      AAA      AAA      SSS      RRR      RRR      TTT      LLL
PPP                PPP      AAA      AAA      SSS      RRR      RRR      TTT      LLL
PPP                PPP      AAA      AAA      SSS      RRR      RRR      TTT      LLL
PPP                PPP      AAA      AAA      SSS      RRR      RRR      TTT      LLL
PPP                PPP      AAA      AAA      SSS      RRR      RRR      TTT      LLL
PPP                PPP      AAA      AAA      SSS      RRR      RRR      TTT      LLL
PPPPPPPPPPPP      AAA      AAA      SSSSSSSSSS      RRRRRRRRRRRR      TTT      LLL
PPPPPPPPPPPP      AAA      AAA      SSSSSSSSSS      RRRRRRRRRRRR      TTT      LLL
PPPPPPPPPPPP      AAA      AAA      SSSSSSSSSS      RRRRRRRRRRRR      TTT      LLL
PPP                AAAAAAAAAAAAAAAAAA      SSS      RRR      RRR      TTT      LLL
PPP                AAAAAAAAAAAAAAAAAA      SSS      RRR      RRR      TTT      LLL
PPP                AAAAAAAAAAAAAAAAAA      SSS      RRR      RRR      TTT      LLL
PPP                AAA      AAA      SSS      RRR      RRR      TTT      LLL
PPP                AAA      AAA      SSS      RRR      RRR      TTT      LLL
PPP                AAA      AAA      SSS      RRR      RRR      TTT      LLL
PPP                AAA      AAA      SSSSSSSSSS      RRR      RRR      TTT      LLLLLLLLLLLLLLLLLL
PPP                AAA      AAA      SSSSSSSSSS      RRR      RRR      TTT      LLLLLLLLLLLLLLLLLL
PPP                AAA      AAA      SSSSSSSSSS      RRR      RRR      TTT      LLLLLLLLLLLLLLLLLL

```



```

1 0001 0 MODULE PASSWRITELN2 ( %TITLE 'WRITELN procedure'
2 0002 0 IDENT = '1-002' ! File: PASWRITEL.B32 Edit: SBL1002
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 module contains PASSWRITELN2, which implements the
36 0036 1 VAX-11 Pascal WRITELN procedure.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SBL 1-April-1981
45 0045 1 1-002 - Add NOWRITE attribute to PFD_WRITEV. SBL 24-May-1982
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';           ! Linkages, PSECTs, externals
55 0117 1 |
56 0118 1 |
57 0119 1 | TABLE OF CONTENTS:
58 0120 1 |
59 0121 1 |
60 0122 1 FORWARD ROUTINE
61 0123 1     PASSWRITELN2: NOVALUE,           ! Do a WRITELN
62 0124 1     PASS$WRITELN: JSB WRITELN NOVALUE, ! Internally callable
63 0125 1     PASS$INIT_WRITE: JSB INIT_WRITE NOVALUE, ! Do common initialization
64 0126 1     PASS$END_WRITE: JSB_END_WRITE NOVAI E, ! Do common WRITE epilogue
65 0127 1     PASS$DO_WRITEV: JSB_DO_WRITEV NOVALUE; ! Do a WRITEV
66 0128 1 |
67 0129 1 |
68 0130 1 | EQUATED SYMBOLS:
69 0131 1 |
70 0132 1 |     NONE
71 0133 1 |
72 0134 1 | FIELDS:
73 0135 1 |
74 0136 1 |     NONE
75 0137 1 |
76 0138 1 | OWN STORAGE:
77 0139 1 |
78 0140 1 |
79 0141 1 | +
80 0142 1 | Declare a Pascal File Descriptor to use for a "string file" WRITEV
81 0143 1 | -
82 0144 1 |
83 0145 1 OWN
84 0146 1     PFD_WRITEV: BLOCK [PFD$K_SIZE+%CHARCOUNT('-string-'), BYTE]
85 0147 1     _FIELD (PFD$FIELDS) PSECT (_PASS$CODE)
86 0148 1     PRESET (
87 0149 1         [PFD$V_TEXT] = 1,
88 0150 1         [PFD$V_NOREAD] = 1,
89 0151 1         [PFD$V_NOWRITE] = 1,
90 0152 1         [PFD$L_LENGTH] = 1,
91 0153 1         [PFD$T_NAME] = %CHARCOUNT('-string-'),
92 0154 1         [PFD$B_NAME1] = %C'-',
93 0155 1         [PFD$B_NAME2] = %C's',
94 0156 1         [PFD$B_NAME3] = %C't',
95 0157 1         [PFD$B_NAME4] = %C'r',
96 0158 1         [PFD$B_NAME5] = %C'i',
97 0159 1         [PFD$B_NAME6] = %C'n',
98 0160 1         [PFD$B_NAME7] = %C'g',
99 0161 1         [PFD$B_NAME8] = %C'-',
100 0162 1     );
```

```

: 102 0163 1 %SBTTL 'PASSWRITELN2 - WRITELN procedure'
: 103 0164 1 GLOBAL ROUTINE PASSWRITELN2 (           ! Do a WRITELN
: 104 0165 1     PFV: REF $PASSPFV_FILE_VARIABLE,     ! File variable
: 105 0166 1     ERROR                               ! Error unwind address
: 106 0167 1     ): NOVALUE =
: 107 0168 1
: 108 0169 1 !++
: 109 0170 1 ! FUNCTIONAL DESCRIPTION:
: 110 0171 1
: 111 0172 1     PASSWRITELN2 implements the VAX-11 Pascal WRITELN procedure.  It
: 112 0173 1     writes the current record buffer of a textfile to the file.
: 113 0174 1
: 114 0175 1 ! CALLING SEQUENCE:
: 115 0176 1
: 116 0177 1     CALL PASSWRITELN2 (PFV.mr.r [, ERROR.j.r])
: 117 0178 1
: 118 0179 1 ! FORMAL PARAMETERS:
: 119 0180 1
: 120 0181 1     PFV                               - The Pascal File Variable (PFV) passed by reference.
: 121 0182 1     The structure of the PFV is defined in PASPFV.REQ.
: 122 0183 1
: 123 0184 1     ERROR                             - Optional.  If specified, the address to unwind to
: 124 0185 1     in case of an error.
: 125 0186 1
: 126 0187 1 ! IMPLICIT INPUTS:
: 127 0188 1
: 128 0189 1     NONE
: 129 0190 1
: 130 0191 1 ! IMPLICIT OUTPUTS:
: 131 0192 1
: 132 0193 1     NONE
: 133 0194 1
: 134 0195 1 ! ROUTINE VALUE:
: 135 0196 1
: 136 0197 1     NONE
: 137 0198 1
: 138 0199 1 ! SIDE EFFECTS:
: 139 0200 1
: 140 0201 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
: 141 0202 1
: 142 0203 1 ! SIGNALLED ERRORS:
: 143 0204 1
: 144 0205 1
: 145 0206 1 !--
: 146 0207 1
: 147 0208 2     BEGIN
: 148 0209 2
: 149 0210 2     LOCAL
: 150 0211 2         FCB: REF $PASSFCB_CONTROL_BLOCK,           ! File control block
: 151 0212 2         PFV_ADDR: VOLATILE,                       ! Enable argument
: 152 0213 2         UNWIND_ACT: VOLATILE,                     ! Enable argument
: 153 0214 2         ERROR_ADDR: VOLATILE;                     ! Enable argument
: 154 0215 2
: 155 0216 2
: 156 0217 2     BUILTIN
: 157 0218 2         ACTUALCOUNT;
: 158 0219 2

```

```

159 0220 2  ENABLE
160 0221 2  PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);           ! Enable error handler
161 0222 2
162 0223 2  IF ACTUALCOUNT () GEQU 2
163 0224 2  THEN
164 0225 2  ERROR_ADDR = .ERROR;                                       ! Set unwind address
165 0226 2
166 0227 2  PFV_ADDR = PFV [PFV$R_PFV];                                  ! Set PFV address
167 0228 2
168 0229 2  !+
169 0230 2  ! Validate PFV and get PFV.
170 0231 2  !-
171 0232 2
172 0233 2  PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
173 0234 2
174 0235 2  !+
175 0236 2  ! Set unwind action to unlock file.
176 0237 2  !-
177 0238 2
178 0239 2  UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
179 0240 2
180 0241 2  !+
181 0242 2  ! Do common initialization.
182 0243 2  !-
183 0244 2
184 0245 2  PASS$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
185 0246 2
186 0247 2  !+
187 0248 2  ! Call PASS$WRITELN to do the work.
188 0249 2  !-
189 0250 2
190 0251 2  PASS$WRITELN (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
191 0252 2
192 0253 2  !+
193 0254 2  ! Indicate successful completion
194 0255 2  ! Undefine the file buffer
195 0256 2  ! Unlock the file variable.
196 0257 2  !-
197 0258 2
198 0259 2  FCB [FCB$L_STATUS] = 0;
199 0260 2  PFV [PFV$V_DFB] = 0;
200 0261 2  PFV [PFV$V_LOCK] = 0;
201 0262 2
202 0263 2  RETURN;
203 0264 2
204 0265 1  END;

```

! End of routine PASSWRITELN2

```

.TITLE PASSWRITELN2 WRITELN procedure
.IDENT \1-002\
.PSECT _PASS$CODE,NOWRT, SHR, PIC,2

```

```

00# 0000 PFD_WRITEV:
19 00004 .BYTE 0[4]
00# 00005 .BYTE 25
        .BYTE 0[3]

```

2D 67 6E 69 72 74 73 00000001 00008  
2D 08 0000C

```
.LONG 1
.BYTE 8, 45, 115, 116, 114, 105, 110, 103, 45
.EXTRN PASSWRITELN2, PASS$WRITELN
.EXTRN PASS$INIT_WRITE
.EXTRN PASS$END_WRITE, PASS$DO_WRITEV
.EXTRN PASS$IO_HANDLER
.EXTRN PASS$VACIDATE_PFV
```

```
00FC 00000
5E 08 C2 00002
7E D4 00005
04 AE 7C 00007
6D 002C CF DE 0000A
02 6C 91 0000F
04 1F 00012
6E 08 AC D0 00014
56 04 AC D0 00018 1$:
08 AE 56 D0 0001C
04 AE 00000000G 00 16 00020
01 D0 00026
0000V 30 0002A
0000V 30 0002D
D4 A7 D4 00030
06 A6 8002 8F AA 00033
04 00039
0000 0003A 2$:
50 08 AC D0 0003C
50 04 A0 D0 00040
F4 A0 9F 00044
F8 A0 9F 00047
FC A0 9F 0004A
03 DD 0004D
5E DD 0004F
00000000G 7E 04 AC 7D 00051
00 03 FB 00055
04 0005C
```

```
.ENTRY PASSWRITELN2, Save R2,R3,R4,R5,R6,R7 : 0164
SUBL2 #8, SP :
CLRL ERROR_ADDR : 0208
CLRQ UNWIND_ACT
MOVAL 2$, (FP)
CMPB (AP), #2 : 0223
BLSSU 1$
MOVL ERROR, ERROR_ADDR : 0225
MOVL PFV, R6 : 0227
MOVL R6, PFV_ADDR
JSB PASS$VACIDATE_PFV : 0233
MOVL #1, UNWIND_ACT : 0239
BSBW PASS$INIT_WRITE : 0245
BSBW PASS$WRITELN : 0251
CLRL -44(FCB) : 0259
BICW2 #32770, 6(R6) : 0261
RET : 0265
.WORD Save nothing : 0208
MOVL 8(AP), R0
MOVL 4(R0), R0
PUSHAB ERROR_ADDR
PUSHAB UNWIND_ACT
PUSHAB PFV_ADDR
PUSHL #3
PUSHL SP
MOVQ 4(AP), --(SP)
CALLS #3, PASS$IO_HANDLER
RET
```

; Routine Size: 93 bytes, Routine Base: \_PASS\$CODE + 0015

```
: 205 0266 1
: 206 0267 1 !<BLF/PAGE>
```

```

: 208 0268 1 %SBITL 'PASS$WRITELN - Internally callable WRITELN '
: 209 0269 1 GLOBAL ROUTINE PASS$WRITELN (
: 210 0270 1     PFV: REF $PASS$PFV_FILE_VARIABLE,      ! File variable
: 211 0271 1     FCB: REF $PASS$FCB_CONTROL_BLOCK  ! File control block
: 212 0272 1 ): JSB_WRITELN NOVALUE =
: 213 0273 1
: 214 0274 1 !++
: 215 0275 1 FUNCTIONAL DESCRIPTION:
: 216 0276 1
: 217 0277 1     PASS$WRITELN implements the VAX-11 Pascal WRITELN procedure. It
: 218 0278 1     writes the current record buffer of a textfile to the file.
: 219 0279 1
: 220 0280 1 CALLING SEQUENCE:
: 221 0281 1
: 222 0282 1     JSB_WRITELN PASS$WRITELN (PFV.mr.r, FCB.mr.r)
: 223 0283 1
: 224 0284 1 FORMAL PARAMETERS:
: 225 0285 1
: 226 0286 1     PFV           - The Pascal File Variable (PFV) passed by reference.
: 227 0287 1                   The structure of the PFV is defined in PASPFV.REQ.
: 228 0288 1
: 229 0289 1     FCB           - The File Control Block for the file.
: 230 0290 1
: 231 0291 1 IMPLICIT INPUTS:
: 232 0292 1
: 233 0293 1     It is assumed that the file has been verified to be a textfile and
: 234 0294 1     that it is locked.
: 235 0295 1
: 236 0296 1 IMPLICIT OUTPUTS:
: 237 0297 1
: 238 0298 1     NONE
: 239 0299 1
: 240 0300 1 ROUTINE VALUE:
: 241 0301 1
: 242 0302 1     NONE
: 243 0303 1
: 244 0304 1 SIDE EFFECTS:
: 245 0305 1
: 246 0306 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
: 247 0307 1
: 248 0308 1 SIGNALLED ERRORS:
: 249 0309 1
: 250 0310 1     LINVALEXC - LINELIMIT value exceeded
: 251 0311 1     ERRDURWRI - error during WRITELN
: 252 0312 1
: 253 0313 1 !--
: 254 0314 1
: 255 0315 2 BEGIN
: 256 0316 2
: 257 0317 2 BIND
: 258 0318 2     RAB = FCB: REF $PASS$FCB_CONTROL_BLOCK;      ! RAB is also FCB address
: 259 0319 2
: 260 0320 2 !+
: 261 0321 2     Check for linelimit exceeded and update remaining line count.
: 262 0322 2     !-
: 263 0323 2
: 264 0324 2     IF .FCB [FCB$L_LINELIMIT] GEQ 0      ! Linelimit enabled?

```



```

265 0325 2 THEN
266 0326 IF .FCB [FCBSL_LINELIMIT] EQL 0 ! Linelimit exceeded?
267 0327 THEN
268 0328 $PASSIO_ERROR (PASS_LINVALEXC,0)
269 0329 ELSE
270 0330 FCB [FCBSL_LINELIMIT] = .FCB [FCBSL_LINELIMIT] - 1;
271 0331
272 0332 !+
273 0333 ! Set up record pointer in RAB for $PUT.
274 0334 !-
275 0335
276 0336 RAB [RAB$L_RBF] = .FCB [FCBSA_RECORD_BEG];
277 0337 RAB [RAB$W_RSZ] = .FCB [FCBSA_RECORD_CUR] - .FCB [FCBSA_RECORD_BEG];
278 0338
279 0339 !+
280 0340 ! Reset record buffer.
281 0341 !-
282 0342
283 0343 FCB [FCBSA_RECORD_CUR] = .FCB [FCBSA_RECORD_BEG];
284 0344
285 0345 !+
286 0346 ! If prompting enabled, set carriage control appropriately.
287 0347 !-
288 0348
289 0349 IF .FCB [FCBSV_PROMPT_ENABLE]
290 0350 THEN
291 0351 IF .FCB [FCBSV_PARTIAL_LINE]
292 0352 THEN
293 0353 FCB [FCBSW_PROMPT_CC] = FCB$K_CC_NLCR ! Nothing before, CR after
294 0354 ELSE
295 0355 FCB [FCBSW_PROMPT_CC] = FCB$K_CC_LFCR; ! LF before, CR after
296 0356
297 0357 !+
298 0358 ! Do the $PUT and check for errors.
299 0359 !-
300 0360
301 0361 IF NOT $PASSRMS_OP ($PUT (RAB=.RAB))
302 0362 THEN
303 0363 $PASSIO_ERROR (PASS_ERRDURWRI);
304 0364
305 0365 !+
306 0366 ! Cancel the "truncate-on-put" option which may have been set by
307 0367 ! a REWRITE.
308 0368 !-
309 0369
310 0370 RAB [RAB$V_TPT] = 0;
311 0371
312 0372 !+
313 0373 ! Indicate no partial line.
314 0374 !-
315 0375
316 0376 FCB [FCBSV_PARTIAL_LINE] = 0;
317 0377
318 0378 RETURN;
319 0379
320 0380 ! End of routine PASS$WRITELN

```



```

: 324 0383 1 %SBTTL 'PASS$INIT_WRITE - Common setup for WRITE procedures'
: 325 0384 1 GLOBAL ROUTINE PASS$INIT_WRITE (                               ! Common setup
: 326 0385 1     PFV: REF $PASS$PFV FILE VARIABLE,                          ! File variable
: 327 0386 1     FCB_IN: REF $PASS$FCB CONTROL_BLOCK;                       ! Input FCB
: 328 0387 1     FCB: REF $PASS$FCB CONTROL_BLOCK                          ! Output FCB
: 329 0388 1 ): JSB_INIT_WRITE NOVALUE =
: 330 0389 1
: 331 0390 1 +-+
: 332 0391 1 FUNCTIONAL DESCRIPTION:
: 333 0392 1
: 334 0393 1     This procedure performs the common setup steps for all WRITE procedures.
: 335 0394 1     These are:
: 336 0395 1         Verify that the file is a textfile.
: 337 0396 1         If it is the standard file OUTPUT or INPUT, open it.
: 338 0397 1         Verify that the file is in Generation mode.
: 339 0398 1         Resolve any lazy lookahead in progress.
: 340 0399 1
: 341 0400 1 CALLING SEQUENCE:
: 342 0401 1
: 343 0402 1     JSB_INIT_WRITE PASS$INIT_WRITE (PFV.mr.r, FCB_IN.mr.r; FCB))
: 344 0403 1
: 345 0404 1 FORMAL PARAMETERS:
: 346 0405 1
: 347 0406 1     PFV           - The Pascal File Variable (PFV) passed by reference in R2.
: 348 0407 1                 The structure of the PFV is defined in PASPFV.REQ.
: 349 0408 1
: 350 0409 1     FCB_IN      - The File Control Block of the file, if any. The
: 351 0410 1                 resultant FCB is passed back in the FCB parameter.
: 352 0411 1
: 353 0412 1     FCB         - The address of the File Control Block (FCB) is an
: 354 0413 1                 output register parameter.
: 355 0414 1
: 356 0415 1 IMPLICIT INPUTS:
: 357 0416 1
: 358 0417 1     NONE
: 359 0418 1
: 360 0419 1 IMPLICIT OUTPUTS:
: 361 0420 1
: 362 0421 1     NONE
: 363 0422 1
: 364 0423 1 ROUTINE VALUE:
: 365 0424 1
: 366 0425 1     NONE
: 367 0426 1
: 368 0427 1 SIDE EFFECTS:
: 369 0428 1
: 370 0429 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
: 371 0430 1
: 372 0431 1 SIGNALLED ERRORS:
: 373 0432 1
: 374 0433 1     FILNOTTEX - file is not a textfile
: 375 0434 1     FILNOTOPE - File not open
: 376 0435 1     FILNOTGEN - file is not in Generation mode
: 377 0436 1
: 378 0437 1 --
: 379 0438 1
: 380 0439 2     BEGIN

```

```

: 381 0440 2
: 382 0441 2
: 383 0442 2
: 384 0443 2
: 385 0444 2
: 386 0445 2
: 387 0446 2
: 388 0447 2
: 389 0448 2
: 390 0449 2
: 391 0450 2
: 392 0451 2
: 393 0452 2
: 394 0453 2
: 395 0454 2
: 396 0455 2
: 397 0456 2
: 398 0457 2
: 399 0458 2
: 400 0459 2
: 401 0460 2
: 402 0461 2
: 403 0462 2
: 404 0463 2
: 405 0464 2
: 406 0465 2
: 407 0466 2
: 408 0467 2
: 409 0468 2
: 410 0469 2
: 411 0470 2
: 412 0471 2
: 413 0472 2
: 414 0473 2
: 415 0474 2
: 416 0475 1

!+
!- Resolve lazy lookahead, if any.
!-
IF NOT .PFV [PFV$V_VALID]
THEN
    PASS$LOOK_AHEAD (PFV [PFV$R_PFV], FCB_IN [FCB$R_FCB]; FCB);

!+
!- Verify that the file is open.
!-
IF NOT .PFV [PFV$V_OPEN]          ! Not open
THEN
    $PASSIO_ERROR (PASS$_FILNOTOPE,0);

!+
!- Verify that this is a textfile
!-
IF NOT .FCB [FCB$V_TEXT]
THEN
    $PASSIO_ERROR (PASS$_FILNOTTEX,0);

!+
!- Test for Generation mode
!-
IF NOT .FCB [FCB$V_GENERATION]
THEN
    $PASSIO_ERROR (PASS$_FILNOTGEN,0);

RETURN;          ! Success

END;

! End of routine PASS$INIT_WRITE

```

```

.EXTRN PASS$LOOK_AHEAD
.EXTRN PASS$_FILNOTOPE
.EXTRN PASS$_FILNOTTEX
.EXTRN PASS$_FILNOTGEN

```

	06	06	A6	E8	00000	PASS\$INIT_WRITE::		
				B[BS	6(PFV), 1\$			: 0445
		00000000G	00	16	00004	JSB	PASS\$LOOK_AHEAD	: 0447
08	07	A6	05	E0	0000A 1\$:	BBS	#5, 7(PFV), 2\$	: 0453
			7E	D4	0000F	CLRL	-(SP)	: 0455
		7E	00G	8F	9A 00011	MOVZBL	#PASS\$_FILNOTOPE, -(SP)	
			17	11	00015	BRB	4\$	
		08	F8	A7	E8 00017 2\$:	BLBS	-8(FCB), 3\$	: 0461
			7E	D4	0001B	CLRL	-(SP)	: 0463
		7E	00G	8F	9A 0001D	MOVZBL	#PASS\$_FILNOTTEX, -(SP)	
			0B	11	00021	BRB	4\$	
0D	FD	A7	04	E0	00023 3\$:	BBS	#4, -3(FCB), 5\$	: 0469
			7E	D4	00028	CLRL	-(SP)	: 0471

PASSWRITELN2  
1-002

WRITELN procedure  
PASS\$INIT\_WRITE - Common setup for WRITE proced

N 11  
16-Sep-1984 02:26:26  
14-Sep-1984 12:52:09

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

00000000G	7E	00G	8F	9A	0002A	MOVZBL	#PASSK	FILNOTGEN,	-(SP)
	00		02	FB	0002E	CALLS	#2,	PASS\$SIGNAL	
				05	00035	RSB			

:  
:  
: 0475

: Routine Size: 54 bytes, Routine Base: \_PASS\$CODE + 00E5

: 417 0476 1  
: 418 0477 1 !<BLF/PAGE>

```

420 0478 1 %SBTTL 'PASS$END WRITE - Common epilogue for WRITE procedures'
421 0479 1 GLOBAL ROUTINE PASS$END_WRITE (
422 0480 1     PFV: REF $PASS$PFV_FILE_VARIABLE,           ! File variable
423 0481 1     FCB: REF $PASS$FCB_CONTROL_BLOCK       ! File control block
424 0482 1 ): JSB_END_WRITE NOVALUE =
425 0483 1
426 0484 1 !++
427 0485 1 ! FUNCTIONAL DESCRIPTION:
428 0486 1
429 0487 1     This procedure is called at the end of every WRITE procedure
430 0488 1     to perform common functions.  These are:
431 0489 1     1. Move last character written to user's file buffer.
432 0490 1     2. Check for PROMPT_ALWAYS and output partial line if
433 0491 1     necessary.
434 0492 1     3. Set STATUS to zero, undefine file buffer and unlock
435 0493 1     file variable.
436 0494 1
437 0495 1 ! CALLING SEQUENCE:
438 0496 1
439 0497 1     JSB_END_WRITE PASS$END_WRITE (PFV.mr.r, FCB.mr.r)
440 0498 1
441 0499 1 ! FORMAL PARAMETERS:
442 0500 1
443 0501 1     PFV           - The Pascal File Variable (PFV) passed by reference in R2.
444 0502 1     The structure of the PFV is defined in PASPFV.REQ.
445 0503 1
446 0504 1     FCB           - The File Control Block of the file.
447 0505 1
448 0506 1 ! IMPLICIT INPUTS:
449 0507 1
450 0508 1     It is assumed that the file is a textfile, is locked and is
451 0509 1     in Generation mode.
452 0510 1
453 0511 1 ! IMPLICIT OUTPUTS:
454 0512 1
455 0513 1     NONE
456 0514 1
457 0515 1 ! ROUTINE VALUE:
458 0516 1
459 0517 1     NONE
460 0518 1
461 0519 1 ! SIDE EFFECTS:
462 0520 1
463 0521 1     NONE
464 0522 1
465 0523 1 ! SIGNALLED ERRORS:
466 0524 1
467 0525 1     NONE
468 0526 1 !--
469 0527 1
470 0528 2     BEGIN
471 0529 2
472 0530 2     !+
473 0531 2     ! If file buffer is not NOREAD, store the last character written to the
474 0532 2     ! line into the user's file buffer.  However, don't do it if there are
475 0533 2     ! no characters in the line!
476 0534 2     !-

```

```

: 477 0535 2
: 478 0536 2 IF (.FCB [FCB$A_RECORD_CUR] NEQA .FCB [FCB$A_RECORD_END]) AND NOT .FCB [FCB$V_NOREAD]
: 479 0537 2 THEN
: 480 0538 2 CH$WCHAR (CH$RCHAR(.FCB [FCB$A_RECORD_CUR]-1), .PFV [PFV$A_BUFFER]);
: 481 0539 2
: 482 0540 2 | +
: 483 0541 2 | If the file has the attribute PROMPT_ALWAYS, write out the partial
: 484 0542 2 | record.
: 485 0543 2 | -
: 486 0544 2
: 487 0545 2 IF .FCB [FCB$V_PROMPT_ALWAYS]
: 488 0546 2 THEN
: 489 0547 2 PASS$PROMPT_FILE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
: 490 0548 2
: 491 0549 2 | +
: 492 0550 2 | Set STATUS to indicate success.
: 493 0551 2 | Unlock file variable.
: 494 0552 2 | -
: 495 0553 2
: 496 0554 2 FCB [FCB$L_STATUS] = 0;
: 497 0555 2 PFV [PFV$V_LOCK] = 0;
: 498 0556 2
: 499 0557 2 RETURN; ! Success
: 500 0558 2
: 501 0559 1 END;

```

! End of routine PASS\$END\_WRITE

.EXTRN PASS\$PROMPT\_FILE

	50	EC	A7	D0	00000	PASS\$END_WRITE::		
						MOVL	-20(FCB), R0	: 0536
	F0	A7		50	D1 00004	CMPL	R0, -16(FCB)	
				0A	13 00008	BEQL	1\$	
05	F8	A7		03	E0 0000A	BBS	#3, -8(FCB), 1\$	
	00	B6		FF	A0 90 0000F	MOVB	-1(R0), a0(PFV)	: 0538
				FE	A7 95 00014	TSTB	-2(FCB)	: 0545
				06	18 00017	BGEQ	2\$	
				00	16 00019	JSB	PASS\$PROMPT_FILE	: 0547
				D4	A7 D4 0001F	CLRL	-44(FCB)	: 0554
	07	A6		80	8F 8A 00022	BICB2	#128, 7(PFV)	: 0555
				05	00027	RSB		: 0559

: Routine Size: 40 bytes, Routine Base: \_PASS\$CODE + 011B

```

: 502 0560 1
: 503 0561 1 !<BLF/PAGE>

```

```

505 0562 1 XSBTTL 'PASS$DO_WRITEV - Do a WRITEV'
506 0563 1 GLOBAL ROUTINE PASS$DO_WRITEV (
507 0564 1     PFV: REF $PASS$PFV_FILE_VARIABLE,           ! PFV of string file
508 0565 1     MAX_LENGTH: WORD,                       ! Maximum length of string
509 0566 1     STRING: REF VECTOR [, WORD],           ! String to write to
510 0567 1     ARGUMENT_LIST: REF VECTOR [, LONG],   ! Argument list
511 0568 1     ROUTINE_ADDR                           ! Routine to call
512 0569 1     ): JSB_DO_WRITEV NOVALUE =
513 0570 1
514 0571 1 ++
515 0572 1 FUNCTIONAL DESCRIPTION:
516 0573 1
517 0574 1 CALLING SEQUENCE:
518 0575 1
519 0576 1     JSB_DO_WRITEV PASS$DO_WRITEV (PFV.mr.r, MAX_LENGTH.rw.v, STRING.wvt.r,
520 0577 1     ARGUMENT_LIST.rlu.r,ROUTINE_ADDR.fzem.r)
521 0578 1
522 0579 1 FORMAL PARAMETERS:
523 0580 1
524 0581 1     PFV                - Pascal File Variable allocated by our caller.
525 0582 1                   It is filled in here.
526 0583 1
527 0584 1     STRING           - The varying string to append to.
528 0585 1
529 0586 1     ARGUMENT_LIST    - Argument list for a CALLG
530 0587 1
531 0588 1     ROUTINE_ADDR     - The address of the PASS$WRITE_xxx routine to call
532 0589 1                   to actually do the work.
533 0590 1
534 0591 1 IMPLICIT INPUTS:
535 0592 1
536 0593 1     NONE
537 0594 1
538 0595 1 IMPLICIT OUTPUTS:
539 0596 1
540 0597 1     NONE
541 0598 1
542 0599 1 ROUTINE VALUE:
543 0600 1
544 0601 1     NONE
545 0602 1
546 0603 1 SIDE EFFECTS:
547 0604 1
548 0605 1     See called routine
549 0606 1
550 0607 1 SIGNALLED ERRORS:
551 0608 1
552 0609 1     NONE
553 0610 1
554 0611 1 --
555 0612 1
556 0613 2 BEGIN
557 0614 2
558 0615 2 LOCAL
559 0616 2     FCB_BLOCK: $PASS$FCB_CONTROL_BLOCK;      ! File Control Block
560 0617 2
561 0618 2 BIND

```



```

562 0619 2          FCB = FCB_BLOCK + FCB$K_BLN: $PASS$FCB_CONTROL_BLOCK;
563 0620 2
564 0621 2      BUILTIN
565 0622 2      CALLG;
566 0623 2
567 0624 2      !+
568 0625 2      ! Set up the Pascal File Variable
569 0626 2      !-
570 0627 2
571 0628 2      PFV [PFV$A_BUFFER] = 0;          ! No file buffer
572 0629 2      PFV [PFV$B_VERSION] = PFV$K_CUR_VERSION; ! PFV version number
573 0630 2      PFV [PFV$W_FLAGS] = 0;          ! Initially zero bit
574 0631 2      PFV [PFV$V_VALID] = 1;         ! PFV is valid
575 0632 2      PFV [PFV$V_FCB_VALID] = 1;    ! FCB address is valid
576 0633 2      PFV [PFV$V_OPEN] = 1;         ! File is "open"
577 0634 2      PFV [PFV$V_EOF_DEFINED] = 1; ! EOF(f) is defined
578 0635 2      PFV [PFV$V_DFB] = 1;          ! File buffer is defined
579 0636 2      PFV [PFV$A_PFD] = PFD_WRITEV [PFD$R_PFD]; ! Address of PFD
580 0637 2      PFV [PFV$A_FCB] = FCB [FCB$R_FCB]; ! Address of FCB
581 0638 2
582 0639 2      !+
583 0640 2      ! Set up the File Control Block (FCB). Only set those fields which
584 0641 2      ! will be used.
585 0642 2      !-
586 0643 2
587 0644 2      FCB [FCB$A_PFV] = PFV [PFV$R_PFV]; ! Address of PFV
588 0645 2      FCB [FCB$A_PFD] = PFD_WRITEV [PFD$R_PFD]; ! Address of name
589 0646 2      FCB [FCB$W_ATTRIB] = PFD_WRITEV [PFD$W_ATTRIB]; ! Attributes
590 0647 2      FCB [FCB$L_OPTIONS1] = 0;      ! Initially 0
591 0648 2      FCB [FCB$V_GENERATION] = 1;   ! In Generation mode
592 0649 2      FCB [FCB$V_STRING] = 1;      ! String file
593 0650 2
594 0651 2      !+
595 0652 2      ! Get the "line" pointer and length, and set the FCB accordingly.
596 0653 2      !-
597 0654 2
598 0655 2      FCB [FCB$A_RECORD_BEG] = STRING [1]; ! Skip over current length
599 0656 2      FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_BEG] + .STRING [0];
600 0657 2      FCB [FCB$L_RECORD_LEN] = .MAX_LENGTH;
601 0658 2      FCB [FCB$A_RECORD_END] = .FCB [FCB$A_RECORD_BEG] + .FCB [FCB$L_RECORD_LEN];
602 0659 2
603 0660 2      !+
604 0661 2      ! Call the appropriate routine;
605 0662 2      !-
606 0663 2
607 0664 2      CALLG (ARGUMENT_LIST [0], .ROUTINE_ADDR);
608 0665 2
609 0666 2      !+
610 0667 2      ! Update the current string length.
611 0668 2      !-
612 0669 2
613 0670 2      STRING [0] = .FCB [FCB$A_RECORD_CUR] - .FCB [FCB$A_RECORD_BEG];
614 0671 2
615 0672 2      RETURN;          ! Success
616 0673 2
617 0674 2      END;          ! End of routine PASS$DO_WRITEV

```

		5E	BC	AE	9E	00000	PASS\$DO_WRITEV::			
				66	D4	00004	MOVAB	-68(SP), SP	: 0563	
		50	04	A6	9E	00006	CLRL	(PFV)	: 0628	
				60	94	0000A	MOVAB	4(PFV), R0	: 0629	
			02	A0	B4	0000C	CLRB	(R0)	: 0630	
02	A0		6007	8F	AB	0000F	CLRW	2(R0)	: 0630	
08	A6		FEA4	CF	9E	00015	BISW2	#24583, 2(R0)	: 0635	
0C	A6		44	AE	9E	0001B	MOVAB	PFD_WRITEV, 8(PFV)	: 0636	
20	AE			56	D0	00020	MOVAB	FCB, 12(PFV)	: 0637	
28	AE		FE95	CF	9E	00024	MOVL	PFV, FCB-36	: 0644	
3C	AE		FE93	CF	B0	0002A	MOVAB	PFD_WRITEV, FCB-28	: 0645	
			40	AE	D4	00030	MOVW	PFD_WRITEV+4, FCB-8	: 0646	
41	AE		1010	8F	AB	00033	CLRL	FCB-4	: 0647	
2C	AE		02	A3	9E	00039	BISW2	#4112, FCB-2	: 0649	
			50	63	3C	0003E	MOVAB	2(R3), FCB-24	: 0655	
30	AE		2C	BE40	9E	00041	MOVZWL	(STRING), R0	: 0656	
38	AE			52	3C	00047	MOVAB	@FCB-24[R0], FCB-20	: 0657	
34	AE		2C	AE	C1	0004B	MOVZWL	MAX_LENGTH, FCB-12	: 0658	
			65	64	FA	00052	ADDL3	FCB-12, FCB-24, FCB-16	: 0658	
63	AE		2C	AE	A3	00055	CALLG	(ARGUMENT_LIST), (ROUTINE_ADDR)	: 0664	
			30	AE	A3	00055	SUBW3	FCB-24, FCB-20, (STRING)	: 0670	
			5E	44	AE	9E	0005B	MOVAB	68(SP), SP	: 0674
				05	0005F		RSB		: :	

: Routine Size: 96 bytes, Routine Base: \_PASS\$CODE + 0143

: 618 0675 1  
: 619 0676 1 !<BLF/PAGE>

PASSWRITELN2  
1-002

WRITELN procedure  
PASS\$DO\_WRITEV - Do a WRITEV

C 12  
16-Sep-1984 02:26:26  
14-Sep-1984 12:52:09

VAX-11 Bliss-32 V4.0-742  
[PASRTL.SRC]PASWRITE.LB32;1

Page 17  
(8)

: 021 0677 1 END  
: 622 0678 1  
: 623 0679 0 ELUDOM

. End of module PASSWRITELN2

PSECT SUMMARY

Name Bytes Attributes  
\_PASS\$CODE 419 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	8	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	130	30	33	00:00.4

COMMAND QUALIFIERS

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

: Size: 398 code + 21 data bytes  
: Run Time: 00:12.8  
: Elapsed Time: 00:28.5  
: Lines/CPU Min: 3190  
: Lexemes/CPU-Min: 21735  
: Memory Used: 99 pages  
: Compilation Complete



