

PPPPPPPPPPPP		AAAAAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTT		LLL
PPPPPPPPPPPP		AAAAAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTT		LLL
PPPPPPPPPPPP		AAAAAAAAAA		SSSSSSSSSSSS		RRRRRRRRRRRR		TTTTTTTTTTTTTT		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		AAAAAAAAAAAAAAAA		SSS		RRR	RRR	TTT		LLL
PPP		AAAAAAAAAAAAAAAA		SSS		RRR	RRR	TTT		LLL
PPP		AAAAAAAAAAAAAAAA		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	SSS		RRR	RRR	TTT		LLL
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL
PPP		AAA	AAA	SSSSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL

52

Sym

PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS

PAS
PASPAS
PAS

1992

PAS

PAS

FAS

PAS

PAS

PAS
PAS

PAS

PAS

PAS

PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS

PAS

```
PPPPPPPP      AAAAAA      SSSSSSSS      VV      VV      AAAAAA      LL      IIIIII      DDDDDDDD      AAAAAA
PPPPPPPP      AAAAAA      SSSSSSSS      VV      VV      AAAAAA      LL      IIIIII      DDDDDDDD      AAAAAA
PP      PP      AA      AA      SS      VV      VV      AA      AA      LL      II      DD      DD      AA      AA
PP      PP      AA      AA      SS      VV      VV      AA      AA      LL      II      DD      DD      AA      AA
PP      PP      AA      AA      SS      VV      VV      AA      AA      LL      II      DD      DD      AA      AA
PP      PP      AA      AA      SS      VV      VV      AA      AA      LL      II      DD      DD      AA      AA
PPPPPPPP      AA      AA      SSSSSS      VV      VV      AA      AA      LL      II      DD      DD      AA      AA
PPPPPPPP      AA      AA      SSSSSS      VV      VV      AA      AA      LL      II      DD      DD      AA      AA
PP      AAAAAAAAAA      SS      VV      VV      AAAAAAAAAA      LL      II      DD      DD      AAAAAAAAAA
PP      AAAAAAAAAA      SS      VV      VV      AAAAAAAAAA      LL      II      DD      DD      AAAAAAAAAA
PP      AA      AA      SS      VV      VV      AA      AA      LL      II      DD      DD      AA      AA
PP      AA      AA      SS      VV      VV      AA      AA      LL      II      DD      DD      AA      AA
PP      AA      AA      SSSSSSSS      VV      VV      AA      AA      LLLLLLLLLL      IIIIII      DDDDDDDD      AA      AA
PP      AA      AA      SSSSSSSS      VV      VV      AA      AA      LLLLLLLLLL      IIIIII      DDDDDDDD      AA      AA
```

```
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 PASS$VALIDATE PFV ( %TITLE 'Validate PFV fields'
2 0002 0 IDENT = '1-002' ! File: PASVALIDA.B32 Edit: SBL1002
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains a procedure which is called by other
36 0036 1 VAX-11 Pascal Run-Time Library procedures to verify that
37 0037 1 a PFV is valid.
38 0038 1
39 0039 1 ENVIRONMENT: User mode - AST reentrant
40 0040 1
41 0041 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 1-001 - Original. SBL 1-April-1981
46 0046 1 1-002 - Add global symbol PASS$VALIDATE PFV END so that PASS$IO HANDLER can
47 0047 1 determine the PC range of PASS$VALIDATE_PFV. SBL 14-Jul-1982
48 0048 1 --
49 0049 1
```

```

51      0050 1 %SBTTL 'Declarations'
52      0051 1
53      0052 1 PROLOGUE DEFINITIONS:
54      0053 1
55      0054 1
56      0055 1 REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures
57      0119 1
58      0120 1 !+
59      0121 1 ! Linkage for VALIDATE_ERROR.
60      0122 1 !-
61      0123 1
62      0124 1 LINKAGE
63      0125 1     JSB_VALIDATE_ERROR =
64      0126 1     JSB (REGISTER=0, REGISTER=6): NOTUSED (2,3,4,5,7,8,9,10,11);
65      0127 1
66      0128 1
67      0129 1 TABLE OF CONTENTS:
68      0130 1
69      0131 1
70      0132 1 FORWARD ROUTINE
71      0133 1     PASS$VALIDATE PFV: JSB_VALIDATE PFV NOVALUE,           ! Validate PFV fields
72      0134 1     VALIDATE_ERROR: JSB_VALIDATE_ERROR NOVALUE;         ! Signal validation error
73      0135 1
74      0136 1 MACROS:
75      0137 1
76      0138 1     NONE
77      0139 1
78      0140 1 EQUATED SYMBOLS:
79      0141 1
80      0142 1
81      0143 1 !+
82      0144 1 Codes used to indicate errors to VALIDATE_ERROR.
83      0145 1 !-
84      0146 1
85      0147 1 LITERAL
86      0148 1     K_INVFILVAR = 0,      ! PASS_INVFILVAR, invalid file variable
87      0149 1     K_FILALRACT = 1;     ! PASS_FILALRACT, file already active
88      0150 1
89      0151 1 !+
90      0152 1 Bind the global symbol PASS$VALIDATE_PFV_END to the address of the instruction
91      0153 1 following PASS$VALIDATE_PFV, which is VALIDATE_ERROR. This is so that
92      0154 1 PASS$IO_HANDLER can look to see if an access violation occurred in
93      0155 1 PASS$VALIDATE_PFV and therefore turn it into a PASS_INVFILVAR error.
94      0156 1 !-
95      0157 1
96      0158 1 GLOBAL BIND
97      0159 1     PASS$VALIDATE_PFV_END = VALIDATE_ERROR;
98      0160 1
99      0161 1
100     0162 1 FIELDS:
101     0163 1
102     0164 1     NONE
103     0165 1
104     0166 1 OWN STORAGE:
105     0167 1
106     0168 1     NONE

```



```

108 0169 1 %SBTTL 'PAS$$VALIDATE_PFV - Validate PFV fields'
109 0170 1 GLOBAL ROUTINE PAS$$VALIDATE_PFV (
110 0171 1     PFV: REF $PASSPFV_FILE VARIABLE;      ! File variable
111 0172 1     FCB: REF $PASSFCB_CONTROL_BLOCK      ! Control block
112 0173 1 ) : JSB_VALIDATE_PFV NOVALUE =
113 0174 1
114 0175 1 !++
115 0176 1 FUNCTIONAL DESCRIPTION:
116 0177 1
117 0178 1     This procedure is called at the beginning of all VAX-11 Pascal
118 0179 1     I/O procedures to verify that the PFV is in fact a valid file
119 0180 1     variable. It also locks the PFV against recursive I/O.
120 0181 1
121 0182 1 CALLING SEQUENCE:
122 0183 1
123 0184 1     JSB_VALIDATE_PFV PAS$$VALIDATE_PFV (PFV.rr.r; FCB.mr.r)
124 0185 1
125 0186 1 FORMAL PARAMETERS:
126 0187 1
127 0188 1     PFV          - The Pascal File Variable (PFV) passed by reference.
128 0189 1                  The structure of the PFV is defined in PASPFV.REQ.
129 0190 1
130 0191 1     FCB          - The File Control Block (FCB) address is stored in
131 0192 1                  the FCB argument. If there is no FCB, zero is stored.
132 0193 1
133 0194 1 IMPLICIT INPUTS:
134 0195 1
135 0196 1     NONE
136 0197 1
137 0198 1 IMPLICIT OUTPUTS:
138 0199 1
139 0200 1     NONE
140 0201 1
141 0202 1 ROUTINE VALUE:
142 0203 1
143 0204 1     NONE
144 0205 1
145 0206 1 SIDE EFFECTS:
146 0207 1
147 0208 1     NONE
148 0209 1
149 0210 1 SIGNALLED ERRORS:
150 0211 1
151 0212 1     INVFILVAR - invalid or corrupted file variable
152 0213 1     FILALRACT - file already active
153 0214 1
154 0215 1 !--
155 0216 1
156 0217 2 BEGIN
157 0218 2
158 0219 2 BUILTIN
159 0220 2     TESTBITSS;
160 0221 2
161 0222 2 !+
162 0223 2     Test for valid PFV
163 0224 2 !-
164 0225 2
```

```

: 165      0226 2      IF .PFV [PFV$B_VERSION] GTRU PFV$K_CUR_VERSION
: 166      0227 2      THEN
: 167      0228 2          BEGIN
: 168      0229 2          VALIDATE_ERROR (K_INVFILVAR);
: 169      0230 2          RETURN;
: 170      0231 2          END;
: 171      0232 2
: 172      0233 2      !+
: 173      0234 2      !- Test for already active file and lock this PFV
: 174      0235 2      !-
: 175      0236 2
: 176      0237 2      IF TESTBITSS (PFV [PFV$V_LOCK])
: 177      0238 2      THEN
: 178      0239 2          BEGIN
: 179      0240 2          VALIDATE_ERROR (K_FILALRACT);
: 180      0241 2          RETURN;
: 181      0242 2          END;
: 182      0243 2
: 183      0244 2      !+
: 184      0245 2      !- Get FCB address and do another validity test.
: 185      0246 2      !-
: 186      0247 2
: 187      0248 2      IF .PFV [PFV$V_FCB_VALID]
: 188      0249 2      THEN
: 189      0250 2          BEGIN
: 190      0251 2          FCB = .PFV [PFV$A_FCB];
: 191      0252 2          IF .FCB [FCB$A_PFV] NEQA PFV [PFV$R_PFV] ! Compare back-pointer
: 192      0253 2          THEN
: 193      0254 2              BEGIN
: 194      0255 2              VALIDATE_ERROR (K_INVFILVAR);
: 195      0256 2              RETURN;
: 196      0257 2              END
: 197      0258 2          END
: 198      0259 2      ELSE IF .PFV [PFV$V_OPEN] ! FCB invalid and QPEN? Error!
: 199      0260 2      THEN
: 200      0261 2          BEGIN
: 201      0262 2          VALIDATE_ERROR (K_INVFILVAR);
: 202      0263 2          RETURN;
: 203      0264 2          END
: 204      0265 2      ELSE
: 205      0266 2          FCB = 0;
: 206      0267 2
: 207      0268 2      RETURN;
: 208      0269 2
: 209      0270 1      END;

```

! End of routine PASS\$VALIDATE_PFV

.TITLE PASS\$VALIDATE_PFV Validate PFV fields
.IDENT \1-002\

.EXTRN PASS\$VALIDATE_PFV

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

04 A6 95 00000 PASS\$VALIDATE_PFV::
TSTB 4(PFV)
1F 12 00003 BNEQ 3\$

: 0226
:

PASS\$VALIDATE_P Validate PFV fields
1-002

PASS\$VALIDATE_PFV - Validate PFV fields

N 8
16-Sep-1984 02:12:59
14-Sep-1984 12:51:59

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

Page 5
(3)

05	04	A6	1F	E3	00005	BBCS	#31, 4(PFV), 1\$	:	0237
		50	01	D0	0000A	MOVL	#1, R0	:	0240
			17	11	0000D	BRB	4\$	:	
0B	07	A6	06	E1	0000F	BBC	#6, 7(PFV), 2\$	:	0248
		57	A6	D0	00014	MOVL	12(PFV), FCB	:	0251
		56	A7	D1	00018	CMPL	-36(FCB), PFV	:	0252
			06	12	0001C	BNEQ	3\$	:	
				05	0001E	RSB		:	0255
05	07	A6	05	E1	0001F	BBC	#5, 7(PFV), 5\$	:	0259
			50	D4	00024	CLRL	R0	:	0262
			0000V	31	00026	BRW	VALIDATE_ERROR	:	
			57	D4	00029	CLRL	FCB	:	0266
				05	0002B	RSB		:	0270

; Routine Size: 44 bytes, Routine Base: _PASS\$CODE + 0000

; 210 0271 1
; 211 0272 1 !<BLF/PAGE>


```

213 0273 1 %SBTTL 'VALIDATE_ERROR - Signal validation error'
214 0274 1 ROUTINE VALIDATE_ERROR (
215 0275 1     ERROR_CODE,                ! Code for error to signal
216 0276 1     PFV: REF $PASS$PFV_FILE_VARIABLE ! File variable
217 0277 1     ) : JSB_VALIDATE_ERROR NOVALUE =
218 0278 1
219 0279 1 !++
220 0280 1 FUNCTIONAL DESCRIPTION:
221 0281 1
222 0282 1     This procedure is called by PASS$VALIDATE_PFV to indicate that
223 0283 1     it wants to signal an error.
224 0284 1
225 0285 1 CALLING SEQUENCE:
226 0286 1
227 0287 1     JSB_VALIDATE_ERROR VALIDATE_ERROR (ERROR_CODE.rl.v, PFV.rr.r)
228 0288 1
229 0289 1 FORMAL PARAMETERS:
230 0290 1
231 0291 1     ERROR_CODE      - The code of the error to signal.
232 0292 1                   K_INVFILVAR - Invalid file variable
233 0293 1                   K_FILALRACT - File already active
234 0294 1
235 0295 1     PFV          - The Pascal File Variable (PFV) passed by reference.
236 0296 1                   The structure of the PFV is defined in PASPFV.REQ.
237 0297 1
238 0298 1 IMPLICIT INPUTS:
239 0299 1
240 0300 1     NONE
241 0301 1
242 0302 1 IMPLICIT OUTPUTS:
243 0303 1
244 0304 1     NONE
245 0305 1
246 0306 1 ROUTINE VALUE:
247 0307 1
248 0308 1     NONE
249 0309 1
250 0310 1 SIDE EFFECTS:
251 0311 1
252 0312 1     NONE
253 0313 1
254 0314 1 SIGNALLED ERRORS:
255 0315 1
256 0316 1     INVFILVAR - invalid or corrupted file variable
257 0317 1     FILALRACT - file already active
258 0318 1
259 0319 1 !--
260 0320 1
261 0321 2 BEGIN
262 0322 2
263 0323 2 LOCAL
264 0324 2     PFD: REF $PASS$PFD_FILE_DESCRIPTOR;
265 0325 2
266 0326 2 !+
267 0327 2     If error is "invalid file variable, just signal it.
268 0328 2 !-
269 0329 2

```



```

: 270      0330 2      IF .ERROR_CODE EQL K_INVFILVAR
: 271      0331 2      THEN
: 272      0332 2      BEGIN
: 273      0333 2      SIGNAL_STOP (PASS$_INVFILVAR,1,PFV [PFV$R_PFV]);
: 274      0334 2      RETURN;
: 275      0335 2      END;
: 276      0336 2
: 277      0337 2      !+
: 278      0338 2      ! Must be "file already active". Include file variable name in
: 279      0339 2      ! signal if the PFD address is not self-relative. If it is, use
: 280      0340 2      ! a dummy string.
: 281      0341 2      !-
: 282      0342 2
: 283      0343 2      PFD = .PFV [PFV$A_PFD];      ! Get PFD address
: 284      0344 3      SIGNAL_STOP (PASS$_FILALRACT,1,(
: 285      0345 3      IF .PFV [PFV$V_RELFPD]
: 286      0346 3      THEN
: 287      0347 3      UPLIT BYTE (%CHARCOUNT('unknown'),'unknown')
: 288      0348 3      ELSE
: 289      0349 3      PFD [PFD$T_NAME]
: 290      0350 2      ));
: 291      0351 2
: 292      0352 2      RETURN;
: 293      0353 2
: 294      0354 1      END;

```

! End of routine VALIDATE_ERROR

```

6E 77 6F 6E 6B 6E 07 0002C P.AAA: .BYTE 7
75 0002D .ASCII \unknown\
                                .EXTRN PASS$_INVFILVAR, PASS$_FILALRACT

```

50	D5	00000	VALIDATE_ERROR:	TSTL	ERROR_CODE	: 0330
	0C	12	00002	BNEQ	1\$	
	56	DD	00004	PUSHL	PFV	: 0333
	01	DD	00006	PUSHL	#1	
	8F	DD	00008	PUSHL	#PASS\$_INVFILVAR	
	1E	11	0000E	BRB	4\$	
08	50	08	A6 DD 00010	1\$: MOVL	8(PFV), PFD	: 0343
	A6		04 E1 00014	BBC	#4, 7(PFV), 2\$	: 0345
	51	DC	AF 9E 00019	MOVAB	P.AAA, R1	: 0347
			51 DD 0001D	PUSHL	R1	
			05 11 0001F	BRB	3\$	
	50		0C C0 00021	2\$: ADDL2	#12, R0	: 0349
			50 DD 00024	PUSHL	R0	
			01 DD 00026	3\$: PUSHL	#1	: 0344
		00000000G	8F DD 00028	PUSHL	#PASS\$_FILALRACT	
00000000G	00		03 FB 0002E	4\$: CALLS	#3, LIB\$STOP	
			05 00035	RSB		: 0354

; Routine Size: 54 bytes, Routine Base: _PASS\$CODE + 0034

```

: 295      0355 1
: 296      0356 1 !<BLF/PAGE>

```


PASS\$VALIDATE_P Validate PFV fields
1-002 VALIDATE_ERROR - Signal validation error

D 9
16-Sep-1984 02:12:59 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:59 [PASRTL.SRC]PASVALIDA.B32;1

Page 8
(5)

: 298 0357 1 END
: 299 0358 1
: 300 0359 0 ELUDOM

! End of module PASS\$VALIDATE_PFV

.EXTRN LIB\$STOP
PASS\$VALIDATE_PFV_END==
VALIDATE_ERROR

PSECT SUMMARY

Name	Bytes	Attributes
_PASS\$CODE	106	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	106	24	33	00:00.4

COMMAND QUALIFIERS

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

: Size: 98 code + 8 data bytes
: Run Time: 00:05.4
: Elapsed Time: 00:19.0
: Lines/CPU Min: 3988
: Lexemes/CPU-Min: 9577
: Memory Used: 65 pages
: Compilation Complete

0297

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY