


```

1 0001 0 MODULE PASSRESET2 ( %TITLE 'RESET procedure'
2 0002 0 IDENT = '1-002' ! File: PASRESET2.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 PASSRESET2, which implements the
36 0036 1 VAX-11 Pascal RESET 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 - Only do $REWIND on file oriented devices. SBL 29-July-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';           ! Externals, linkages, PSECTs, structures
: 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   PASSRESET2: NOVALUE;               ! Do a RESET
: 62      0124 1
: 63      0125 1
: 64      0126 1  ! MACROS:
: 65      0127 1
: 66      0128 1   NONE
: 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   NONE

```

```

80 0141 1 %SBTTL 'PASSRESET2 - RESET procedure'
81 0142 1 GLOBAL ROUTINE PASSRESET2 (
82 0143 1     PFV: REF $PASSPFV_FILE_VARIABLE,
83 0144 1     ERROR
84 0145 1     ): NOVALUE =
85 0146 1
86 0147 1
87 0148 1  +-+
88 0149 1  FUNCTIONAL DESCRIPTION:
89 0150 1      PASSRESET2 implements the VAX-11 Pascal RESET procedure. It
90 0151 1      opens the file if not already open, rewinds to the beginning of
91 0152 1      the file, enters Inspection mode and does a GET.
92 0153 1
93 0154 1  CALLING SEQUENCE:
94 0155 1      CALL PASSRESET2 (PFV.mr.r)
95 0156 1
96 0157 1
97 0158 1  FORMAL PARAMETERS:
98 0159 1
99 0160 1      PFV          - The Pascal File Variable (PFV) passed by reference.
100 0161 1      The structure of the PFV is defined in PASFV.REQ.
101 0162 1
102 0163 1      ERROR      - Optional. If specified, the address to unwind to
103 0164 1      in case of an error.
104 0165 1
105 0166 1  IMPLICIT INPUTS:
106 0167 1      NONE
107 0168 1
108 0169 1  IMPLICIT OUTPUTS:
109 0170 1      NONE
110 0171 1
111 0172 1  ROUTINE VALUE:
112 0173 1      NONE
113 0174 1
114 0175 1  SIDE EFFECTS:
115 0176 1      Throws away partial line contents, if any.
116 0177 1      Positions to beginning of file.
117 0178 1      Switches to Inspection mode.
118 0179 1      Does a GET.
119 0180 1
120 0181 1
121 0182 1
122 0183 1
123 0184 1
124 0185 1  SIGNALLED ERRORS:
125 0186 1      ERRDURRES - Error during RESET
126 0187 1      RESNOTALL - RESET not allowed on unopened internal file
127 0188 1      INSNOTALL - Inspection mode not allowed for WRITEONLY file
128 0189 1
129 0190 1
130 0191 1  --
131 0192 1
132 0193 2  BEGIN
133 0194 2
134 0195 2  LOCAL
135 0196 2      FCB: REF $PASSFCB_CONTROL_BLOCK,
136 0197 2      PFV_ADDR: VOLATILE,

```

```

! Do a RESET
! File variable
! Error unwind address

```

```

! File control block
! Enable argument

```

```

: 137 0198 2      UNWIND_ACT: VOLATILE,          ! Enable argument
: 138 0199 2      ERROR_ADDR: VOLATILE;      ! Enable argument
: 139 0200 2
: 140 0201 2      BIND
: 141 0202 2      RAB = FCB: REF BLOCK [, BYTE];      ! RAB is also FCB address
: 142 0203 2
: 143 0204 2      BUILTIN
: 144 0205 2      ACTUALCOUNT;
: 145 0206 2
: 146 0207 2      ENABLE
: 147 0208 2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);      ! Enable error handler
: 148 0209 2
: 149 0210 2      IF ACTUALCOUNT () GEQU 2
: 150 0211 2      THEN
: 151 0212 2      ERROR_ADDR = .ERROR;          ! Set unwind address
: 152 0213 2
: 153 0214 2      !+
: 154 0215 2      ! Set enable argument for PFV address.
: 155 0216 2      !-
: 156 0217 2
: 157 0218 2      PFV_ADDR = PFV [PFV$R_PFV];
: 158 0219 2
: 159 0220 2      !+
: 160 0221 2      ! Validate and lock PFV.
: 161 0222 2      !-
: 162 0223 2
: 163 0224 2      PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
: 164 0225 2
: 165 0226 2      !+
: 166 0227 2      ! Set unwind action to unlock file.
: 167 0228 2      !-
: 168 0229 2
: 169 0230 2      UNWIND_ACT = PASS$UNWIND_UNLOCK;
: 170 0231 2
: 171 0232 2      !+
: 172 0233 2      ! Open file if it should be implicitly opened.
: 173 0234 2      !-
: 174 0235 2
: 175 0236 2      IF NOT .PFV [PFV$V_VALID]
: 176 0237 2      THEN
: 177 0238 2      PASS$OPEN_IMPLICIT (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
: 178 0239 2
: 179 0240 2      !+
: 180 0241 2      ! See if the file is open. If not, open it.
: 181 0242 2      !-
: 182 0243 2
: 183 0244 2      IF NOT .PFV [PFV$V_OPEN]      ! Not open
: 184 0245 2      THEN
: 185 0246 2      BEGIN
: 186 0247 2      LOCAL
: 187 0248 2      PFD: REF $PASS$PFD_FILE_DESCRIPTOR;      ! File descriptor
: 188 0249 2
: 189 0250 2      !+
: 190 0251 2      ! If PFD address is relative, make it absolute.
: 191 0252 2      !-
: 192 0253 2
: 193 0254 2      IF .PFV [PFV$V_RELPFD]

```

```

194 0255 3      THEN
195 0256 4      BEGIN
196 0257 4      PFV [PFV$A_PFD] = .PFV [PFV$A_PFD] + PFV [PFV$R_PFV];
197 0258 4      PFV [PFV$V_REL_PFD] = 0;
198 0259 4      END;
199 0260 3
200 0261 3      PFD = .PFV [PFV$A_PFD]; ! Get PFD address
201 0262 3
202 0263 3      IF .PFD [PFD$V_EXTERN] ! Is it an external file?
203 0264 3      THEN
204 0265 3          PASS$OPEN (PFV [PFV$R_PFV], PASS$HISTORY_OLD; FCB)
205 0266 3      ELSE
206 0267 3          $PASSIO_ERROR (PASS$_RESNOTALL,0); ! RESET not allowed
207 0268 3      END;
208 0269 3
209 0270 3      !+
210 0271 3      ! If the file has the NOREAD attribute, signal an error.
211 0272 3      !-
212 0273 3
213 0274 3      IF .FCB [FCB$V_NOREAD]
214 0275 3      THEN
215 0276 3          $PASSIO_ERROR (PASS$_INSNOTALL,0); ! Inspection mode not allowed
216 0277 3
217 0278 3      !+
218 0279 3      ! If a textfile, and if in Generation mode, and if the buffer
219 0280 3      ! is not empty, do a WRITELN.
220 0281 3      !-
221 0282 3
222 0283 3      IF .FCB [FCB$V_TEXT] AND .FCB [FCB$V_GENERATION] AND
223 0284 3          (.FCB [FCB$A_RECORD_CUR] GTKA .FCB [FCB$A_RECORD_BEG])
224 0285 3      THEN
225 0286 3          PASS$WRITELN (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
226 0287 3
227 0288 3      !+
228 0289 3      ! Cancel any lazy lookahead in progress.
229 0290 3      !-
230 0291 3
231 0292 3      FCB [FCB$V_LAZY] = 0;
232 0293 3
233 0294 3      !+
234 0295 3      ! Is the device a file oriented device? If so, proceed with rewind.
235 0296 3      !-
236 0297 3
237 0298 3      IF .FCB [FCB$V_FOD] ! File Oriented Device
238 0299 3      THEN
239 0300 3          BEGIN
240 0301 3
241 0302 3          RAB [RAB$B_RAC] = RAB$C_SEQ; ! Set sequential access
242 0303 3          RAB [RAB$V_NLK] = 0; ! Cancel previous no-lock
243 0304 3          RAB [RAB$B_KRF] = 0; ! Rewind primary key if indexed
244 0305 3          IF NOT $PASSRMS_OP ($REWIND (RAB=.RAB))
245 0306 3          THEN
246 0307 3              IF .RAB [RAB$L_STS] NEQ RMS$_IOP AND ! Inappropriate operation
247 0308 3                  .RAB [RAB$L_STS] NEQ RMS$_EOF AND ! End of file
248 0309 3                  .RAB [RAB$L_STS] NEQ RMS$_BOF ! Already at beginning of file
249 0310 3              THEN
250 0311 3                  $PASSIO_ERROR (PASS$_ERRDURRES); ! Error during RESET

```


			7E	64	0000C	CLRL	ERROR_ADDR	0193
			AE	7C	0000E	CLRQ	UNWIND_ACT	
	6D	00FF	CF	DE	00011	MOVAL	14\$, (FP)	
	02		6C	91	00G16	CMPB	(AP), #2	0210
			04	1F	00019	BLSSU	1\$	
	6E	08	AC	DO	0001B	MOVL	ERROR, ERROR_ADDR	0212
	56	04	AC	DO	0001F	MOVL	PFV, R6	0218
08	AE		56	DO	00023	MOVL	R6, PFV_ADDR	
		00000000G	00	16	00027	JSB	PASS\$VACIDATE_PFV	0224
04	AE		01	DO	0002D	MOVL	#1, UNWIND_ACT	0230
	58	04	A6	9E	00031	MOVAB	4(R6), R8	0236
	06	02	A8	E8	00035	BLBS	2(R8), 2\$	
		00000000G	00	16	00039	JSB	PASS\$OPEN_IMPLICIT	0238
2A	68		1D	E0	0003F	BBS	#29, (R8), 5\$	0244
08	68		1C	E1	00043	BBC	#28, (R8), 3\$	0254
	08		56	C0	00047	ADDL2	R6, 8(R6)	0257
	03		10	8A	0004B	BICB2	#16, 3(R8)	0258
			A6	DO	0004F	MOVL	8(R6), PFD	0261
0D	04	08	05	E1	00053	BBC	#5, 4(PFD), 4\$	0263
			03	DD	00058	PUSHL	R6	0265
		00000000G	56	DD	0005A	PUSHL	R6	
			02	FB	0005C	CALLS	#2, PASS\$OPEN	
			08	11	00063	BRB	5\$	
			7E	D4	00065	CLRL	-(SP)	0267
	7E	00G	8F	9A	00067	MOVZBL	#PASSK_RESNOTALL, -(SP)	
0A	F8	A7	0B	11	0006B	BRB	6\$	
			03	E1	0006D	BBC	#3, -8(FCB), 7\$	0274
			7E	D4	00072	CLRL	-(SP)	0276
	7E	00G	8F	9A	00074	MOVZBL	#PASSK_INSNOTALL, -(SP)	
	69		02	FB	00078	CALLS	#2, PASS\$SIGNAL	
				04	0007B	RET		
	12	F8	A7	E9	0007C	BLBC	-8(FCB), 8\$	0283
0D	FD	A7	04	E1	00080	BBC	#4, -3(FCB), 8\$	
	E8	A7	A7	D1	00085	CMP	-20(FCB), -24(FCB)	0284
			06	1B	0008A	BLEQU	8\$	
		00000000G	00	16	0008C	JSB	PASS\$WRITELN	0286
			A7	9E	00092	MOVAB	-4(FCB), R2	0292
	01	52	04	8A	00096	BICB2	#4, 1(R2)	
		4C	A2	E9	0009A	BLBC	3(R2), 11\$	0298
			A7	94	0009E	CLRB	30(FCB)	0302
	06	A7	10	8A	000A1	BICB2	#16, 6(FCB)	0303
			A7	94	000A5	CLRB	53(FCB)	0304
			57	DD	000AB	PUSHL	FCB	0305
		00000000G	01	FB	000AA	CALLS	#1, SYSSREWIND	
		36	50	E8	000B1	BLBS	\$\$STATUS, 11\$	
		0001825A	50	D1	000B4	CMP	\$\$STATUS, #98906	
			04	12	000BB	BNEQ	10\$	
			A2	E8	000BD	BLBS	3(R2), 9\$	
		00018574	50	E8	000C1	BLBS	\$\$STATUS, 11\$	
			A7	D1	000C4	CMP	8(FCB), #99700	0307
			1C	13	000CC	BEQL	11\$	
		0001827A	A7	D1	000CE	CMP	8(FCB), #98938	0308
			12	13	000D6	BEQL	11\$	
		00018198	A7	D1	000D8	CMP	8(FCB), #98712	0309
			08	13	000E0	BEQL	11\$	
			8F	9A	000E2	MOVZBL	#PASSK_ERRDURRES, -(SP)	0311
			01	FB	000E6	CALLS	#1, PASS\$SIGNAL	

PASSRESET2
1-G02

RESET procedure
PASSRESET2 - RESET procedure

D 16
16-Sep-1984 02:05:23 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:54 [PASRTL.SRC]PASRESET2.B32;1

Page 9
(4)

: 291 0351 1 END
: 292 0352 1
: 293 0353 0 ELUDOM

! End of module PASSRESET2

PSECT SUMMARY

Name	Bytes	Attributes
_PASSCODE	311	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	13	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	122	28	33	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASRESET2/OBJ=OBJ\$:PASRESET2 MSRC\$:PASRESET2/UPDATE-(ENH\$:PASRESET2)

: Size: 311 code + 0 data bytes
: Run Time: 00:08.4
: Elapsed Time: 00:28.7
: Lines/CPU Min: 2536
: Lexemes/CPU-Min: 17446
: Memory Used: 136 pages
: Compilation Complete

