


```

PPPPPPPP      AAAAAA      SSSSSSSS      CCCCCCCC      LL      000000      SSSSSSSS      EEEEEEEEEE      222222
PPPPPPPP      AAAAAA      SSSSSSSS      CCCCCCCC      LL      000000      SSSSSSSS      EEEEEEEEEE      222222
PP      PP      AA      AA      SS      CC      LL      00      00      SS      EE      22      22
PP      PP      AA      AA      SS      CC      LL      00      00      SS      EE      22      22
PP      PP      AA      AA      SS      CC      LL      00      00      SS      EE      22      22
PP      PP      AA      AA      SS      CC      LL      00      00      SS      EE      22      22
PPPPPPPP      AA      AA      SSSSSS      CC      LL      00      00      SSSSSS      EEEEEEEE      22
PPPPPPPP      AA      AA      SSSSSS      CC      LL      00      00      SSSSSS      EEEEEEEE      22
PP      AAAAAAAAAA      SS      CC      LL      00      00      SS      EE      22
PP      AAAAAAAAAA      SS      CC      LL      00      00      SS      EE      22
PP      AA      AA      SS      CC      LL      00      00      SS      EE      22
PP      AA      AA      SS      CC      LL      00      00      SS      EE      22
PP      AA      AA      SSSSSSSS      CCCCCCCC      LLLLLLLLLL      000000      SSSSSSSS      EEEEEEEEEE      2222222222      ....
PP      AA      AA      SSSSSSSS      CCCCCCCC      LLLLLLLLLL      000000      SSSSSSSS      EEEEEEEEEE      2222222222      ....

```

```

LL      I I I I I I      SSSSSSSS
LL      I I I I I I      SSSSSSSS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SSSSSS
LL      I I      SSSSSS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LLLLLLLLLLLL      I I I I I I      SSSSSSSS
LLLLLLLLLLLL      I I I I I I      SSSSSSSS

```

```

1 0001 0 MODULE PASSCLOSE2 ( %TITLE 'close a file'
2 0002 0 IDENT = '1-006' ! File: PASCLOSE2.B32 Edit: SBL1006
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 PASSCLOSE2 and PASS$CLOSE, which close a file.
36 0036 1
37 0037 1 ENVIRONMENT: User mode - AST reentrant
38 0038 1
39 0039 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
40 0040 1
41 0041 1 MODIFIED BY:
42 0042 1
43 0043 1 1-001 - Original. SBL 1-April-1981
44 0044 1 1-002 - Give error in PASSCLOSE2 if FCB is not valid. SBL 21-June-1982
45 0045 1 1-003 - Correct erroneous error check in PASS$CLOSE. SBL 9-July-1982
46 0046 1 1-004 - Make sure XAB chain doesn't point to garbage. SBL 29-July-1982
47 0047 1 1-005 - Zero FCB$L STATUS before closing file. QAR FT3-2. SBL 30-Aug-1982
48 0048 1 1-006 - Use PASS$AB KEYWD_NAME_TABLE to detect duplicate keywords.
49 0049 1 SBL 19-Aug-1983
50 0050 1 --
51 0051 1

```

```
: 53 0052 1 %SBTTL 'Declarations'  
: 54 0053 1 |  
: 55 0054 1 | PROLOGUE DEFINITIONS:  
: 56 0055 1 |  
: 57 0056 1 |  
: 58 0057 1 REQUIRE 'RTLIN:PASPROLOG';           ! Linkages, externals, PSECTs, structures  
: 59 0121 1 |  
: 60 0122 1 |  
: 61 0123 1 | TABLE OF CONTENTS:  
: 62 0124 1 |  
: 63 0125 1 |  
: 64 0126 1 FORWARD ROUTINE  
: 65 0127 1     PASSCLOSE2: NOVALUE,           ! Called by compiled code  
: 66 0128 1     PASS$CLOSE: NOVALUE.         ! Called internally by RTL  
: 67 0129 1 |  
: 68 0130 1 |  
: 69 0131 1 | MACROS:  
: 70 0132 1 |  
: 71 0133 1 |     NONE  
: 72 0134 1 |  
: 73 0135 1 | EQUATED SYMBOLS:  
: 74 0136 1 |  
: 75 0137 1 |     NONE  
: 76 0138 1 |  
: 77 0139 1 | FIELDS:  
: 78 0140 1 |  
: 79 0141 1 |     NONE  
: 80 0142 1 |  
: 81 0143 1 | OWN STORAGE:  
: 82 0144 1 |  
: 83 0145 1 |     NONE
```

```

: 85 0146 1 %SBTTL 'PASSCLOSE2 - Close a file'
: 86 0147 1 GLOBAL ROUTINE PASSCLOSE2 (           ! Close a file
: 87 0148 1                                     ! File variable
: 88 0149 1     PFV: REF $PASSPFV_FILE_VARIABLE, ! File variable
: 89 0150 1     KEYWORDS: VECTOR [, LONG]       ! Keywords and specifiers
: 90 0151 1 ): NOVALUE =
: 91 0152 1
: 92 0153 1 |**
: 93 0154 1 | FUNCTIONAL DESCRIPTION:
: 94 0155 1 |     This routine closes a Pascal file. This entry is called from the
: 95 0156 1 |     compiled code for the CLOSE procedure.
: 96 0157 1 |
: 97 0158 1 | CALLING SEQUENCE:
: 98 0159 1 |
: 99 0160 1 |     CALL PASSCLOSE2 (PFV.mr.r [, keyword.rz.v [, keyword.rz.v ...]])
: 100 0161 1 |
: 101 0162 1 | FORMAL PARAMETERS:
: 102 0163 1 |
: 103 0164 1 |     PFV           - The Pascal File Variable (PFV) passed by reference.
: 104 0165 1 |                   The structure of the PFV is defined in PASFV.REQ.
: 105 0166 1 |
: 106 0167 1 |     keyword       - Zero or more values representing keywords in the
: 107 0168 1 |                   CLOSE procedure. Some keywords are followed by
: 108 0169 1 |                   associated arguments and values. See PASOPEKEY.REQ
: 109 0170 1 |                   for the keyword value definitions.
: 110 0171 1 |
: 111 0172 1 |                   If the keyword ERROR is specified, it must be first
: 112 0173 1 |                   in the list, otherwise it will be ignored.
: 113 0174 1 |
: 114 0175 1 | IMPLICIT INPUTS:
: 115 0176 1 |
: 116 0177 1 |     NONE
: 117 0178 1 |
: 118 0179 1 | IMPLICIT OUTPUTS:
: 119 0180 1 |
: 120 0181 1 |     NONE
: 121 0182 1 |
: 122 0183 1 | COMPLETION STATUS: (or ROUTINE VALUE:)
: 123 0184 1 |
: 124 0185 1 |     NONE
: 125 0186 1 |
: 126 0187 1 | SIDE EFFECTS:
: 127 0188 1 |
: 128 0189 1 |     See PASS$CLOSE
: 129 0190 1 |
: 130 0191 1 | SIGNALLED ERRORS:
: 131 0192 1 |
: 132 0193 1 |     FILALRCLO - file already closed
: 133 0194 1 |
: 134 0195 1 | --
: 135 0196 1 |
: 136 0197 2 | BEGIN
: 137 0198 2 |
: 138 0199 2 | LOCAL
: 139 0200 2 |     FCB: REF $PASSFCB_CONTROL_BLOCK, ! File Control Block
: 140 0201 2 |     PFV_ADDR: VOLATILE,             ! Enable argument
: 141 0202 2 |     UNWIND_ACT: VOLATILE,          ! Enable argument

```

```

142 0203 2      ERROR_ADDR: VOLATILE;          Enable argument
143 0204 2
144 0205 2      BUILTIN
145 0206 2      ACTUALCOUNT,
146 0207 2      AP,
147 0208 2      CALLG;
148 0209 2
149 0210 2      ENABLE
150 0211 2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);
151 0212 2
152 0213 2      |*
153 0214 2      | Set PFV_ADDR enable argument.
154 0215 2      |
155 0216 2
156 0217 2      PFV_ADDR = PFV [PFV$R_PFV];
157 0218 2
158 0219 2      |*
159 0220 2      | Validate and lock PFV.
160 0221 2      |
161 0222 2
162 0223 2      PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
163 0224 2
164 0225 2      |*
165 0226 2      | Set unwind action to unlock file.
166 0227 2      |
167 0228 2
168 0229 2      UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
169 0230 2
170 0231 2      |*
171 0232 2      | If ERROR is present, it is the first keyword in the list. See
172 0233 2      | if that is true. If so, set enable argument to the return address.
173 0234 2      |
174 0235 2
175 0236 2      IF ACTUALCOUNT () GEQU 3      ! At least 3 needed for PFV, keyword, value
176 0237 2      THEN
177 0238 2          IF .KEYWORDS [0] EQL PASS$K_ERROR      ! Is it the ERROR keyword?
178 0239 2          THEN
179 0240 2              ERROR_ADDR = .KEYWORDS [1];
180 0241 2
181 0242 2      |*
182 0243 2      | If file variable is not valid, call PASS$OPEN_IMPLICIT to see if the
183 0244 2      | file should be implicitly opened. Even though we will just close it
184 0245 2      | again, we must do this to satisfy the language semantics for INPUT
185 0246 2      | and OUTPUT.
186 0247 2      |
187 0248 2
188 0249 2      IF NOT .PFV [PFV$V_VALID]
189 0250 2      THEN
190 0251 2          PASS$OPEN_IMPLICIT (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
191 0252 2
192 0253 2      |*
193 0254 2      | If FCB pointer is not valid, give an error.
194 0255 2      |
195 0256 2
196 0257 2      IF NOT .PFV [PFV$V_FCB_VALID]
197 0258 2      THEN
198 0259 2          $PASS$IO_ERROR (PASS_FILALRCLO,0);

```

```

: 199 0260 2
: 200 0261 2
: 201 0262 2
: 202 0263 2
: 203 0264 2
: 204 0265 2
: 205 0266 2
: 206 0267 2
: 207 0268 2
: 208 0269 2
: 209 0270 2
: 210 0271 2
: 211 0272 2
: 212 0273 2
: 213 0274 2
: 214 0275 2
: 215 0276 2
: 216 0277 2
: 217 0278 2
: 218 0279 2
: 219 0280 2
: 220 0281 2
: 221 0282 2
: 222 0283 2
: 223 0284 2
: 224 0285 2
: 225 0286 2
: 226 0287 2
: 227 0288 2
: 228 0289 1

```

```

+
Zero the STATUS value in the FCB. This will get stored in the
PFV by PASS$REMOVE_FILE.
-

FCB = .PFV [PFV$A FCB];
FCB [FCB$STATUS] = 0;

+
Call PASS$CLOSE to do the work.
-

CALLG (.AP, PASS$CLOSE);

+
Remove it from the list of open files.
-

PASS$REMOVE_FILE (FCB [FCB$R_FCB]);

+
Unlock the file variable.
-

PFV [PFV$V_LOCK] = 0;

RETURN;

END;

```

! End of routine PASS\$CLOSE2

```

.TITLE PASSCLOSE2 Close a file
.IDENT \1-006\

.EXTRN PASSCLOSE2, PASS$CLOSE
.EXTRN PASS$IO_HANDLER
.EXTRN PASS$VACIDATE PFV
.EXTRN PASS$OPEN_IMPLICIT
.EXTRN PASS$SIGNAL, PASS$FILALRCLO
.EXTRN PASS$REMOVE_FILE

```

```

.PSECT _PASSCODE, NOWRT, SHR, PIC, 2

```

```

00C4 00000
SE 08 C2 00002
7E D4 00005
04 AE 7C 00007
6D 005E CF DE 0000A
56 04 AC D0 0000F
08 AE 56 D0 00013
04 AE 0000G000G 00 16 00017
AE 01 D0 0001D
03 6C 91 00021
19 08 0A 1F 00024
AC D1 00026
04 12 0002A

```

```

.ENTRY PASSCLOSE2, Save R2,R6,R7
SUBL2 #8, SP
CLRL ERROR_ADDR
CLRQ UNWIND_ACT
MOVAL 4$, (FP)
MOVL PFV, R6
MOVL R6, PFV_ADDR
JSB PASS$VACIDATE PFV
MOVL #1, UNWIND_ACT
CMPB (AP), #3
BLSSU 1$
CMPL KEYWORDS, #25
BNEQ 1$

```

```

: 0147
: 0197
: 0217
: 0223
: 0229
: 0236
: 0239

```

PASSCLOSE2
1-006

Close a file
PASSCLOSE2 - Close a file

M 14
16-Sep-1984 01:29:07 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:21 [PASRTL.SRC]PASCLOSE2.B32;1

Page 6
(3)

	6E	0C	AC	DO	0002C		MOVL	KEYWORDS+4, ERROR_ADDR	:	0240
	06	06	A6	E8	00030	1\$:	BLBS	6(R6), 2\$:	0249
		00000000G	00	16	00034		JSB	PASS\$OPEN_IMPLICIT	:	0251
OE	07	A6	06	E0	0003A	2\$:	BBS	#6, 7(R6), 3\$:	0257
			7E	D4	0003F		CLRL	-(SP)	:	0259
	7E	00G	8F	9A	00041		MOVZBL	#PASS\$K.FILALRCLO, -(SP)	:	
	00000000G	00	02	FB	00045		CALLS	#2, PASS\$\$SIGNAL	:	
				04	0004C		RET		:	
	57	0C	A6	DO	0004D	3\$:	MOVL	12(R6), FCB	:	0266
		D4	A7	D4	00051		CLRL	-44(FCB)	:	0267
	0000V	CF	6C	FA	00054		CALLG	(AP), PASS\$\$CLOSE	:	0273
			57	DD	00059		PUSHL	FCB	:	0279
	00000000G	00	01	FB	0005B		CALLS	#1, PASS\$REMOVE_FILE	:	
			50	AC	DO	00062	MOVL	PFV, R0	:	0285
	07	A0	80	8F	8A	00066	BICB2	#128, 7(R0)	:	
				04	0006B		RET		:	0289
				0000	0006C	4\$:	.WORD	Save nothing	:	0197
	50	08	AC	DO	0006E		MOVL	8(AP), R0	:	
	50	04	A0	DO	00072		MOVL	4(R0), R0	:	
			F4	A0	9F	00076	PUSHAB	ERROR_ADDR	:	
			F8	A0	9F	00079	PUSHAB	UNWIND_ACT	:	
			FC	A0	9F	0007C	PUSHAB	PFV_ADDR	:	
				03	DD	0007F	PUSHL	#3	:	
				5E	DD	00081	PUSHL	SP	:	
	7E	04	AC	7D	00083		MOVQ	4(AP), -(SP)	:	
	00000000G	00	03	FB	00087		CALLS	#3, PASS\$IO_HANDLER	:	
				04	0008E		RET		:	

: Routine Size: 143 bytes, Routine Base: _PASS\$CODE + 0000

: 229 0290 1
: 230 0291 1 !<BLF/PAGE>


```

232 0292 1 %SBTTL 'PASS$CLOSE - Close a file'
233 0293 1 GLOBAL ROUTINE PASS$CLOSE (           ! Close a file
234 0294 1     IN PFV: REF $PASS$PFV_FILE_VARIABLE, ! File variable
235 0295 1     KEYWORDS: VECTOR [, [ONG]         ! Keywords and specifiers
236 0296 1     ): NOVALUE =
237 0297 1
238 0298 1 !++
239 0299 1 ! FUNCTIONAL DESCRIPTION:
240 0300 1
241 0301 1     This procedure closes the Pascal file specified by 'PFV'
242 0302 1     using the attributes possibly supplied as 'keyword'. PASS$CLOSE
243 0303 1     is to be called from other RTL routines only after they have tested
244 0304 1     and set PFV$V_LOCK in 'PFV'.
245 0305 1
246 0306 1 ! CALLING SEQUENCE:
247 0307 1
248 0308 1     PASS$CLOSE (PFV.mr.r [, keyword.rz.v [, keyword.rz.v ...]])
249 0309 1
250 0310 1 ! FORMAL PARAMETERS:
251 0311 1
252 0312 1     PFV           - The Pascal File Variable (PFV) passed by reference.
253 0313 1                 The structure of the PFV is defined in PASPFV.MDL.
254 0314 1
255 0315 1     keyword        - Zero or more values representing keywords in the
256 0316 1                 CLOSE procedure. Some keywords are followed by
257 0317 1                 associated arguments and values. See PASOPNKEY.MDL
258 0318 1                 for the keyword value definitions.
259 0319 1
260 0320 1 ! IMPLICIT INPUTS:
261 0321 1
262 0322 1     The PFV$V_LOCK bit is assumed to have been tested-and-set by the caller.
263 0323 1     The PFV$V_FCB_VALID bit is assumed set.
264 0324 1
265 0325 1 ! IMPLICIT OUTPUTS:
266 0326 1
267 0327 1     NONE
268 0328 1
269 0329 1 ! COMPLETION STATUS:
270 0330 1
271 0331 1     NONE
272 0332 1
273 0333 1 ! SIDE EFFECTS:
274 0334 1
275 0335 1     The file is closed. The record buffer and resultant name string
276 0336 1     are deallocated.
277 0337 1
278 0338 1 ! SIGNALLED ERRORS:
279 0339 1
280 0340 1     ERRDURCLO - error during CLOSE
281 0341 1     INVARGPAS - invalid argument to Pascal Run-Time Library
282 0342 1     FILNAMREQ - file name required to save, submit or print file
283 0343 1
284 0344 1 !--
285 0345 1
286 0346 2     BEGIN
287 0347 2
288 0348 2     LOCAL

```

```

: 289      0349      2      PFV: REF $PASS$PFV_FILE_VARIABLE,      Pascal File Variable
: 290      0350      2      FCB: REF $PASS$FCB_CONTROL_BLOCK,      File control block
: 291      0351      2      BLOCK_ADDR: VOLATILE,                  Address of dynamically allocated block
: 292      0352      2      STATUS,                                Status from $CLOSE
: 293      0353      2      KEYWORDS_SEEN: BITVECTOR [PASS$K_OPEKEYHI+1],
: 294      0354      2      USER_ACTION_BPV: REF VECTOR [, LONG];      Keep track of keywords seen
: 295      0355      2      USER_ACTION routine BPV
: 296      0356      2
: 297      0357      2      BUILTIN
: 298      0358      2      TESTBITCS,
: 299      0359      2      TESTBITSC,
: 300      0360      2      ACTUALCOUNT;
: 301      0361      2
: 302      0362      2      BIND
: 303      0363      2      RAB = FCB: REF BLOCK [, BYTE],
: 304      0364      2      FAB = FCB: REF $PASS$FAB_FCB_STRUCT;
: 305      0365      2
: 306      0366      2      !+
: 307      0367      2      !- Get PFV address.
: 308      0368      2
: 309      0369      2      PFV = .IN_PFV;
: 310      0370      2
: 311      0371      2
: 312      0372      2      !+
: 313      0373      2      !- Get addresses of PFV.
: 314      0374      2
: 315      0375      2
: 316      0376      2      FCB = .PFV [PFV$A_FCB];
: 317      0377      2
: 318      0378      2      !+
: 319      0379      2      !- Do the keyword processing.
: 320      0380      2
: 321      0381      2
: 322      0382      2      KEYWORDS_SEEN <0,32> = 0;      ! Indicate no keywords seen
: 323      0383      2      KEYWORDS_SEEN <32,PASS$K_OPEKEYHI+-31> = 0;
: 324      0384      2
: 325      0385      2
: 326      0386      2      !+
: 327      0387      2      !- Look through keyword list and set all specified attributes.
: 328      0388      2
: 329      0389      2
: 330      0390      2      INCR I FROM 0 TO (ACTUALCOUNT () - 2) DO
: 331      0391      2      BEGIN
: 332      0392      2
: 333      0393      2      LOCAL
: 334      0394      2      KEYWD_VALUE;
: 335      0395      2
: 336      0396      2      !+
: 337      0397      2      !- Check for valid keyword code. If not already seen, use the
: 338      0398      2      keyword.
: 339      0399      2
: 340      0400      2
: 341      0401      2      KEYWD_VALUE = .KEYWORDS [I];
: 342      0402      2      IF .KEYWD_VALUE LSS PASS$K_OPEKEYLO OR .KEYWD_VALUE GTR PASS$K_OPEKEYHI
: 343      0403      2      THEN
: 344      0404      2      $PAS$IO_ERROR (PASS$INVARGPAS,0);
: 345      0405      2

```

```
346 0406 3 IF TESTBITCS (KEYWORDS_SEEN [.PASS$AB_KEYWD_NAME_TABLE[.KEYWD_VALUE]])
347 0407 3 THEN
348 0408 3 CASE .KEYWD_VALUE FROM PASSK_OPEKEYLO TO PASSK_OPEKEYHI OF
349 0409 3 SET
350 0410 3
351 0411 3 [PASSK_DISPOSITION_SAVE]:
352 0412 4 BEGIN
353 0413 4 FCB [FCB$V_DELETE] = 0;
354 0414 4 FCB [FCB$V_PRINT] = 0;
355 0415 4 FCB [FCB$V_SUBMIT] = 0;
356 0416 4 FCB [FCB$V_SAVE] = 1;
357 0417 3 END;
358 0418 3
359 0419 3 [PASSK_DISPOSITION_DELETE]:
360 0420 4 BEGIN
361 0421 4 FCB [FCB$V_PRINT] = 0;
362 0422 4 FCB [FCB$V_SUBMIT] = 0;
363 0423 4 FCB [FCB$V_SAVE] = 0;
364 0424 4 FCB [FCB$V_DELETE] = 1;
365 0425 3 END;
366 0426 3
367 0427 3 [PASSK_DISPOSITION_PRINT]:
368 0428 4 BEGIN
369 0429 4 FCB [FCB$V_DELETE] = 0;
370 0430 4 FCB [FCB$V_SUBMIT] = 0;
371 0431 4 FCB [FCB$V_SAVE] = 1;
372 0432 4 FCB [FCB$V_PRINT] = 1;
373 0433 3 END;
374 0434 3
375 0435 3 [PASSK_DISPOSITION_PRINT_DELETE]:
376 0436 4 BEGIN
377 0437 4 FCB [FCB$V_SUBMIT] = 0;
378 0438 4 FCB [FCB$V_SAVE] = 0;
379 0439 4 FCB [FCB$V_PRINT] = 1;
380 0440 4 FCB [FCB$V_DELETE] = 1;
381 0441 3 END;
382 0442 3
383 0443 3 [PASSK_DISPOSITION_SUBMIT]:
384 0444 4 BEGIN
385 0445 4 FCB [FCB$V_PRINT] = 0;
386 0446 4 FCB [FCB$V_DELETE] = 0;
387 0447 4 FCB [FCB$V_SUBMIT] = 1;
388 0448 4 FCB [FCB$V_SAVE] = 1;
389 0449 3 END;
390 0450 3
391 0451 3 [PASSK_DISPOSITION_SUBMIT_DELETE]:
392 0452 4 BEGIN
393 0453 4 FCB [FCB$V_PRINT] = 0;
394 0454 4 FCB [FCB$V_SAVE] = 0;
395 0455 4 FCB [FCB$V_SUBMIT] = 1;
396 0456 4 FCB [FCB$V_DELETE] = 1;
397 0457 3 END;
398 0458 3
399 0459 3 [PASSK_USER_ACTION]:
400 0460 4 BEGIN
401 0461 4 USER_ACTION_BPV = .KEYWORDS [(I=.I+1)];
402 0462 3 END;
```

```

403 0463 3
404 0464 3
405 0465 4 [PASSK_ERROR]:
406 0466 4 BEGIN
407 0467 4 |
408 0468 4 | The ERROR parameter, if specified, was already processed
409 0469 4 | in PASS$CLOSE, so just ignore it here.
410 0470 4 |
411 0471 4 | I = .I + 1; ! Ignore next parameter
412 0472 4 | END;
413 0473 3 [INRANGE,OUTRANGE]:
414 0474 3 $PASSIO_ERROR (PASS_INVARGPAS,0);
415 0475 3
416 0476 3 TES;
417 0477 3
418 0478 2 END;
419 0479 2
420 0480 2 !+
421 0481 2 | Check for incompatible DISPOSITION.
422 0482 2 |
423 0483 2
424 0484 2 IF .FCB [FAB$V_TMD] AND
425 0485 2 (.FCB [FCB$V_SAVE] OR .FCB [FCB$V_SUBMIT] OR .FCB [FCB$V_PRINT])
426 0486 2 THEN
427 0487 2 $PASSIO_ERROR (PASS_FILNAMREQ,0); ! File name required
428 0488 2
429 0489 2 !+
430 0490 2 | Fill in fields in RMS control blocks
431 0491 2 |
432 0492 2
433 0493 2 IF .FCB [FCB$V_DELETE] AND NOT .FAB [FAB$V_TMD]
434 0494 2 THEN
435 0495 2 FAB [FAB$V_DLT] = 1;
436 0496 2
437 0497 2 IF .FCB [FCB$V_SUBMIT]
438 0498 2 THEN
439 0499 2 FAB [FAB$V_SCF] = 1;
440 0500 2
441 0501 2 IF .FCB [FCB$V_PRINT]
442 0502 2 THEN
443 0503 2 FAB [FAB$V_SPL] = 1;
444 0504 2
445 0505 2 !+
446 0506 2 | If a textfile, and if in Generation mode, and if the buffer
447 0507 2 | is not empty, do a WRITELN.
448 0508 2 |
449 0509 2
450 0510 2 IF .FCB [FCB$V_TEXT] AND .FCB [FCB$V_GENERATION] AND
451 0511 2 (.FCB [FCB$A_RECORD_CUR] GTRA .FCB [FCB$A_RECORD_BEG])
452 0512 2 THEN
453 0513 2 PASS$WRITELN (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
454 0514 2
455 0515 2 !+
456 0516 2 | Make sure the XAB chain doesn't point to garbage.
457 0517 2 |
458 0518 2
459 0519 2 FAB [FAB$L_XAB] = 0;

```

```

460 0520 2
461 0521 2
462 0522 2
463 0523 2
464 0524 2
465 0525 2
466 0526 2
467 0527 3
468 0528 3
469 0529 3
470 0530 4
471 0531 4
472 0532 4
473 0533 4
474 0534 4
475 0535 4
476 0536 4
477 0537 4
478 0538 4
479 0539 4
480 0540 3
481 0541 3
482 0542 3
483 0543 3
484 0544 3
485 0545 3
486 0546 3
487 0547 3
488 0548 3
489 0549 3
490 0550 3
491 0551 3
492 0552 3
493 0553 3
494 0554 3
495 0555 3
496 0556 3
497 0557 3
498 0558 2
499 0559 2
500 0560 2
501 0561 2
502 0562 2
503 0563 2
504 0564 2
505 0565 2
506 0566 2
507 0567 2
508 0568 2
509 0569 2
510 0570 2
511 0571 2
512 0572 2
513 0573 2
514 0574 2
515 0575 2
516 0576 2

!+
!- Close the file.
!-
IF .FAB [FAB$W_IFI] NEQ 0
THEN
  BEGIN
  IF .KEYWORDS_SEEN [PASS$USER_ACTION]
  THEN
    BEGIN
    LINKAGE
    USER_ACTION_LNK = CALL (REGISTER=1,STANDARD,STANDARD);

    STATUS = USER_ACTION_LNK (.USER_ACTION_BPV [0],      ! Address
                              .USER_ACTION_BPV [1],      ! Environment
                              FAB [0,0,0,0],            ! FAB address
                              RAB [0,0,0,0],            ! RAB address
                              PFV [PFV$R_PFV]);          ! PFV address
    END
  ELSE
    STATUS = $PASS$RMS_OP ($CLOSE (FAB=FAB [0,0,0,0]));

!+
!- Set the IFI to zero so we don't try another $CLOSE if the last one
!- failed.
!-
FAB [FAB$W_IFI] = 0;

!+
!- If the close failed, signal an error.
!-
IF NOT .STATUS
THEN
  $PASS$IO_ERROR (PASS$ERRDURCLO);

END;

!+
!- Deallocate the record buffer, if necessary.
!-
IF TESTBITSC (FCB [FCB$V_DYNAMIC_UBF])
THEN
  PASS$FREE_VM (.RAB [RAB$W_USZ], RAB [RAB$L_UBF]);

!+
!- Deallocate key types vector, if necessary.
!-
IF .FCB [FCB$A_KEY_TYPES] NEQ 0
THEN
  PASS$FREE_VM (.FCB [FCB$L_NKEYS]*4, FCB [FCB$A_KEY_TYPES]);

!+
!- Deallocate resultant name string, if dynamic. This should be the

```

```

: 517      0577 2      ! last thing to do.
: 518      0578 2      !-
: 519      0579 2
: 520      0580 2      IF TESTBITSC (FCB [FCB$V_DYNAMIC_RSN])
: 521      0581 2      THEN
: 522      0582 3      BEGIN
: 523      0583 3      LOCAL
: 524      0584 3      NAM: REF BLOCK [, BYTE];
: 525      0585 3      NAM = .FAB [FAB$L_NAM];
: 526      0586 3      PASS$FREE_VM (.NAM [NAM$B_RSL], NAM [NAM$L_RSA]);
: 527      0587 2      END;
: 528      0588 2
: 529      0589 2      !+
: 530      0590 2      ! Mark PFV to indicate file closed.
: 531      0591 2      !-
: 532      0592 2
: 533      0593 2      PFV [PFV$V_OPEN] = 0;
: 534      0594 2      PFV [PFV$V_VALID] = 0;      ! Invalidate flags
: 535      0595 2      PFV [PFV$V_DFB] = 0;      ! Invalidate file buffer
: 536      0596 2      PFV [PFV$V_EOF_DEFINED] = 0;
: 537      0597 2
: 538      0598 2      RETURN;
: 539      0599 2
: 540      0600 1      END;

```

! End of routine PASS\$CLOSE

```

.EXTRN PASSK_INVARGPAS
.EXTRN PASS$AB_KEYWD_NAME_TABLE
.EXTRN PASSK_FILNAMREQ
.EXTRN PASS$WRITELN, SYSSCLOSE
.EXTRN PASSK_ERRDURCLO
.EXTRN PASS$FREE_VM

```

```

                                OFFC 00000
                                .ENTRY PASS$CLOSE, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 0293
                                R10,R11
                                MOVAB PASS$SIGNAL, R11
                                MOVAB PASS$FREE_VM, R10
                                SUBL2 #8, SP
                                MOVL IN PFV, PFV ; 0370
                                MOVL 12(PFV), FCB ; 0376
                                CLRL KEYWORDS_SEEN ; 0382
                                BICW2 #511, KEYWORDS_SEEN+4 ; 0383
                                MOVZBL (AP), R4 ; 0390
                                SUBL2 #2, R4
                                MNEGL #1, I
                                BRW 15$
                                MOVL KEYWORDS[I], KEYWD_VALUE ; 0401
                                BLEQ 4$ ; 0402
                                CMPL KEYWD_VALUE, #40
                                BGTR 4$
                                MOVZBL PASS$AB_KEYWD_NAME_TABLE[KEYWD_VALUE], R0 ; 0406
                                BBSS R0, KEYWORDS_SEEN, -1$
                                CASEL KEYWD_VALUE, -#1, #39 ; 0408
                                .WORD 4$-3$,-
                                4$-3$,-
                                4$-3$,-

```


FF32

52 01
18 A7
09 FC A7
05
7E
6B
58
0A 68
05 48 A7
49 A7
05 A7
49 A7
49 A7
11
0D 68
E8 A7
00000000G 00
68 A7
46 A7
43 13
10 03 52
AE
56
0084 8F
04 A9
00 B9
03
00000000G 00
0D 50
0001825A 8F 50
04 12
E7 03 A8
51 50
46 A7
08 51
7E 00G 8F
6B 01
0A 68 13
24 A7
7E 20 A7
6A 02
CC A7
0B 13

0B 11 000EA
52 D6 000EC 13\$:
08 AC 42 D0 000EE
02 11 000F3
52 D6 000F5 14\$:
54 F1 000F7 15\$:
04 E1 000FD
05 E0 00102
FD A7 EB 00107
FC A7 95 0010B
0A 18 0010E
7E D4 00110 16\$:
00G 8F 9A 00112
02 FB 00116 17\$:
04 00119
FC A7 9E 0011A 18\$:
06 E1 0011E
04 E0 00122
80 8F 88 00127
01 AB E9 0012C 19\$:
40 8F 88 00130
68 95 00135 20\$:
04 18 00137
49 A7 20 88 00139
11 F8 A7 E9 0013D 21\$:
68 0C E1 00141
E8 A7 D1 00145
06 1B 0014A
00000000G 00 16 0014C
68 A7 D4 00152 22\$:
46 A7 B5 00155
43 13 00158
44 A7 9E 0015A
02 E1 0015E
56 DD 00163
0084 8F BB 00165
04 A9 D0 00169
03 FB 0016D
19 11 00171
52 DD 00173 23\$:
01 FB 00175
0D 50 E8 0017C
0001825A 8F 50 D1 0017F
04 12 00186
E7 03 A8 E8 00188
51 50 D0 0018C 24\$:
46 A7 B4 0018F
08 51 E8 00192
7E 00G 8F 9A 00195
6B 01 FB 00199
04 04 0019C
0A 68 13 E5 0019D 25\$:
24 A7 9F 001A1
7E 20 A7 3C 001A4
6A 02 FB 001A8
CC A7 D5 001AB 26\$:
0B 13 001AE

BRB 15\$
INCL I
MOVL KEYWORDS[I], USER_ACTION_BPV
BRB 15\$
INCL I
ACBL R4, #1, I, 2\$
BBC #4, 72(FAB), 18\$
BBS #5, -4(FAB), 16\$
BLBS -3(FAB), 16\$
TSTB -4(FAB)
BGEQ 18\$
CLRL -(SP)
MOVZBL #PASSK FILNAMREQ, -(SP)
CALLS #2, PASSSSIGNAL
RET
MOVAB -4(FAB), R8
BBC #6, (R8), 19\$
BBS #4, 72(FAB), 19\$
BISB2 #128, 73(FAB)
BLBC 1(R8), 20\$
BISB2 #64, 73(FAB)
TSTB (R8)
BGEQ 21\$
BISB2 #32, 73(FAB)
BLBC -8(FAB), 22\$
BBC #12, (R8), 22\$
CMPL -20(FAB), -24(FAB)
BLEQU 22\$
JSB PASS\$WRITELN
CLRL 104(FAB)
TSTW 70(FAB)
BEQL 25\$
MOVAB 68(R7), R2
BBC #2, KEYWORDS_SEEN+3, 23\$
PUSHL PFV
PUSHR #^M<R2,R7>
MOVL 4(USER_ACTION_BPV), R1
CALLS #3, @0(USER_ACTION_BPV)
BRB 24\$
PUSHL R2
CALLS #1, SYSSCLOSE
BLBS \$\$\$STATUS, 24\$
CMPL \$\$\$STATUS, #98906
BNEQ 24\$
BLBS 3(R8), 23\$
MOVL \$\$\$STATUS, STATUS
CLRW 70(FAB)
BLBS STATUS, 25\$
MOVZBL #PASSK ERRDURCLO, -(SP)
CALLS #1, PASSSSIGNAL
RET
BCC #19, (R8), 26\$
PUSHAB 36(FAB)
MOVZWL 32(FAB), -(SP)
CALLS #2, PASS\$FREE_VM
TSTL -52(FAB)
BEQL 27\$

0408
0461
0408
0470
0390
0484
0485
0487
0493
0495
0497
0499
0501
0503
0510
0511
0513
0519
0525
0536
0528
0538
0541
0548
0554
0556
0564
0566
0571

PASSCLOSE2
1-006

Close a file
PASS\$CLOSE - Close a file

I 15
16-Sep-1984 01:29:07
14-Sep-1984 12:51:21

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASCLOSE2.B32;1

Page 15
(4)

			CC	A7	9F	001B0		PUSHAB	-52(FAB)	:	0573
7E	D0	A7		02	78	001B3		ASHL	#2, -48(FAB), -(SP)	:	
		6A		02	FB	001B8		CALLS	#2, PASS\$FREE_VM	:	
0E		68		12	E5	001BB	27\$:	BBCC	#18, (R8), 28\$:	0580
		50		5C	A7	D0	001BF	MOVL	108(FAB), NAM	:	0585
		7E		04	A0	9F	001C3	PUSHAB	4(NAM)	:	0586
		6A		03	A0	9A	001C6	MOVZBL	3(NAM), -(SP)	:	
	06	A6		02	FB	001CA		CALLS	#2, PASS\$FREE_VM	:	
			2007	8F	AA	001CD	28\$:	BICW2	#8199, 6(PFV)	:	0596
				04	001D3			RET		:	0600

; Routine Size: 468 bytes, Routine Base: _PASS\$CODE + 008F

; 541 0601 1
; 542 0602 1 !<BLF/PAGE>

PASSCLOSE2
1-006

Close a file
PASS\$CLOSE - Close a file

J 15
16-Sep-1984 01:29:07 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:21 [PASRTL.SRC]PASCLOSE2.B32;1

Page 16
(5)

P
1

: 544 0603 1 END
: 545 0604 1
: 546 0605 0 ELUDOM

. End of module PASS\$CLOSE2

PSECT SUMMARY

Name	Bytes	Attributes
_PASS\$CODE	611	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	16	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	111	25	33	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASCLOSE2/OBJ=OBJ\$:PASCLOSE2 MSRC\$:PASCLOSE2/UPDATE=(ENH\$:PASCLOSE2)

: Size: 611 code + 0 data bytes
: Run Time: 00:14.1
: Elapsed Time: 00:51.1
: Lines/CPU Min: 2578
: Lexemes/CPU-Min: 19367
: Memory Used: 193 pages
: Compilation Complete

0293 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

