



```

PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR  IIIIII  CCCCCCCC  HH      HH      AAAAAA
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR  IIIIII  CCCCCCCC  HH      HH  AAAAAA
PP      PP  AA      AA  SS      WW      WW  RR      RR  II      CC      HH      HH  AA      AA
PP      PP  AA      AA  SS      WW      WW  RR      RR  II      CC      HH      HH  AA      AA
PP      PP  AA      AA  SS      WW      WW  RR      RR  II      CC      HH      HH  AA      AA
PP      PP  AA      AA  SS      WW      WW  RR      RR  II      CC      HH      HH  AA      AA
PPPPPPPP      AA      AA  SSSSSS  WW      WW  RRRRRRRR  II      CC      HHHHHHHHHH  AA      AA
PPPPPPPP      AA      AA  SSSSSS  WW      WW  RRRRRRRR  II      CC      HHHHHHHHHH  AA      AA
PP      AAAAAAAAAA      SS  WW  WW  WW  RR  RR  II      CC      HH      HH  AAAAAAAAAA
PP      AAAAAAAAAA      SS  WW  WW  WW  RR  RR  II      CC      HH      HH  AAAAAAAAAA
PP      AA      AA  SS  WWW  WWW  RR  RR  II      CC      HH      HH  AA      AA
PP      AA      AA  SS  WWW  WWW  RR  RR  II      CC      HH      HH  AA      AA
PP      AA      AA  SSSSSSSS  WW      WW  RR      RR  IIIIII  CCCCCCCC  HH      HH  AA      AA
PP      AA      AA  SSSSSSSS  WW      WW  RR      RR  IIIIII  CCCCCCCC  HH      HH  AA      AA

```

```

LL      IIIIII  SSSSSSSS
LL      .IIIII  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
LLLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLLL  IIIIII  SSSSSSSS

```

```

....
....
....
....

```

.....

```

1 0001 0 MODULE PASSWRITE_CHAR ( %TITLE 'Write a character'
2 0002 0 IDENT = '1-003' ! File: PASWRICHA.B32 Edit: DG1003
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 writes a character
36 0036 1 to a textfile.
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 - Make total-width a longword. SBL 29-Jun-1982
46 0046 1 1-003 - If field length is zero, don't write character. DG 21-Dec-1983
47 0047 1 --
48 0048 1

```

```
.. 50      0049 1 %SBTTL 'Declarations'  
.. 51      0050 1  
.. 52      0051 1 : PROLOGUE DEFINITIONS:  
.. 53      0052 1 :  
.. 54      0053 1  
.. 55      0054 1 REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures  
.. 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     PASSWRITE_CHAR: NOVALUE,           ! Write to textfile  
.. 63      0125 1     PASSWRITEV_CHAR: NOVALUE;         ! Write to string  
.. 64      0126 1  
.. 65      0127 1 :  
.. 66      0128 1 : MACROS:  
.. 67      0129 1 :  
.. 68      0130 1 :     NONE  
.. 69      0131 1 :  
.. 70      0132 1 : EQUATED SYMBOLS:  
.. 71      0133 1 :  
.. 72      0134 1 :     NONE  
.. 73      0135 1 :  
.. 74      0136 1 : FIELDS:  
.. 75      0137 1 :  
.. 76      0138 1 :     NONE  
.. 77      0139 1 :  
.. 78      0140 1 : OWN STORAGE:  
.. 79      0141 1 :  
.. 80      0142 1 :     NONE  
.. 81      0143 1 :
```

```

83 0144 1 %SBTTL 'PASSWRITE CHAR - Write character to textfile'
84 0145 1 GLOBAL ROUTINE PASSWRITE CHAR (
85 0146 1     PFV: REF $PASSPFV_FILE_VARIABLE,           ! File variable
86 0147 1     CHARACTER: BYTE,                         ! Character to write
87 0148 1     TOTAL_WIDTH: SIGNED,                    ! Total field width
88 0149 1     ERROR:                                ! Error unwind address
89 0150 1 ): NOVALUE =
90 0151 1
91 0152 1 ++
92 0153 1 FUNCTIONAL DESCRIPTION:
93 0154 1
94 0155 1     This procedure writes a single character to the specified textfile.
95 0156 1
96 0157 1 CALLING SEQUENCE:
97 0158 1
98 0159 1     CALL PASSWRITE_CHAR (PFV.mr.r, CHARACTER.rc.v, TOTAL_WIDTH.rl.v
99 0160 1     [, ERROR.j.r])
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 PASPFV.REQ.
105 0166 1
106 0167 1     CHARACTER      - The single character to write.
107 0168 1
108 0169 1     TOTAL_WIDTH     - Total field width.
109 0170 1
110 0171 1     ERROR          - Optional. If specified, the address to unwind to
111 0172 1     in case of an error.
112 0173 1
113 0174 1 IMPLICIT INPUTS:
114 0175 1
115 0176 1     NONE
116 0177 1
117 0178 1 IMPLICIT OUTPUTS:
118 0179 1
119 0180 1     NONE
120 0181 1
121 0182 1 ROUTINE VALUE:
122 0183 1
123 0184 1     NONE
124 0185 1
125 0186 1 SIDE EFFECTS:
126 0187 1
127 0188 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
128 0189 1
129 0190 1 SIGNALLED ERRORS:
130 0191 1
131 0192 1     LINTOOLON - Line too long
132 0193 1     NEGWIDDIG - Negative width or digits specification is not allowed
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 FIELD_WIDTH, ! Total width of field
141 0202 2 PFV_ADDR: VOLATILE, ! Enable argument
142 0203 2 UNWIND_ACT: VOLATILE, ! Enable argument
143 0204 2 ERROR_ADDR: VOLATILE; ! Enable argument
144 0205 2
145 0206 2 BUILTIN
146 0207 2 ACTUALCOUNT; ! Count of arguments
147 0208 2
148 0209 2 ENABLE
149 0210 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
150 0211 2
151 0212 2 !+
152 0213 2 ! Get ERROR parameter, if present.
153 0214 2 !-
154 0215 2
155 0216 2 IF ACTUALCOUNT () GEQU 4
156 0217 2 THEN
157 0218 2 ERROR_ADDR = .ERROR; ! Set unwind address
158 0219 2
159 0220 2 PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
160 0221 2
161 0222 2 !+
162 0223 2 ! Validate PFV and get PFV.
163 0224 2 !-
164 0225 2
165 0226 2 PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
166 0227 2
167 0228 2 !+
168 0229 2 ! Set unwind action to unlock file.
169 0230 2 !-
170 0231 2
171 0232 2 UNWIND_ACT = PASS$UNWIND_UNLOCK;
172 0233 2
173 0234 2 !+
174 0235 2 ! Do common initialization.
175 0236 2 !-
176 0237 2
177 0238 2 PASS$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
178 0239 2
179 0240 2 !+
180 0241 2 ! Check for invalid width.
181 0242 2 !-
182 0243 2
183 0244 2 IF .TOTAL_WIDTH LSS 0
184 0245 2 THEN
185 0246 2 $PASSIO_ERROR (PASS$_NEGWIDDIG,0);
186 0247 2
187 0248 2 !+
188 0249 2 ! See if we have enough room in the record.
189 0250 2 !-
190 0251 2
191 0252 2 FIELD_WIDTH = .TOTAL_WIDTH;
192 0253 2
193 0254 2 BEGIN
194 0255 2 LOCAL
195 0256 2 EXTRA; ! Extra characters past end of line
196 0257 2 EXTRA = (.FCB [FCB$_RECORD_CUR] + .FIELD_WIDTH) - .FCB [FCB$_RECORD_END];

```

```

197 0258 3 IF .EXTRA GTR 0
198 0259 3 THEN
199 0260 3 $PASSIO_ERROR (PASS_LINTOOLON,1,.EXTRA);
200 0261 3 END;
201 0262 3
202 0263 3 IF .FIELD_WIDTH - 1 GEQ 0
203 0264 3 THEN
204 0265 3 BEGIN
205 0266 3
206 0267 3 | +
207 0268 3 | Move leading blanks, if any
208 0269 3 | -
209 0270 3
210 0271 3 FCB [FCB$A_RECORD_CUR] = CH$FILL (' ', .FIELD_WIDTH - 1, .FCB [FCB$A_RECORD_CUR]);
211 0272 3
212 0273 3 | +
213 0274 3 | Move character to buffer
214 0275 3 | -
215 0276 3
216 0277 3 CH$WCHAR_A (.CHARACTER, FCB [FCB$A_RECORD_CUR]);
217 0278 3
218 0279 3 END;
219 0280 3
220 0281 3 | +
221 0282 3 | Call WRITE epilogue routine to move the last character written to the
222 0283 3 | user's buffer and to unlock the file variable.
223 0284 3 | -
224 0285 3
225 0286 3 PASS$END_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
226 0287 3
227 0288 3 RETURN;
228 0289 3
229 0290 1 END;

```

! End of routine PASSWRITE\_CHAR

```

.TITLE PASSWRITE_CHAR Write a character
.IDENT \1-003\

.EXTRN PASSWRITE_CHAR, PASSWRITEV_CHAR
.EXTRN PASS$IO_HANDLER
.EXTRN PASS$VACIDATE_PFV
.EXTRN PASS$INIT_WRITE
.EXTRN PASS$SIGNAL, PASSK_NEGWIDDIG
.EXTRN PASSK_LINTOOLON
.EXTRN PASS$END_WRITE

```

```

.PSECT _PASSCODE, NOWRT, SHR, PIC, 2

```

```

01FC 0000
58 0000000G 00 9E 00002
5E 08 C2 00009
7E D4 0000C
04 AE 7C 0000E
6D 006C CF DE 00011
04 6C 91 00016
04 1F 00019
6E 10 AC D0 0001B

```

```

.ENTRY PASSWRITE_CHAR, Save R2,R3,R4,R5,R6,R7,R8 : 0145
MOVAB PASS$SIGNAL, R8
SUBL2 #8, SP
CLRL ERROR_ADDR : 0197
CLRQ UNWIND_ACT
MOVAL 5$, (FP)
CMPB (AP), #4 : 0216
BLSSU 1$
MOVL ERROR, ERROR_ADDR : 0218

```

PASSWRITE\_CHAR  
1-003

Write a character  
PASSWRITE\_CHAR - Write character to textfile

G 13  
16-Sep-1984 02:16:00  
14-Sep-1984 12:52:04

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

Page 6  
(3)

	56	04	AC	D0	0001F	1\$:	MOVL	PFV, R6		0220
08	AE		56	D0	00023		MOVL	R6, PFV_ADDR		
		00000000G	00	16	00027		JSB	PASS\$VACIDATE PFV		0226
04	AE		01	D0	0002D		MOVL	#1, UNWIND_ACT		0232
		00000000G	00	16	00031		JSB	PASS\$INIT_WRITE		0238
			0C	AC	D5	00037	TSTL	TOTAL_WIDTH		0244
			0A	18	0003A		BGEQ	2\$		
			7E	D4	0003C		CLRL	-(SP)		0246
	7E	00G	8F	9A	0003E		MOVZBL	#PASSK_NEGWIDDIG, -(SP)		
	68		02	FB	00042		CALLS	#2, PASS\$SIGNAL		
				04	00045		RET			
	52	0C	AC	D0	00046	2\$:	MOVL	TOTAL_WIDTH, FIELD_WIDTH		0252
50	52	EC	A7	C1	0004A		ADDL3	-20(FCB), FIELD_WIDTH, R0		0257
	50	FO	A7	C2	0004F		SUBL2	-16(FCB), EXTRA		
			0C	15	00053		BLEQ	3\$		0258
			50	DD	00055		PUSHL	EXTRA		0260
			01	DD	00057		PUSHL	#1		
	7E	00G	8F	9A	00059		MOVZBL	#PASSK_LINTOOLON, -(SP)		
	68		03	FB	0005D		CALLS	#3, PASS\$SIGNAL		
				04	00060		RET			
			52	D5	00061	3\$:	TSTL	FIELD_WIDTH		0263
			15	15	00063		BLEQ	4\$		
			52	D7	00065		DECL	R2		0271
52		20	6E	00	2C	00067	MOVCS	#0, (SP), #32, R2, @-20(FCB)		
			EC	B7		0006C				
	EC		A7	53	D0	0006E	MOVL	R3, -20(FCB)		
	EC		B7	08	AC	90	MOVB	CHARACTER, @-20(FCB)		0277
			EC	A7	D6	00077	INCL	-20(FCB)		
				00	16	0007A	4\$:	JSB	PASS\$END_WRITE	0286
				04	00080		RET			0290
				0000	00081	5\$:	.WORD	Save nothing		0197
	50		08	AC	D0	00083	MOVL	8(AP), R0		
	50		04	A0	D0	00087	MOVL	4(R0), R0		
			F4	A0	9F	0008B	PUSHAB	ERROR_ADDR		
			F8	A0	9F	0008E	PUSHAB	UNWIND_ACT		
			FC	A0	9F	00091	PUSHAB	PFV_ADDR		
				03	DD	00094	PUSHL	#3		
			5E	DD	00096		PUSHL	SP		
			7E	04	AC	7D	MOVQ	4(AP), -(SP)		
	00000000G		00	03	FB	0009C	CALLS	#3, PASS\$IO_HANDLER		
				04	000A3		RET			

; Routine Size: 164 bytes, Routine Base: \_PASS\$CODE + 0000

; 230 0291 1  
; 231 0292 1 !<BLF/PAGE>



```

233 0293 1 %SBTTL 'PASSWRITEV_CHAR - Write character to string'
234 0294 1 GLOBAL ROUTINE PASSWRITEV_CHAR (
235 0295 1     MAX_LENGTH: WORD,           ! Maximum length of string
236 0296 1     STRING_LINE: REF VECTOR [, WORD], ! String to write to
237 0297 1     VALUE,                 ! Value to write
238 0298 1     TOTAL_WIDTH: SIGNED,    ! Total field width
239 0299 1     ERROR                 ! Error unwind address
240 0300 1 ) : NOVALUE =
241 0301 1
242 0302 1 +-
243 0303 1 FUNCTIONAL DESCRIPTION:
244 0304 1
245 0305 1     This procedure writes a character to the specified string.
246 0306 1
247 0307 1 CALLING SEQUENCE:
248 0308 1
249 0309 1     CALL PASSWRITEV_CHAR (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
250 0310 1     VALUE.rbu.v, TOTAL_WIDTH.rl.v [, ERROR.j.r])
251 0311 1
252 0312 1 FORMAL PARAMETERS:
253 0313 1
254 0314 1     MAX_LENGTH      - The maximum length of STRING_LINE.
255 0315 1
256 0316 1     STRING_LINE    - A varying string to which the output will be appended.
257 0317 1
258 0318 1     VALUE          - The value to write.
259 0319 1
260 0320 1     TOTAL_WIDTH    - The width of the field to write.
261 0321 1
262 0322 1     ERROR          - Optional. If specified, the address to unwind to
263 0323 1     in case of an error.
264 0324 1
265 0325 1 IMPLICIT INPUTS:
266 0326 1
267 0327 1     NONE
268 0328 1
269 0329 1 IMPLICIT OUTPUTS:
270 0330 1
271 0331 1     NONE
272 0332 1
273 0333 1 ROUTINE VALUE:
274 0334 1
275 0335 1     NONE
276 0336 1
277 0337 1 SIDE EFFECTS:
278 0338 1
279 0339 1     NONE
280 0340 1
281 0341 1 SIGNALLED ERRORS:
282 0342 1
283 0343 1     See PASSWRITE_CHAR
284 0344 1
285 0345 1 --
286 0346 1
287 0347 2 BEGIN
288 0348 2
289 0349 2 LOCAL

```

PASSWRITE\_CHAR Write a character  
 1-003 PASSWRITEV\_CHAR - Write character to string

1 13  
 16-Sep-1984 02:16:00  
 14-Sep-1984 12:52:04

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

Page 8  
 (4)

```

: 290 0350 2 PFV: $PASSPFV FILE VARIABLE, ! Pascal File Variable
: 291 0351 2 ARG_LIST: VECTOR [4, LONG], ! Argument list
: 292 0352 2 PFV_ADDR: VOLATILE, ! Enable argument
: 293 0353 2 UNWIND_ACT: VOLATILE, ! Enable argument
: 294 0354 2 ERROR_ADDR: VOLATILE; ! Enable argument
: 295 0355 2
: 296 0356 2 BUILTIN
: 297 0357 2 ACTUALCOUNT; ! Count of arguments
: 298 0358 2
: 299 0359 2 ENABLE
: 300 0360 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
: 301 0361 2
: 302 0362 2 !+
: 303 0363 2 ! Get ERROR parameter, if present.
: 304 0364 2 !-
: 305 0365 2
: 306 0366 2 IF ACTUALCOUNT () GEQU 5
: 307 0367 2 THEN
: 308 0368 2 ERROR_ADDR = .ERROR; ! Set unwind address
: 309 0369 2
: 310 0370 2 PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
: 311 0371 2
: 312 0372 2 !+
: 313 0373 2 ! Set up ARG_LIST.
: 314 0374 2 !-
: 315 0375 2
: 316 0376 2 ARG_LIST [0] = 3; ! Three arguments
: 317 0377 2 ARG_LIST [1] = PFV [PFV$R_PFV]; ! PFV address
: 318 0378 2 ARG_LIST [2] = .VALUE; ! Value to write
: 319 0379 2 ARG_LIST [3] = .TOTAL_WIDTH;
: 320 0380 2
: 321 0381 2 !+
: 322 0382 2 ! Call PASS$DO_WRITEV to do the work, giving it the address of
: 323 0383 2 ! PASSWRITE_CHAR to call.
: 324 0384 2 !-
: 325 0385 2
: 326 0386 2 PASS$DO_WRITEV (PFV [PFV$R_PFV], .MAX_LENGTH, STRING_LINE [0], ARG_LIST,
: 327 0387 2 PASSWRITE_CHAR);
: 328 0388 2
: 329 0389 2 RETURN;
: 330 0390 2
: 331 0391 1 END; ! End of routine PASSWRITEV_CHAR

```

.EXTRN PASS\$DO\_WRITEV

```

          007C 0000 .ENTRY PASSWRITEV_CHAR, Save R2,R3,R4,R5,R6 : 0294
5E        28 C2 0002  SUBL2 #40, SP :
          7E D4 0005  CLRL ERROR_ADDR : 0347
          04 AE 7C 0007  CLRQ UNWIND_ACT
6D        0439 CF DE 0000A MOVAL 2$, (FP)
05        6C 91 0000F CMPB (AP), #5 : 0366
          04 1F 00012 BLSSU 1$
          6E 14 AC D0 00014 MOVL ERROR, ERROR_ADDR : 0368
08        AE 1C AE 9E 00018 1$: MOVAB PFV, PFV_ADDR : 0370
0C        AE 03 D0 0001D MOVL #3, ARG_LIST : 0376

```

PASSWRITE\_CHAR Write a character  
 1-003 PASSWRITEV\_CHAR - Write character to string

J 13  
 16-Sep-1984 02:16:00  
 14-Sep-1984 12:52:04

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

Page 9  
 (4)

10	AE	1C	AE	9E	00021	MOVAB	PFV, ARG_LIST+4	:	0377
14	AE	0C	AC	7D	00026	MOVQ	VALUE, ARG_LIST+8	:	0378
	55	FF2D	CF	9E	0002B	MOVAB	PASSWRITE_CHAR, R5	:	0386
	54	0C	AE	9E	00030	MOVAB	ARG_LIST, -R4	:	
	56	1C	AE	9E	00034	MOVAB	PFV, R6	:	
	53	08	AC	D0	00038	MOVL	STRING_LINE, R3	:	
	52	04	AC	3C	0003C	MOVZWL	MAX_LENGTH, R2	:	
		00000000G	00	16	00040	JSB	PASS\$DO_WRITEV	:	
				04	00046	RET		:	0391
				0000	00047	.WORD	Save nothing	:	0347
	50	08	AC	D0	00049	MOVL	8(AP), R0	:	
	50	04	A0	D0	0004D	MOVL	4(R0), R0	:	
		D4	A0	9F	00051	PUSHAB	ERROR_ADDR	:	
		D8	A0	9F	00054	PUSHAB	UNWIND_ACT	:	
		DC	A0	9F	00057	PUSHAB	PFV_ADDR	:	
				03	DD	PUSHL	#3	:	
				5E	DD	PUSHL	SP	:	
	7E	04	AC	7D	0005E	MOVQ	4(AP), -(SP)	:	
	00000000G	00	03	FB	00062	CALLS	#3, PASS\$IO_HANDLER	:	
				04	00069	RET		:	

: Routine Size: 106 bytes, Routine Base: \_PASSCODE + 00A4

: 332 0392 1  
 : 333 0393 1 !<BLF/PAGE>

PASSWRITE\_CHAR Write a character  
1-003 PASSWRITEV\_CHAR - Write character to string

K 13  
16-Sep-1984 02:16:00 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:52:04 [PASRTL.SRC]PASWRICHA.B32;1

Page 10  
(5)

: 335 0394 1 END  
: 336 0395 1  
: 337 0396 0 ELUDOM

! End of module PASSWRITE\_CHAR

PSECT SUMMARY

Name	Bytes	Attributes
_PASSCODE	270	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]TARLET.L32;1	9776	0 0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	96 22	33	00:00.4

COMMAND QUALIFIERS

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

: 338 0397 0  
: Size: 270 code + 0 data bytes  
: Run Time: 00:07.2  
: Elapsed Time: 00:22.9  
: Lines/CPU Min: 3312  
: Lexemes/CPU-Min: 12884  
: Memory Used: 84 pages  
: Compilation Complete



