

```

PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPPPP    AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPPPP    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
PPPPPPPPPPPPPP    AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT              LLL
PPPPPPPPPPPPPP    AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT              LLL
PPPPPPPPPPPPPP    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          SSS              RRR              RRR              TTT              LLL
PPP              AAA          AAA          SSSSSSSSSSSS      RRR              RRR              TTT              LLLLLLLLLLLLLLLLLL
PPP              AAA          AAA          SSSSSSSSSSSS      RRR              RRR              TTT              LLLLLLLLLLLLLLLLLL
PPP              AAA          AAA          SSSSSSSSSSSS      RRR              RRR              TTT              LLLLLLLLLLLLLLLLLL

```

—\$2

Sym

PAS

PAS
PASPAS
PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS
PAS

PAS

100

PAS

PAS

143

PAS

PAS

PAS

PAS

PAS
PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS

PAS
PAS

PAS

```
PPPPPPPP      AAAAAA      SSSSSSSS      WW      WW      RRRRRRRR      IIIIII      BBBB88888      000000      000000
PPPPPPPP      AAAAAA      SSSSSSSS      WW      WW      RRRRRRRR      IIIIII      BBBB88888      000000
PP      PP      AA      AA      SS      WW      WW      RR      RR      BB      BB      00      00      00      00
PP      PP      AA      AA      SS      WW      WW      RR      RR      BB      BB      00      00      00      00
PP      PP      AA      AA      SS      WW      WW      RR      RR      BB      BB      00      00      00      00
PP      PP      AA      AA      SS      WW      WW      RR      RR      BB      BB      00      00      00      00
PPPPPPPP      AA      AA      SSSSSS      WW      WW      RRRRRRRR      II      BB8888888      00      00      00      00
PPPPPPPP      AA      AA      SSSSSS      WW      WW      RRRRRRRR      II      BB8888888      00      00      00      00
PP      AAAAAAAAAA      SS      WW      WW      RR      RR      BB      BB      00      00      00      00
PP      AAAAAAAAAA      SS      WW      WW      RR      RR      BB      BB      00      00      00      00
PP      AA      AA      SS      WWW      WWW      RR      RR      BB      BB      00      00      00      00
PP      AA      AA      SS      WWW      WWW      RR      RR      BB      BB      00      00      00      00
PP      AA      AA      SSSSSSSS      WW      WW      RR      RR      BB      BB      00      00      00      00
PP      AA      AA      SSSSSSSS      WW      WW      RR      RR      BB      BB      00      00      00      00
                                     ....
                                     ....
                                     ....
                                     ....

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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
```



```
1 0001 0 MODULE PASSWRITE_BOOLEAN ( %TITLE 'Write a boolean value'
2 0002 0 IDENT = '1-002' ! File: PASWRIB00.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 writes a boolean value
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 --
47 0047 1
```

PASSWRITE_BOOLE Write a boolean value
1-002 Declarations

F 12
16-Sep-1984 02:15:17
14-Sep-1984 12:52:02

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASWRIB00.B32;1

Page 2
(2)

```
: 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';
: 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     PASSWRITE_BOOLEAN: NOVALUE,
: 62      0124 1     PASSWRITEV_BOOLEAN: NOVALUE;
: 63      0125 1
: 64      0126 1
: 65      0127 1 MACROS:
: 66      0128 1
: 67      0129 1     NONE
: 68      0130 1
: 69      0131 1 EQUATED SYMBOLS:
: 70      0132 1
: 71      0133 1     NONE
: 72      0134 1
: 73      0135 1 FIELDS:
: 74      0136 1
: 75      0137 1     NONE
: 76      0138 1
: 77      0139 1 OWN STORAGE:
: 78      0140 1
: 79      0141 1     NONE
: 80      0142 1
```

! Externals, linkages, PSECTs, structures

! Write to textfile
! Write to string


```

82      0143 1 %SBTTL 'PASSWRITE_BOOLEAN - Write a boolean to textfile'
83      0144 1 GLOBAL ROUTINE PASSWRITE_BOOLEAN (
84      0145 1     PFV: REF $PASSPFV FILE_VARIABLE,
85      0146 1     BOOLEAN: BITVECTOR [1],
86      0147 1     TOTAL_WIDTH: SIGNED,
87      0148 1     ERROR
88      0149 1 ): NOVALUE =
89      0150 1
90      0151 1 ++
91      0152 1 FUNCTIONAL DESCRIPTION:
92      0153 1
93      0154 1     This procedure writes a boolean value to the specified textfile.
94      0155 1
95      0156 1 CALLING SEQUENCE:
96      0157 1
97      0158 1     CALL PASSWRITE_BOOLEAN (PFV.mr.r, BOOLEAN.rv.v, TOTAL_WIDTH.rl.v
98      0159 1     [ERROR.j.r])
99      0160 1
100     0161 1 FORMAL PARAMETERS:
101     0162 1
102     0163 1     PFV          - The Pascal File Variable (PFV) passed by reference.
103     0164 1     The structure of the PFV is defined in PASPFV.REQ.
104     0165 1
105     0166 1     BOOLEAN      - The boolean value to write.
106     0167 1
107     0168 1     TOTAL_WIDTH  - Total field width.
108     0169 1
109     0170 1     ERROR        - Optional. Address to unwind to if an error occurs.
110     0171 1
111     0172 1 IMPLICIT INPUTS:
112     0173 1
113     0174 1     NONE
114     0175 1
115     0176 1 IMPLICIT OUTPUTS:
116     0177 1
117     0178 1     NONE
118     0179 1
119     0180 1 ROUTINE VALUE:
120     0181 1
121     0182 1     NONE
122     0183 1
123     0184 1 SIDE EFFECTS:
124     0185 1
125     0186 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
126     0187 1
127     0188 1 SIGNALLED ERRORS:
128     0189 1
129     0190 1     See PASSWRITE_ENUMERATED
130     0191 1
131     0192 1 --
132     0193 1
133     0194 2 BEGIN
134     0195 2
135     0196 2 LOCAL
136     0197 2     PFV_ADDR: VOLATILE,
137     0198 2     UNWIND_ACT: VOLATILE,
138     0199 2     ERROR_ADDR: VOLATILE;

```

! File variable
 ! Value to write
 ! Total field width
 ! Error unwind address

! Enable argument
 ! Enable argument
 ! Enable argument

```

: 139      0200  2
: 140      0201  2
: 141      0202  2      BUILTIN
: 142      0203  2      ACTUALCOUNT;
: 143      0204  2
: 144      0205  2      ENABLE
: 145      0206  2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);      ! Enable error handler
: 146      0207  2
: 147      0208  2      !+
: 148      0209  2      !- Get ERROR parameter, if present.
: 149      0210  2
: 150      0211  2      IF ACTUALCOUNT () GEQU 4
: 151      0212  2      THEN
: 152      0213  2      ERROR_ADDR = .ERROR;      ! Set unwind address
: 153      0214  2
: 154      0215  2      PFV_ADDR = PFV [PFV$R_PFV];      ! Set PFV address
: 155      0216  2
: 156      0217  2      !+
: 157      0218  2      !- Call PASSWRITE_ENUMERATED to do the work.
: 158      0219  2
: 159      0220  2
: 160      0221  2      PASSWRITE_ENUMERATED (
: 161      0222  2      PFV [PFV$R_PFV],      ! File variable
: 162      0223  2      PASS$GR_BOOLEAN_PETD,      ! Enumerated type descriptor for type BOOLEAN
: 163      0224  2      .BOOLEAN [0],      ! Value to write
: 164      0225  2      .TOTAL_WIDTH);      ! Field width
: 165      0226  2
: 166      0227  2      RETURN;
: 167      0228  2
: 168      0229  1      END;      ! End of routine PASSWRITE_BOOLEAN

```

```

.TITLE PASSWRITE_BOOLEAN Write a boolean value
.IDENT \1-002\

```

```

.EXTRN PASSWRITE_BOOLEAN
.EXTRN PASSWRITE_BOOLEAN
.EXTRN PASS$IO_HANDLER
.EXTRN PASSWRITE_ENUMERATED
.EXTRN PASS$GR_BOOLEAN_PETD

```

```

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

```

```

.ENTRY PASSWRITE_BOOLEAN, Save nothing
SUBL2 #8, SP
CLRL ERROR_ADDR
CLRQ UNWIND_ACT
MOVAL 2$, (FP)
CMPB (AP), #4
BLSSU 1$
MOVL ERROR, ERROR_ADDR
MOVL PFV, PFV_ADDR
PUSHL TOTAL_WIDTH
EXTZV #0, #T, BOOLEAN, -(SP)
PUSHAB PASS$GR_BOOLEAN_PETD
PUSHL PFV
CALLS #4, PASSWRITE_ENUMERATED

```

```

                                0000 00000
                                08 C2 00002
                                7E D4 00005
                                04 AE 7C 00007
                                6D 0029 CF DE 0000A
                                04 6C 91 0000F
                                04 04 1F 00012
                                08 6E 10 AC D0 00014
                                AE 04 AC D0 00018 1$:
                                0C AC DD 0001D
                                7E 08 AC 01 00000000G
                                00 00000000G 04 00 00000000G
                                04 AC DD 0002C
                                04 FB 0002F

```

```

: 0144
:
: 0194
:
: 0211
:
: 0213
: 0215
: 0225
: 0224
: 0222
:

```



```
172 0232 1 %SBTTL 'PAS$WRITEV_BOOLEAN - Write BOOLEAN to string'
173 0233 1 GLOBAL ROUTINE PAS$WRITEV_BOOLEAN (
174 0234 1     MAX_LENGTH: WORD,
175 0235 1     STRING_LINE: REF VECTOR [, WORD],
176 0236 1     BOOLEAN,
177 0237 1     TOTAL_WIDTH: SIGNED,
178 0238 1     ERROR
179 0239 1 ) : NOVALUE =
180 0240 1
181 0241 1 ++
182 0242 1 FUNCTIONAL DESCRIPTION:
183 0243 1
184 0244 1     This procedure writes a BOOLEAN to the specified string.
185 0245 1
186 0246 1 CALLING SEQUENCE:
187 0247 1
188 0248 1     CALL PAS$WRITEV_BOOLEAN (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
189 0249 1     BOOLEAN.rv.v, TOTAL_WIDTH.rl.v [, ERROR.j.r])
190 0250 1
191 0251 1 FORMAL PARAMETERS:
192 0252 1
193 0253 1     MAX_LENGTH      - The maximum length of STRING_LINE.
194 0254 1
195 0255 1     STRING_LINE    - A varying string to which the output will be appended.
196 0256 1
197 0257 1     BOOLEAN        - The value to write.
198 0258 1
199 0259 1     TOTAL_WIDTH    - The width of the field to write.
200 0260 1
201 0261 1     ERROR          - Optional.  If specified, the address to unwind to
202 0262 1                    in case of an error.
203 0263 1
204 0264 1 IMPLICIT INPUTS:
205 0265 1
206 0266 1     NONE
207 0267 1
208 0268 1 IMPLICIT OUTPUTS:
209 0269 1
210 0270 1     NONE
211 0271 1
212 0272 1 ROUTINE VALUE:
213 0273 1
214 0274 1     NONE
215 0275 1
216 0276 1 SIDE EFFECTS:
217 0277 1
218 0278 1     NONE
219 0279 1
220 0280 1 SIGNALLED ERRORS:
221 0281 1
222 0282 1     See PAS$WRITE_BOOLEAN
223 0283 1
224 0284 1 --
225 0285 1
226 0286 2 BEGIN
227 0287 2
228 0288 2 LOCAL
```



```

229 0289 2 PFV: $PASSPFV FILE VARIABLE, ! Pascal File Variable
230 0290 2 ARG_LIST: VECTOR [4, LONG], ! Argument list
231 0291 2 PFV_ADDR: VOLATILE, ! Enable argument
232 0292 2 UNWIND_ACT: VOLATILE, ! Enable argument
233 0293 2 ERROR_ADDR: VOLATILE; ! Enable argument
234 0294 2
235 0295 2 BUILTIN
236 0296 2 ACTUALCOUNT; ! Count of arguments
237 0297 2
238 0298 2 ENABLE
239 0299 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
240 0300 2
241 0301 2 !+
242 0302 2 ! Get ERROR parameter, if present.
243 0303 2 !-
244 0304 2
245 0305 2 IF ACTUALCOUNT () GEQU 5
246 0306 2 THEN
247 0307 2 ERROR_ADDR = .ERROR; ! Set unwind address
248 0308 2
249 0309 2 PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
250 0310 2
251 0311 2 !+
252 0312 2 ! Set up ARG_LIST.
253 0313 2 !-
254 0314 2
255 0315 2 ARG_LIST [0] = 3; ! Three arguments
256 0316 2 ARG_LIST [1] = PFV [PFV$R_PFV]; ! PFV address
257 0317 2 ARG_LIST [2] = .BOOLEAN; ! Value to write
258 0318 2 ARG_LIST [3] = .TOTAL_WIDTH; ! Field width
259 0319 2
260 0320 2 !+
261 0321 2 ! Call PASS$DO WRITEV to do the work, giving it the address of
262 0322 2 ! PASSWRITE_BOOLEAN to call.
263 0323 2 !-
264 0324 2
265 0325 2 PASS$DO WRITEV (PFV [PFV$R_PFV], .MAX_LENGTH, STRING_LINE [0], ARG_LIST,
266 0326 2 PASSWRITE_BOOLEAN);
267 0327 2
268 0328 2 RETURN;
269 0329 2
270 0330 1 END; ! End of routine PASSWRITEV_BOOLEAN

```

				.EXTRN	PASS\$DO_WRITEV	
5E		007C	00000	.ENTRY	PASSWRITEV_BOOLEAN, Save R2,R3,R4,R5,R6	: 0233
		28	C2 00002	SUBL2	#40, SP	: 0286
		7E	D4 00005	CLRL	ERROR_ADDR	
	04	AE	7C 00007	CLRQ	UNWIND_ACT	
6D	0039	CF	DE 0000A	MOVAL	2\$, (FP)	
05		6C	91 0000F	CMPB	(AP), #5	: 0305
		04	1F 00012	BLSSU	1\$	
6E	14	AC	D0 00014	MOVL	ERROR, ERROR_ADDR	: 0307
08	AE	AE	9E 00018 1\$:	MOVAB	PFV, PFV_ADDR	: 0309
0C	AE	03	D0 0001D	MOVL	#3, ARG_LIST	: 0315

1-002

PASSWRITEV_BOOLEAN - Write BOOLEAN to string

L 12

16-Sep-1984 02:15:17

14-Sep-1984 12:52:02

VAX-11 Bliss-32 v4.0-742

[PASRTL.SRC]PASWRIBOO.B32:1

Page 8

(4)

10	AE	1C	AE	9E	00021
14	AE	0C	AC	7D	00026
	55	FF77	CF	9E	0002B
	54	0C	AE	9E	00030
	56	1C	AE	9E	00034
	53	08	AC	D0	00038
	52	04	AC	3C	0003C
		00000000G	00	16	00040
				04	00046
				0000	00047
	50	08	AC	D0	00049
	50	04	A0	D0	0004D
		D4	A0	9F	00051
		D8	A0	9F	00054
		DC	A0	9F	00057
			03	DD	0005A
			5E	DD	0005C
	7E	04	AC	7D	0005E
00000000G	00		03	FB	00062
				04	00069

28:

```

MOVAB    PFV, ARG_LIST+4
MOVQ     BOOLEAN, ARG_LIST+8
MOVAB    PASS$WRITE_BOOLEAN, R5
MOVAB    ARG_LIST, R4
MOVAB    PFV, R6
MOVL     STRING_LINE, R3
MOVZWL   MAX_LENGTH, R2
JSB      PASS$DO_WRITEV
RET

.word    Save_nothing
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

```

0316
0317
0325

0330
0286

; Routine Size: 106 bytes, Routine Base: _PASS\$CODE + 005A

```

; 271          0331 1
; 272          0332 1 !<BLF/PAGE>

```


PASSWRITE_BOOLE Write a boolean value
1-002 PASSWRITEV_BOOLEAN - Write BOOLEAN to string

M 12
16-Sep-1984 02:15:17 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:52:02 [PASRTL.SRC]PASWRIBOO.B32;1

Page 9
(5)

: 274 0333 1 END
: 275 0334 1
: 276 0335 0 ELUDOM

! End of module PASSWRITE_BOOLEAN

PSECT SUMMARY

Name	Bytes	Attributes
_PASSCODE	196	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	26	6	33	00:00.4

COMMAND QUALIFIERS

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

: 277 0336 0
: Size: 196 code + 0 data bytes
: Run Time: 00:05.5
: Elapsed Time: 00:20.0
: Lines/CPU Min: 3645
: Lexemes/CPU-Min: 9254
: Memory Used: 50 pages
: Compilation Complete

0297

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY