


```

PPPPPPPP      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEE      AAAAAA      CCCCCCCC      HH      HH      AAAAAA
PPPPPPPP      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEE      AAAAAA      CCCCCCCC      HH      HH      AAAAAA
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      CC      HH      HH      AA      AA
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      CC      HH      HH      AA      AA
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      CC      HH      HH      AA      AA
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      CC      HH      HH      AA      AA
PPPPPPPP      AA      AA      SSSSSS      RRRRRRRR      EEEEEEEEEE      AA      AA      CC      HHHHHHHHHH      AA      AA
PPPPPPPP      AA      AA      SSSSSS      RRRRRRRR      EEEEEEEEEE      AA      AA      CC      HHHHHHHHHH      AA      AA
PP      AAAAAAAAAA      SS      RR      RR      EE      AAAAAAAAAA      CC      HH      HH      AAAAAAAAAA
PP      AAAAAAAAAA      SS      RR      RR      EE      AAAAAAAAAA      CC      HH      HH      AAAAAAAAAA
PP      AA      AA      SS      RR      RR      EE      AA      AA      CC      HH      HH      AA      AA
PP      AA      AA      SS      RR      RR      EE      AA      AA      CC      HH      HH      AA      AA
PP      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEE      AA      AA      CCCCCCCC      HH      HH      AA      AA
PP      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEE      AA      AA      CCCCCCCC      HH      HH      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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

```

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

```

.....

```

1 0001 0 MODULE PASS$READ_CHAR ( %TITLE 'Read a character'
2 0002 0 IDENT = '1-002' ! File. PASREACHA.B32 Edit: SBL1002
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains procedures to read a character from a
36 0036 1 textfile and a string.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SBL 1-April-1981
45 0045 1 1-002 - Use PASS$END_READ. SBL 26-May-1982
46 0046 1 --
47 0047 1

```

PASSREAD_CHAR
1-002

Read a character
Declarations

N 2
16-Sep-1984 01:53:40
14-Sep-1984 12:51:46

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

Page 2
(?)

PA
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     PASSREAD_CHAR,                 ! Read from textfile
62      0124 1     PASSREADV_CHAR;                ! Read from string
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 |
```

```

82 0143 1 %SBTTL 'PASS$READ_CHAR - Read a character from textfile'
83 0144 1 GLOBAL ROUTINE PASS$PEAD_CHAR (
84 0145 1     PFV: REF $PASS$PFV_FILE_VARIABLE,           ! File variable
85 0146 1     ERROR                                     ! Error unwind address
86 0147 1     ) =
87 0148 1
88 0149 1 ++
89 0150 1 FUNCTIONAL DESCRIPTION:
90 0151 1
91 0152 1     This function reads a single character from the specified textfile
92 0153 1     and returns that character as its function value.
93 0154 1
94 0155 1 CALLING SEQUENCE:
95 0156 1
96 0157 1     char.wc.v = PASS$READ_CHAR (PFV.mr.r [, ERROR.j.r])
97 0158 1
98 0159 1 FORMAL PARAMETERS:
99 0160 1
100 0161 1     PFV           - The Pascal File Variable (PFV) passed by reference.
101 0162 1                 The structure of the PFV is defined in PAS$PFV.REQ.
102 0163 1
103 0164 1     ERROR        - Optional.  If specified, the address to unwind to
104 0165 1                 in case of an error.
105 0166 1
106 0167 1 IMPLICIT INPUTS:
107 0168 1
108 0169 1     NONE
109 0170 1
110 0171 1 IMPLICIT OUTPUTS:
111 0172 1
112 0173 1     NONE
113 0174 1
114 0175 1 ROUTINE VALUE:
115 0176 1
116 0177 1     The character read from the file
117 0178 1
118 0179 1 SIDE EFFECTS:
119 0180 1
120 0181 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
121 0182 1
122 0183 1 SIGNALLED ERRORS:
123 0184 1
124 0185 1
125 0186 1 --
126 0187 1
127 0188 2 BEGIN
128 0189 2
129 0190 2 LOCAL
130 0191 2     CHAR_READ,           ! Character read
131 0192 2     FCB:=REF $PASS$FCB_CONTROL_BLOCK, ! File Control block
132 0193 2     PFV_ADDR: VOLATILE, ! Enable argument
133 0194 2     UNWIND_ACT: VOLATILE, ! Enable argument
134 0195 2     ERROR_ADDR: VOLATILE; ! Enable argument
135 0196 2
136 0197 2 BUILTIN
137 0198 2     ACTUALCOUNT;      ! Count of arguments
138 0199 2

```

```

139 0200 2  ENABLE
140 0201 2  PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
141 0202 2
142 0203 2  !+
143 0204 2  ! Get ERROR parameter, if present.
144 0205 2  !-
145 0206 2
146 0207 2  IF ACTUALCOUNT () GEQU 2
147 0208 2  THEN
148 0209 2  ERROR_ADDR = .ERROR; ! Set unwind address
149 0210 2
150 0211 2  PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
151 0212 2
152 0213 2  !+
153 0214 2  ! Validate PFV and get PFV.
154 0215 2  !-
155 0216 2
156 0217 2  PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
157 0218 2
158 0219 2  !+
159 0220 2  ! Set unwind action to unlock file.
160 0221 2  !-
161 0222 2
162 0223 2  UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
163 0224 2
164 0225 2  !+
165 0226 2  ! Do common initialization.
166 0227 2  !-
167 0228 2
168 0229 2  PASS$INIT_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
169 0230 2
170 0231 2  !+
171 0232 2  ! Get the character to return and advance the record pointer.
172 0233 2  !-
173 0234 2
174 0235 2  IF .FCB [FCB$A_RECORD_CUR] GEQA .FCB [FCB$A_RECORD_END]
175 0236 2  THEN
176 0237 2  BEGIN
177 0238 2  CHAR_READ = %C' ';
178 0239 2  PFV [PFV$V_VALID] = 0;
179 0240 2  FCB [FCB$V_LAZY] = 1; ! Start lookahead for next line
180 0241 2  END
181 0242 2  ELSE
182 0243 2  CHAR_READ = CHR$CHAR_A (FCB [FCB$A_RECORD_CUR]);
183 0244 2
184 0245 2  !+
185 0246 2  ! Do end-of-READ processing.
186 0247 2  !-
187 0248 2
188 0249 2  PASS$END_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
189 0250 2
190 0251 2  RETURN .CHAR_READ; ! Return character
191 0252 2
192 0253 2  END; ! End of routine PASS$READ_CHAR

```

.TITLE PASS\$READ_CHAR Read a character

| | | | | | | |
|--|--|--|--|--------|---|--------|
| | | | | .IDENT | \1-002\ | |
| | | | | .EXTRN | PASS\$READ_CHAR, PASS\$READV_CHAR | |
| | | | | .EXTRN | PASS\$\$IO_HANDLER | |
| | | | | .EXTRN | PASS\$\$VALIDATE_PFV | |
| | | | | .EXTRN | PASS\$\$INIT_READ, PASS\$\$END_READ | |
| | | | | .PSECT | _PASS\$CODE, NOWRT, SHR, PIC, 2 | |
| | | | | .ENTRY | PASS\$READ_CHAR, Save R2,R3,R4,R5,R6,R7 | : 0144 |
| | | | | SUB! 2 | #8, SP | |
| | | | | CLRL | ERROR_ADDR | : 0188 |
| | | | | CLRQ | UNWIND_ACT | |
| | | | | MOVAL | 4\$, (FP) | |
| | | | | CMPB | (AP), #2 | : 0207 |
| | | | | BLSSU | 1\$ | |
| | | | | MOVL | ERROR, ERROR_ADDR | : 0209 |
| | | | | MOVL | PFV, R6 | : 0211 |
| | | | | JSB | PASS\$\$VALIDATE_PFV | : 0217 |
| | | | | MOVL | #1, UNWIND_ACT | : 0223 |
| | | | | JSB | PASS\$\$INIT_READ | : 0229 |
| | | | | CMPL | -20(FCB), -16(FCB) | : 0235 |
| | | | | BLSSU | 2\$ | |
| | | | | MOVL | #32, CHAR_READ | : 0238 |
| | | | | BICB2 | #1, 6(R6) | : 0239 |
| | | | | BISB2 | #4, -3(FCB) | : 0240 |
| | | | | BRB | 3\$ | : 0235 |
| | | | | MOVZBL | @-20(FCB), CHAR_READ | : 0243 |
| | | | | INCL | -20(FCB) | |
| | | | | JSB | PASS\$\$END_READ | : 0249 |
| | | | | RET | | : 0253 |
| | | | | .WORD | Save nothing | : 0188 |
| | | | | 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 | | |

; Routine Size: 113 bytes, Routine Base: _PASS\$CODE + 0000

: 193 0254 1
: 194 0255 1 !<BLF/PAGE>

```

196 0256 1 %SBTTL 'PASS$READV_CHAR - Read character from string'
197 0257 1 GLOBAL ROUTINE PASS$READV_CHAR (
198 0258 1     STRING: REF BLOCK [, BYTE],           ! String descriptor
199 0259 1     ERROR                               ! Error unwind address
200 0260 1 ) =
201 0261 1
202 0262 1 ++
203 0263 1 FUNCTIONAL DESCRIPTION:
204 0264 1
205 0265 1     This function reads a character from the specified string
206 0266 1     and returns it as the function value.
207 0267 1
208 0268 1 CALLING SEQUENCE:
209 0269 1
210 0270 1     Character.wbu.v = PASS$READV_CHAR (STRING.mt.ds [, ERROR.ja.r])
211 0271 1
212 0272 1 FORMAL PARAMETERS:
213 0273 1
214 0274 1     STRING           - The string to read from, passed as a class S
215 0275 1                   (assumed) descriptor. The length and pointer
216 0276 1                   are updated to reflect the unread string.
217 0277 1
218 0278 1     ERROR           - Optional. If specified, the address to unwind to
219 0279 1                   in case of an error.
220 0280 1
221 0281 1 IMPLICIT INPUTS:
222 0282 1
223 0283 1     NONE
224 0284 1
225 0285 1 IMPLICIT OUTPUTS:
226 0286 1
227 0287 1     NONE
228 0288 1
229 0289 1 ROUTINE VALUE:
230 0290 1
231 0291 1     The value of the character read.
232 0292 1
233 0293 1 SIDE EFFECTS:
234 0294 1
235 0295 1     NONE
236 0296 1
237 0297 1 SIGNALLED ERRORS:
238 0298 1
239 0299 1     NONE
240 0300 1
241 0301 1 --
242 0302 1
243 0303 2 BEGIN
244 0304 2
245 0305 2 LOCAL
246 0306 2     PFV: $PASS$PFV_FILE_VARIABLE, ! Pascal File Variable
247 0307 2     RESULT,                       ! Result value
248 0308 2     ARG_LIST: VECTOR [2, LONG],    ! Argument list
249 0309 2     PFV_ADDR: VOLATILE,           ! Enable argument
250 0310 2     ERROR_ADDR: VOLATILE;        ! Enable argument
251 0311 2
252 0312 2 BUILTIN

```



```

253 0313 2 ACTUALCOUNT; . Count of arguments
254 0314 2
255 0315 2 ENABLE
256 0316 2 PASS$IO_HANDLER (PFV_ADDR, ERROR_ADDR); ! Enable error handler
257 0317 2
258 0318 2 !+
259 0319 2 ! Get ERROR parameter, if present.
260 0320 2 !-
261 0321 2
262 0322 2 IF ACTUALCOUNT () GEQU 2
263 0323 2 THEN
264 0324 2 ERROR_ADDR = .ERROR; ! Set unwind address
265 0325 2
266 0326 2 PFV_ADDR = PFV; ! Set PFV address
267 0327 2
268 0328 2 !+
269 0329 2 ! Set up ARG_LIST.
270 0330 2 !-
271 0331 2
272 0332 2 ARG_LIST [0] = 1; ! One argument
273 0333 2 ARG_LIST [1] = PFV; ! PFV address
274 0334 2
275 0335 2 !+
276 0336 2 ! Call PASS$DO_READV to do the work, giving it the address of
277 0337 2 ! PASSREAD_CHAR to call.
278 0338 2 !-
279 0339 2
280 0340 2 PASS$DC_READV (PFV, .STRING, ARG_LIST, PASSREAD_CHAR; RESULT);
281 0341 2
282 0342 2 RETURN .RESULT;
283 0343 2
284 0344 1 END; ! End of routine PASS$READV_CHAR

```

```

                                .EXTRN PASS$DO_READV
                                .ENTRY PASS$READV_CHAR, Save R2,R3,R4,R6
005C 00000 SE 1C C2 00002   SUBL2 #28, SP
0000 00005 7E D4 00005   CLRL ERROR_ADDR
0000 00007 04 AE D4 00007   CLRL PFV_ADDR
0030 0000A 6D 0030 CF DE 0000A   MOVAL 2$, (FP)
0000 0000F 02 0000 6C 91 0000F   CMPB (AP), #2
0000 00012 04 04 1F 00012   BLSSU 1$
0000 00014 6E 08 AC D0 00014   MOVL ERROR, ERROR_ADDR
0000 00018 04 AE 10 AE 9E 00018 1$: MOVAB PFV, PFV_ADDR
0000 0001D 08 AE 01 D0 0001D   MOVL #1, ARG_LIST
0000 00021 0C AE 10 AE 9E 00021   MOVAB PFV, ARG_LIST+4
0000 00026 54 FF65 CF 9E 00026   MOVAB PASS$READ_CHAR, R4
0000 0002B 53 08 AE 9E 0002B   MOVAB ARG_LIST, R3
0000 0002F 56 10 AE 9E 0002F   MOVAB PFV, R6
0000 00033 52 04 AC D0 00033   MOVL STRING, R2
0000 00037 00000000G 00 16 00037   JSB PASS$DO_READV
0000 0003D 04 0003D   RET
0000 0003E 2$: .WORD Save nothing
0000 00040 50 08 AC D0 00040   MOVL 8(AP), R0
0000 00044 50 04 A0 D0 00044   MOYL 4(R0), R0

```

PASSREAD_CHAR
1-002

Read a character
PASSREADV_CHAR - Read character from string

G 3
16-Sep-1984 01:53:40
14-Sep-1984 12:51:46

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

Page 8
(4)

| | | | | | | | |
|-----------|----|----|----|-------|-------|--------|----------------------|
| | | E0 | A0 | 9F | 00048 | PUSHAB | ERROR_ADDR |
| | | E4 | A0 | 9F | 00048 | PUSHAB | PFV_ADDR |
| | | | 02 | DD | 0004E | PUSHL | #2 |
| | | | 5E | DD | 00050 | PUSHL | SP |
| | 7E | 04 | AC | 7D | 00052 | MOVQ | 4(AP), -(SP) |
| 00000000G | 00 | | 03 | FB | 00056 | CALLS | #3, PASS\$IO_HANDLER |
| | | | 04 | 0005D | | RET | |

; Routine Size: 94 bytes, Routine Base: _PASSCODE + 0071

```

: 285      0345 1
: 286      0346 1 !<BLF/PAGE>

```

PASSREAD_CHAR Read a character
1-002 PASSREADV_CHAR - Read character from string

H 3
16-Sep-1984 01:53:40 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:46 [PASRTL.SRC]PASREACHA.B32;1

Page 9
(5)

PA
1-

: 288 0347 1 END
: 289 0348 1
: 290 0349 0 ELUDOM

! End of module PASSREAD_CHAR

PSECT SUMMARY

| Name | Bytes | Attributes |
|-----------|-------|--|
| _PASSCODE | 207 | 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 | 90 | 21 | 33 | 00:00.4 |

COMMAND QUALIFIERS

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

: Size: 207 code + 0 data bytes
: Run Time: 00:05.9
: Elapsed Time: 00:20.0
: Lines/CPU Min: 3525
: Lexemes/CPU-Min: 9464
: Memory Used: 65 pages
: Compilation Complete

