


```

PPPPPPPP      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEEE      AAAAAA      SSSSSSSS      TTTTTTTTTT      RRRRRRRR
PPPPPPPP      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEEE      AAAAAA      SSSSSSSS      TTTTTTTTTT      RRRRRRRR
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      SS      TT      RR      RR
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      SS      TT      RR      RR
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      SS      TT      RR      RR
PP      PP      AA      AA      SS      RP      RR      EE      AA      AA      SS      TT      RR      RR
PPPPPPPP      AA      AA      SSSSSS      RRRRRRRR      EEEEEEEEEEE      AA      AA      SSSSSS      TT      RRRRRRRR
PPPPPPPP      AA      AA      SSSSSS      RRRRRRRR      EEEEEEEEEEE      AA      AA      SSSSSS      TT      RRRRRRRR
PP      AAAAAAAAAA      SS      RR      RR      EE      AAAAAAAAAA      SS      TT      RR      RR
PP      AAAAAAAAAA      SS      RR      RR      EE      AAAAAAAAAA      SS      TT      RR      RR
PP      AA      AA      SS      RR      RR      EE      AA      AA      SS      TT      RR      RR
PP      AA      AA      SSSSSSSS      RR      RR      EE      AA      AA      SSSSSSSS      TT      RR      RR
PP      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEEE      AA      AA      SSSSSSSS      TT      RR      RR

```

```

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      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

PA
1-
.....
.....
.....
.....

```

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

```

```
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_STRING: NOVALUE,       ! Read from textfile
62 0124 1     PASSREADV_STRING: NOVALUE;     ! Read 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 'PASSREAD_STRING - Read a string from textfile'
83 0144 1 GLOBAL ROUTINE PASSREAD_STRING (
84 0145 1     STRING: REF VECTOR [, BYTE],           ! String to read into
85 0146 1     PFV: REF $PASSPFV FILE_VARIABLE,     ! File variable
86 0147 1     STRING_LENGTH: WORD,               ! Length of string
87 0148 1     ERROR                               ! Error unwind address
88 0149 1 ) : NOVALUE =
89 0150 1
90 0151 1 ++
91 0152 1 | FUNCTIONAL DESCRIPTION:
92 0153 1 |
93 0154 1 |     This function READs a fixed-length string from the specified textfile.
94 0155 1 |     If the string length is less than the number of characters remaining
95 0156 1 |     in the current line, only STRING_LENGTH characters will be read,
96 0157 1 |     otherwise the string will be padded with blanks.
97 0158 1 |
98 0159 1 | CALLING SEQUENCE:
99 0160 1 |
100 0161 1 |     STRING.wt.r = PASSREAD_STRING (PFV.mr.r, STRING_LENGTH.rwu.v
101 0162 1 |     [, ERROR.ja.r])
102 0163 1 |
103 0164 1 | FORMAL PARAMETERS:
104 0165 1 |
105 0166 1 |     PFV           - The Pascal File Variable (PFV) passed by reference.
106 0167 1 |                 The structure of the PFV is defined in PASPFV.REQ.
107 0168 1 |
108 0169 1 |     STRING_LENGTH - The length of the string to read.
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 |     STRING        - The string read, returned as a function value by
124 0185 1 |                 having the string address passed as the first
125 0186 1 |                 procedure parameter, in accordance with the
126 0187 1 |                 VAX Procedure Calling Standard.
127 0188 1 |
128 0189 1 |                 If an error occurs and is continued by a user handler,
129 0190 1 |                 the result returned is a blank string.
130 0191 1 |
131 0192 1 | SIDE EFFECTS:
132 0193 1 |
133 0194 1 |     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
134 0195 1 |
135 0196 1 | SIGNALLED ERRORS:
136 0197 1 |
137 0198 1 |
138 0199 1 | --

```

```

139 0200 1
140 0201 2 BEGIN
141 0202 2
142 0203 2 LOCAL
143 0204 2 CHARS_REMAINING, ! Number of characters remaining in line
144 0205 2 FCB: REF $PASSFCB CONTROL_BLOCK, ! File Control block
145 0206 2 PFV_ADDR: VOLATILE, ! Enable argument
146 0207 2 UNWIND_ACT: VOLATILE, ! Enable argument
147 0208 2 ERROR_ADDR: VOLATILE; ! Enable argument
148 0209 2
149 0210 2 BUILTIN
150 0211 2 ACTUALCOUNT; ! Count of arguments
151 0212 2
152 0213 2 ENABLE
153 0214 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
154 0215 2
155 0216 2 !+
156 0217 2 !- Get ERROR parameter, if present.
157 0218 2 !-
158 0219 2
159 0220 2 IF ACTUALCOUNT () GEQU 4
160 0221 2 THEN
161 0222 2 ERROR_ADDR = .ERPOR; . Set unwind address
162 0223 2
163 0224 2 PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
164 0225 2
165 0226 2 !+
166 0227 2 !- Validate PFV and get PFV.
167 0228 2 !-
168 0229 2
169 0230 2 PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
170 0231 2
171 0232 2 !+
172 0233 2 !- Set unwind action to unlock file.
173 0234 2 !-
174 0235 2
175 0236 2 UNWIND_ACT = PASS$UNWIND_UNLOCK;
176 0237 2
177 0238 2 !+
178 0239 2 !- Do common initialization.
179 0240 2 !-
180 0241 2
181 0242 2 PASS$INIT_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
182 0243 2
183 0244 2 !+
184 0245 2 !- Determine number of characters remaining in line.
185 0246 2 !-
186 0247 2
187 0248 2 CHARS_REMAINING = .FCB [FCB$A_RECORD_END] - .FCB [FCB$A_RECORD_CUR];
188 0249 2
189 0250 2 !+
190 0251 2 !- Select lesser of CHARS_REMAINING and STRING_LENGTH as the number of
191 0252 2 !- characters to move.
192 0253 2 !-
193 0254 2
194 0255 2 CHARS_REMAINING = MINU (.CHARS_REMAINING, .STRING_LENGTH);
195 0256 2

```

```

196 0257 2 !+
197 0258 2 ! Move characters to user variable and update record position.
198 0259 2 !-
199 0260 2
200 0261 2 CH$COPY (.CHARS_REMAINING, .FCB [FCB$A_RECORD_CUR], %C' ',
201 0262 2 .STRING_LENGTH, STRING [0]);
202 0263 2 FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_CUR] + .CHARS_REMAINING;
203 0264 2
204 0265 2 !+
205 0266 2 ! Do end-of-READ processing.
206 0267 2 !-
207 0268 2
208 0269 2 PASS$END_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
209 0270 2
210 0271 2 RETURN;
211 0272 2
212 0273 1 END;
    
```

! End of routine PASS\$READ_STRING

.TITLE	PASS\$READ_STRING Read a fixed-length string	
.IDENT	\1-002\	
.EXTRN	PASS\$READ_STRING	
.EXTRN	PASS\$READV_STRING	
.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_STRING, Save R2, R3, R4, R5, R6, R7	: 0144
SUBL2	#8, SP	: 0201
CLRL	ERROR_ADDR	
CLRQ	UNWIND_ACT	
MOVAL	3\$, (FP)	
CMPB	(AP), #4	: 0220
BLSSU	1\$	
MOVL	ERROR, ERROR_ADDR	: 0222
MOVL	PFV, PFV_ADDR	: 0224
MOVL	PFV, R6	: 0230
JSB	PASS\$VALIDATE_PFV	
MOVL	#1, UNWIND_ACT	: 0236
MOVL	PFV, R6	: 0242
JSB	PASS\$INIT_READ	
SUBL3	-20(FCB), -16(FCB), CHARS_REMAINING	: 0248
MOVL	CHARS_REMAINING, R0	: 0255
CMPZV	#0, #T6, STRING_LENGTH, R0	
BGEQU	2\$	
MOVZWL	STRING_LENGTH, R0	
MOVL	R0, CHARS_REMAINING	
MOVCS	CHARS_REMAINING, @-20(FCB), #32, -	: 0262
	STRING_LENGTH, @STRING	
ADDL2	CHARS_REMAINING, -20(FCB)	: 0263
MOVL	PFV, R6	: 0269
JSB	PASS\$END_READ	
RET		: 0273
.WORD	Save nothing	: 0201

				00FC 0000	
	5E			08 C2 00002	
				7E D4 00C05	
		04		AE 7C 00007	
	6D	0057		CF DE 0000A	
	04			6C 91 0000F	
				04 1F 00012	
	6E	10		AC D0 00014	
	08			AE 08 AC D0 00018	1\$:
	56			AC D0 0001D	
			00000000G	00 16 00021	
	04			AE 01 D0 C0027	
	56			AC D0 0002B	
			00000000G	00 16 0002F	
	56			A7 EC A7 C3 00035	
	50			56 D0 0003B	
50	OC	AC		00 ED 0003E	
				04 1E 00044	
	50			AC 3C 00046	
	56			50 D0 0004A	2\$:
OC	AC	20		56 2C 0004D	
				BC 00054	
	LC			A7 56 C0 00056	
				56 AC D0 0005A	
			00000000G	00 16 0005E	
				04 00064	
				0000 00065	3\$:

PASS\$READ_STRING
1-002

Read a fixed-length string
PASS\$READ_STRING - Read a string from textfile

G 13
16-Sep-1984 02:03:05
14-Sep-1984 12:51:51

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

Page 6
(3)

	50	08	AC	D0	00067
	50	04	A0	D0	0006B
		F4	A0	9F	0006F
		F8	A0	9F	00072
		FC	A0	9F	00075
			03	DD	00078
			5E	DD	0007A
	7E	04	AC	7D	0007C
00000000G	00		03	FB	00080
			04		00087

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: 136 bytes, Routine Base: _PASS\$CODE + 0000

: 213 0274 1
: 214 0275 1 !<BLF/PAGE>

PAS
1-C
.....


```

216 0276 1 %SBTTL 'PASS$READV_STRING - Read string from string'
217 0277 1 GLOBAL ROUTINE PASS$READV_STRING (
218 0278 1     STRING: REF VECTOR [, BYTE],           ! String to read into
219 0279 1     LINE_DSC: REF VECTOR [, BYTE],      ! Line to read from
220 0280 1     STRING_LENGTH: WORD,                ! Length of string
221 0281 1     ERROR                               ! Error unwind address
222 0282 1 ) : NOVALUE =
223 0283 1
224 0284 1 +-+
225 0285 1 FUNCTIONAL DESCRIPTION:
226 0286 1
227 0287 1     This function READs a fixed-length string from the specified string line.
228 0288 1     If the string length is less than the number of characters remaining
229 0289 1     in the current line, only STRING_LENGTH characters will be read,
230 0290 1     otherwise the string will be padded with blanks.
231 0291 1
232 0292 1 CALLING SEQUENCE:
233 0293 1
234 0294 1     STRING.wt.r = PASS$READ_STRING (LINE_DSC.mq.r, STRING_LENGTH.rwu.v
235 0295 1     [, ERROR.ja.r])
236 0296 1
237 0297 1 FORMAL PARAMETERS:
238 0298 1
239 0299 1     LINE_DSC           - The string to read from, passed as a class S
240 0300 1                   (assumed) descriptor. The length and pointer
241 0301 1                   are updated to reflect the unread string.
242 0302 1
243 0303 1     STRING_LENGTH     - The length of the string to read.
244 0304 1
245 0305 1     ERROR             - Optional. If specified, the address to unwind to
246 0306 1                   in case of an error.
247 0307 1
248 0308 1 IMPLICIT INPUTS:
249 0309 1
250 0310 1     NONE
251 0311 1
252 0312 1 IMPLICIT OUTPUTS:
253 0313 1
254 0314 1     NONE
255 0315 1
256 0316 1 ROUTINE VALUE:
257 0317 1
258 0318 1     The value of the string read.
259 0319 1
260 0320 1 SIDE EFFECTS:
261 0321 1
262 0322 1     NONE
263 0323 1
264 0324 1 SIGNALLED ERRORS:
265 0325 1
266 0326 1     See PASS$READ_STRING
267 0327 1
268 0328 1 --
269 0329 1
270 0330 2 BEGIN
271 0331 2
272 0332 2 LOCAL

```


PASSREAD_STRING Read a fixed-length string
 1-002 PASSREADV_STRING - Read string from string

J 13
 16-Sep-1984 02:03:05
 14-Sep-1984 12:51:51

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

Page 9
 (4)

10	AE	04	AC	D0	00021	MOVL	STRING, ARG_LIST+4	:	0361
14	AE	1C	AE	9E	00026	MOVAB	PFV, ARG_LIST+8	:	0362
18	AE	0C	AC	3C	0002B	MOVZWL	STRING_LENGTH, ARG_LIST+12	:	0363
	54	FF44	CF	9E	00030	MOVAB	PASSREAD_STRING, R4	:	0370
	53	0C	AE	9E	00035	MOVAB	ARG_LIST, R3	:	
	56	1C	AE	9E	00039	MOVAB	PFV, R6	:	
	52	08	AC	D0	0003D	MOVL	LINE_DSC, R2	:	
		00000000G	00	16	00041	JSB	PASS\$DO_READV	:	
				04	00047	RET		:	0374
				0000	00048	.WORD	Save nothing	:	0330
	50	08	AC	D0	0004A	MOVL	8(AP), R0	:	
	50	04	AJ	D0	0004E	MOVL	4(R0), R0	:	
		D4	A0	9F	00052	PUSHAB	ERROR_ADDR	:	
		D8	A0	9F	00055	PUSHAB	UNWIND_ACT	:	
		DC	A0	9F	00058	PUSHAB	PFV_ADDR	:	
			03	DD	0005B	PUSHL	#3	:	
			5E	DD	0005D	PUSHL	SP	:	
	7E	04	AC	7D	0005F	MOVQ	4(AP), -(SP)	:	
	00000000G	00	03	FB	00063	CALLS	#3, PASS\$IO_HANDLER	:	
				04	0006A	RET		:	

: Routine Size: 107 bytes, Routine Base: _PASSCODE + 0088

: 315 0375 1
 : 316 0376 1 !<BLF/PAGE>

PASS\$READ_STRING Read a fixed-length string
1-002 PASS\$READV_STRING - Read string from string

K 13
16-Sep-1984 02:03:05 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:51 [PASRTL.SRC]PASREASTR.B32;1

Page 10
(5)

: 318 0377 1 END
: 319 0378 1
: 320 0379 0 ELUDOM

! End of module PASS\$READ_STRING

PSECT SUMMARY

Name Bytes Attributes
:_PASS\$CODE 243 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	00:00.9
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	90	21	33	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:PASREASTR/OBJ=OBJ\$:PASREASTR MSRCS\$:PASREASTR/UPDATE=(ENHS:PASREASTR)

: Size: 243 code + 0 data bytes
: Run Time: 00:06.5
: Elapsed Time: 00:24.0
: Lines/CPU Min: 3525
: Lexemes/CPU-Min: 11627
: Memory Used: 68 pages
: Compilation Complete



PASREAB00
LIS

PASREAST1
LIS

PASREAREH
LIS

PASRESETK
LIS

PASREAREG
LIS

PASRAB
LIS

PASRESET2
LIS

PASREADLN
LIS

PASREAREF
LIS

PASREAVAR
LIS

PASREARED
LIS

PASREACHA
LIS

PASREADUT
LIS

PASREAENU
LIS

PASREAUNS
LIS

PASREAINTE
LIS

PASREASTR
LIS