

```

PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPP      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
PPPPPPPPPPPP      AAA              AAA              SSSSSSSSSS      RRRRRRRRRRRR      TTT              LLL
PPPPPPPPPPPP      AAA              AAA              SSSSSSSSSS      RRRRRRRRRRRR      TTT              LLL
PPPPPPPPPPPP      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
PAS

PAS
PAS

PAS

PAS

PAS
PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS

PAS
PAS

PAS

PAS

PAS
PAS

PAS
PAS

PAS

22

PAS

PAS

TABLE 1

PAS

PAS

PAS

PAS

PAS
PAS

PAS

PAS

PAS

PAS
PAS

PAS

PAS

PAS
PAS

PAS

```
PPPPPPPP      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEE      AAAAAA      VV      VV      AAAAAA      RRRRRRRR
PPPPPPPP      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEE      AAAAAA      VV      VV      AAAAAA      RRRRRRRR
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      VV      VV      AA      AA      RR      RR
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      VV      VV      AA      AA      RR      RR
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      VV      VV      AA      AA      RR      RR
PP      PP      AA      AA      SS      RR      RR      EE      AA      AA      VV      VV      AA      AA      RR      RR
PPPPPPPP      AA      AA      SSSSSS      RRRRRRRR      EEEEEEEE      AA      AA      VV      VV      AA      AA      RRRRRRRR
PPPPPPPP      AA      AA      SSSSSS      RRRRRRRR      EEEEEEEE      AA      AA      VV      VV      AA      AA      RRRRRRRR
PP      AAAAAAAAAA      SS      RR      RR      EE      AAAAAAAAAA      VV      VV      AAAAAAAAAA      RR      RR
PP      AAAAAAAAAA      SS      RR      RR      EE      AAAAAAAAAA      VV      VV      AAAAAAAAAA      RR      RR
PP      AA      AA      SS      RR      RR      EE      AA      AA      VV      VV      AA      AA      RR      RR
PP      AA      AA      SS      RR      RR      EE      AA      AA      VV      VV      AA      AA      RR      RR
PP      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEE      AA      AA      VV      VV      AA      AA      RR      RR
PP      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEE      AA      AA      VV      VV      AA      AA      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      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```



```
1 0001 0 MODULE PASSREAD_VARYING ( %TITLE 'Read a varying string'
2 0002 0 IDENT = '1-002' ! File: PASREAVAR.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 which read a varying 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
```

PASSREAD_VARYIN Read a varying string
1-002 Declarations

L 14
16-Sep-1984 02:04:39
14-Sep-1984 12:51:53

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASREAVAR.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';      ! 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_VARYING: NOVALUE,      ! Read from textfile
: 62      0124 1     PASSREADV_VARYING: NOVALUE;      ! 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 'PASSREAD VARYING - Read a varying string from textfile'
83 0144 1 GLOBAL ROUTINE PASSREAD VARYING (
84 0145 1     STRING: REF VECTOR [, WORD],           ! String to read into
85 0146 1     PFV: REF $PASSPFV_FILE_VARIABLE,       ! File variable
86 0147 1     MAX_LENGTH: WORD,                     ! Maximum string length
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 varying string from the specified textfile.
94 0155 1     Up to MAX_LENGTH characters, or until end-of-line, whichever comes
95 0156 1     first, will be read. The number of characters read is stored in
96 0157 1     the first word of the string.
97 0158 1
98 0159 1 CALLING SEQUENCE:
99 0160 1
100 0161 1     STRING.wvt.v = PASSREAD_VARYING (PFV.mr.r, MAX_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     MAX_LENGTH    - The maximum length of the string.
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 varying string read returned as a function value.
124 0185 1                 In accordance with the VAX Procedure Calling Standard,
125 0186 1                 the address of the string is passed to this procedure
126 0187 1                 as the first parameter, and all other parameters are
127 0188 1                 shifted left one position.
128 0189 1
129 0190 1                 If an error occurs and is continued by a user handler,
130 0191 1                 the value returned is a zero-length string.
131 0192 1
132 0193 1 SIDE EFFECTS:
133 0194 1
134 0195 1     If the f      is the standard file INPUT or OUTPUT, it is implicitly opened.
135 0196 1
136 0197 1 SIGNALLED ERRORS
137 0198 1
138 0199 1     NONE
  
```

```

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

```



```

: 196 0257 2 CHARS_REMAINING = MINU (.CHARS_REMAINING, .MAX_LENGTH);
: 197 0258 2
: 198 0259 2
: 199 0260 2
: 200 0261 2
: 201 0262 2
: 202 0263 2 CH$MOVE (.CHARS_REMAINING, .FCB [FCB$A_RECORD_CUR], STRING [1]);
: 203 0264 2 FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_CUR] + .CHARS_REMAINING;
: 204 0265 2
: 205 0266 2
: 206 0267 2
: 207 0268 2
: 208 0269 2
: 209 0270 2 STRING [0] = .CHARS_REMAINING;
: 210 0271 2
: 211 0272 2
: 212 0273 2
: 213 0274 2
: 214 0275 2
: 215 0276 2 PASS$END_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
: 216 0277 2
: 217 0278 2 RETURN;
: 218 0279 2
: 219 0280 1 END;

```

! End of routine PASS\$READ_VARYING

.TITLE PASS\$READ_VARYING Read a varying string
 .IDENT \1-002\

.EXTRN PASS\$READ_VARYING
 .EXTRN PASS\$READ_VARYING
 .EXTRN PASS\$IO_HANDLER
 .EXTRN PASS\$VALIDATE_PFV
 .EXTRN PASS\$INIT_READ, PASS\$END_READ

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

			01FC 00000	.ENTRY	PASS\$READ_VARYING, Save R2,R3,R4,R5,R6,R7,R8	: 0144
	5E		08 C2 00002	SUBL2	#8, SP	
			7E D4 00005	CLRL	ERROR_ADDR	: 0203
		04	AE 7C 00007	CLRQ	UNWIND_ACT	
	6D	005B	CF DE 0000A	MOVAL	3\$, (FP)	
	04		6C 91 0000F	CMPB	(AP), #4	: 0222
			04 1F 00012	BLSSU	1\$	
	6E	10	AC D0 00014	MOVL	ERROR, ERROR_ADDR	: 0224
08	AE	08	AC D0 00018	MOVL	PFV, PFV_ADDR	: 0226
	56	08	AC D0 0001D	MOVL	PFV, R6	: 0232
		00000000G	00 16 00021	JSB	PASS\$VALIDATE_PFV	
04	AE	01	D0 00027	MOVL	#1, UNWIND_ACT	: 0238
	56	08	AC D0 0002B	MOVL	PFV, R6	: 0244
		00000000G	00 16 0002F	JSB	PASS\$INIT_READ	
	F0	EC	A7 C3 00035	SUBL3	-20(FCB), -16(FCB), CHARS_REMAINING	: 0250
		56	D0 0003B	MOVL	CHARS_REMAINING, R0	: 0257
50			00 ED 0003E	CMPZV	#0, #T6, MAX_LENGTH, R0	
			04 1E 00044	BGEQU	2\$	
	50	OC	AC 3C 00046	MOVZWL	MAX_LENGTH, R0	
	56		50 D0 0004A	MOVL	R0, CHARS_REMAINING	
						: 2\$:

02	A8	EC	58	04	AC	D0	0004D	MOVL	STRING, R8	:	0263
		EC	B7		56	28	00051	MOVC3	CHARS_REMAINING, a-20(FCB), 2(R8)	:	
			A7		56	C0	00057	ADDL2	CHARS_REMAINING, -20(FCB)	:	0264
			68		56	B0	0005B	MOVW	CHARS_REMAINING, (R8)	:	0270
			56	08	AC	D0	0005E	MOVL	PFV, R6	:	0276
				00000000G	00	16	00062	JSB	PASS\$END_READ	:	
						04	00068	RET		:	0280
						0000	00069	.WORD	Save nothing	:	0203
			50	08	AC	D0	0006B	MOVL	8(AP), R0	:	
			50	04	A0	D0	0006F	MOVL	4(R0), R0	:	
				F4	A0	9F	00073	PUSHAB	ERROR_ADDR	:	
				F8	A0	9F	00076	PUSHAB	UNWIND_ACT	:	
				FC	A0	9F	00079	PUSHAB	PFV_ADDR	:	
					03	DD	0007C	PUSHL	#3	:	
					5E	DD	0007E	PUSHL	SP	:	
			7E	04	AC	7D	00080	MOVQ	4(AP), -(SP)	:	
			00000000G	00	03	FB	00084	CALLS	#3, PASS\$IO_HANDLER	:	
						04	0008B	RET		:	

; Routine Size: 140 bytes, Routine Base: _PASS\$CODE + 0000

; 220 0281 1
 ; 221 0282 1 !<BLF/PAGE>


```

: 223 0283 1 %SBTTL 'PASS$READV_VARYING- Read varying from string'
: 224 0284 1 GLOBAL ROUTINE PASS$READV_VARYING (
: 225 0285 1     STRING: REF VECTOR [, BYTE],           ! String to read into
: 226 0286 1     LINE_DSC: REF VECTOR [, BYTE],         ! Line to read from
: 227 0287 1     MAX_LENGTH: WORD,                     ! Maximum length of string
: 228 0288 1     ERROR                                ! Error unwind address
: 229 0289 1 ) : NOVALUE =
: 230 0290 1
: 231 0291 1 ++
: 232 0292 1 FUNCTIONAL DESCRIPTION:
: 233 0293 1
: 234 0294 1     This procedure reads a varying-length string from the specified string line.
: 235 0295 1     Up to MAX_LENGTH characters, or until end-of-line, whichever comes
: 236 0296 1     first, will be read. The number of characters read is stored in
: 237 0297 1     the first word of the string.
: 238 0298 1
: 239 0299 1 CALLING SEQUENCE:
: 240 0300 1
: 241 0301 1     STRING.wvt.r = PASS$READV_VARYING (LINE_DSC.mq.r, MAX_LENGTH.rwu.v
: 242 0302 1     [, ERROR.ja.r])
: 243 0303 1
: 244 0304 1 FORMAL PARAMETERS:
: 245 0305 1
: 246 0306 1     LINE_DSC          - The string to read from, passed as a class S
: 247 0307 1                      (assumed) descriptor. The length and pointer
: 248 0308 1                      are updated to reflect the unread string.
: 249 0309 1
: 250 0310 1     MAX_LENGTH       - The maximum length of the string to read.
: 251 0311 1
: 252 0312 1     ERROR           - Optional. If specified, the address to unwind to
: 253 0313 1                      in case of an error.
: 254 0314 1
: 255 0315 1 IMPLICIT INPUTS:
: 256 0316 1
: 257 0317 1     NONE
: 258 0318 1
: 259 0319 1 IMPLICIT OUTPUTS:
: 260 0320 1
: 261 0321 1     NONE
: 262 0322 1
: 263 0323 1 ROUTINE VALUE:
: 264 0324 1
: 265 0325 1     The value of the string read.
: 266 0326 1
: 267 0327 1 SIDE EFFECTS:
: 268 0328 1
: 269 0329 1     NONE
: 270 0330 1
: 271 0331 1 SIGNALLED ERRORS:
: 272 0332 1
: 273 0333 1     See PASS$READ_VARYING
: 274 0334 1
: 275 0335 1 --
: 276 0336 1
: 277 0337 2 BEGIN
: 278 0338 2
: 279 0339 2 LOCAL

```

PASS\$READ_VARYIN Read a varying string
1-002 PASS\$READV_VARYING- Read varying from string

E 15
16-Sep-1984 02:04:39
14-Sep-1984 12:51:53

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PAS\$REAVAR.B32;1

Page 8
(4)

```

: 280      0340 2      PFV: $PASS$PFV_FILE_VARIABLE,      ! Pascal File Variable
: 281      0341 2      RESULT,                          ! Result value
: 282      0342 2      ARG_LIST: VECTOR [4, LONG],       ! Argument list
: 283      0343 2      PFV_ADDR: VOLATILE,               ! Enable argument
: 284      0344 2      UNWIND_ACT: VOLATILE,              ! Enable argument
: 285      0345 2      ERROR_ADDR: VOLATILE;             ! Enable argument
: 286      0346 2
: 287      0347 2      BUILTIN
: 288      0348 2      ACTUALCOUNT;                      ! Count of arguments
: 289      0349 2
: 290      0350 2      ENABLE
: 291      0351 2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);      ! Enable error handler
: 292      0352 2
: 293      0353 2      !+
: 294      0354 2      ! Get ERROR parameter, if present.
: 295      0355 2      !-
: 296      0356 2
: 297      0357 2      IF ACTUALCOUNT () GEQU 3
: 298      0358 2      THEN
: 299      0359 2      ERROR_ADDR = .ERROR;                ! Set unwind address
: 300      0360 2
: 301      0361 2      PFV_ADDR = PFV [PFV$R_PFV];          ! Set PFV address
: 302      0362 2
: 303      0363 2      !+
: 304      0364 2      ! Set up ARG_LIST.
: 305      0365 2      !-
: 306      0366 2
: 307      0367 2      ARG_LIST [0] = 3;                    ! Three arguments
: 308      0368 2      ARG_LIST [1] = STRING [0];          ! Destination string address
: 309      0369 2      ARG_LIST [2] = PFV [PFV$R_PFV];      ! Address of PFV
: 310      0370 2      ARG_LIST [3] = .MAX_LENGTH;          ! Maximum string length
: 311      0371 2
: 312      0372 2      !+
: 313      0373 2      ! Call PASS$DO_READV to do the work, giving it the address of
: 314      0374 2      ! PASS$READ_VARYING to call.
: 315      0375 2      !-
: 316      0376 2
: 317      0377 2      PASS$DO_READV (PFV [PFV$R_PFV], LINE_DSC [0], ARG_LIST, PASS$READ_VARYING);
: 318      0378 2
: 319      0379 2      RETURN;
: 320      0380 2
: 321      0381 1      END;                                ! End of routine PASS$READV_VARYING

```

```

                                .EXTRN  PASS$DO_READV
                                .ENTRY   PASS$READV_VARYING, Save R2,R3,R4,R6
                                SUBL2    #40, SP
                                CLRL     ERROR_ADDR
                                CLRQ     UNWIND_ACT
                                MOVAL    2$, (FP)
                                CMPB     (AP), #3
                                BLSSU    1$
                                MOVL     ERROR, ERROR_ADDR
                                MOVAB    PFV, PFV_ADDR
                                MOVL     #3, ARG_LIST
                                : 0284
                                :
                                : 0337
                                :
                                :
                                : 0357
                                :
                                :
                                : 0359
                                : 0361
                                : 0367

```


PASS\$READ_VARYIN Read a varying string
 1-002 PASS\$READV_VARYING- Read varying from string

F 15
 16-Sep-1984 02:04:39
 14-Sep-1984 12:51:53

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

Page 9
 (4)

10	AE	04	AC	D0	00021	MOVL	STRING, ARG_LIST+4	:	0368
14	AE	1C	AE	9E	00026	MOVAB	PFV, ARG_LIST+8	:	0369
18	AE	0C	AC	3C	0002B	MOVZWL	MAX_LENGTH, ARG_LIST+12	:	0370
54		FF40	CF	9E	00030	MOVAB	PASS\$READ_VARYING, R4	:	0377
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		:	0381
				0000	00048	.WORD	Save nothing	:	0337
50		08	AC	D0	0004A	MOVL	8(AP), R0	:	
50		04	AO	D0	0004E	MOVL	4(R0), R0	:	
		D4	AO	9F	00052	PUSHAB	ERROR_ADDR	:	
		D8	AO	9F	00055	PUSHAB	UNWIND_ACT	:	
		DC	AO	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: _PASS\$CODE + 008C

: 322 0382 1
 : 323 0383 1 !<BLF/PAGE>

PASS\$READ_VARYIN Read a varying string
1-002 PASS\$READV_VARYING- Read varying from string

G 15
16-Sep-1984 02:04:39 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:53 [PASRTL.SRC]PASREAVAR.B32;1

Page 10
(5)

: 325 0384 1 END
: 326 0385 1
: 327 0386 0 ELUDOM

! End of module PASS\$READ_VARYING

PSECT SUMMARY

Name	Bytes	Attributes
_PASS\$CODE	247	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\$:PASREAVAR/OBJ=OBJ\$:PASREAVAR MSRC\$:PASREAVAR/UPDATE=(ENH\$:PASREAVAR)

: Size: 247 code + 0 data bytes
: Run Time: 00:06.6
: Elapsed Time: 00:25.2
: Lines/CPU Min: 3498
: Lexemes/CPU-Min: 11574
: Memory Used: 69 pages
: Compilation Complete

0296 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

