


```

PPPPPPPP      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEE      AAAAAA      RRRRRRRR      EEEEEEEEEE      DDDDDDDD
PPPPPPPP      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEE      AAAAAA      RRRRRRRR      EEEEEEEEEE      DDDDDDDD
PP      PP      AA      AA      SS      RRR      RR      EE      AA      AA      RR      RR      EE      DD      DD
PP      PP      AA      AA      SS      RRR      RR      EE      AA      AA      RR      RR      EE      DD      DD
PP      PP      AA      AA      SS      RRR      RR      EE      AA      AA      RR      RR      EE      DD      DD
PP      PP      AA      AA      SS      RRR      RR      EE      AA      AA      RR      RR      EE      DD      DD
PPPPPPPP      AA      A/      SSSSSS      RRRRRRRR      EEEEEEEEE      AA      AA      RRRRRRRR      EEEEEEEEE      DD      DD
PPPPPPPP      AA      A1      SSSSSS      RRRRRRRR      EEEEEEEEE      AA      AA      RRRRRRRR      EEEEEEEEE      DD      DD
PP      AAAAAAAAAA      SS      RR      RR      EE      AAAAAAAAAA      RR      RR      EE      DD      DD
PP      AAAAAAAAAA      SS      RR      RR      EE      AAAAAAAAAA      RR      RR      EE      DD      DD
PP      AA      AA      SS      RR      RR      EE      AA      AA      RR      RR      EE      DD      DD
PP      AA      AA      SS      RR      RR      EE      AA      AA      RR      RR      EE      DD      DD
PP      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEE      AA      AA      RR      RR      EEEEEEEEEE      DDDDDDDD
PP      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEE      AA      AA      RR      RR      EEEEEEEEEE      DDDDDDDD

```

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

```

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_REAL_D ( %TITLE 'Read a D_floating value'
2 0002 0 IDENT = '1-002' ! File: PASREARED.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 D_floating value
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_REAL_D: CALL_VAL01 NOVALUE,       ! Read from textfile  
62      0124 1     PASSREADV_REAL_D: CALL_VAL01 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 REAL D - Read an D_floating value from textfile'
83 0144 1 GLOBAL ROUTINE PASSREAD_REAL_D (
84 0145 1     PFV: REF $PASS$PFV_FICE_VARIABLE,           ! File variable
85 0146 1     ERROR;                                   ! Error unwind address
86 0147 1     LO_RESULT,                               ! Low order part of result
87 0148 1     HI_RESULT,                               ! High order part of result
88 0149 1 ) : CALL_VAL01 NOVALUE=
89 0150 1
90 0151 1 +-+
91 0152 1 FUNCTIONAL DESCRIPTION:
92 0153 1
93 0154 1     This function reads an D_floating value from the specified textfile
94 0155 1     and returns it as the function value.
95 0156 1
96 0157 1 CALLING SEQUENCE:
97 0158 1
98 0159 1     Double.wd.v = PASSREAD_REAL_D (PFV.mr.r [, ERROR.ja.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     ERROR        - Optional. If specified, the address to unwind to
106 0167 1                 in case of an error.
107 0168 1
108 0169 1 IMPLICIT INPUTS:
109 0170 1
110 0171 1     NONE
111 0172 1
112 0173 1 IMPLICIT OUTPUTS:
113 0174 1
114 0175 1     NONE
115 0176 1
116 0177 1 ROUTINE VALUE:
117 0178 1
118 0179 1     The D_floating value of the number read.
119 0180 1
120 0181 1 SIDE EFFECTS:
121 0182 1
122 0183 1     If the file is the standard file INPUT or OUTPUT, it is implicitly opened.
123 0184 1
124 0185 1 SIGNALLED ERRORS:
125 0186 1
126 0187 1     INVSYNREA - invalid syntax for real value
127 0188 1     NOTVALTYP - "string" is not a value of type "type"
128 0189 1
129 0190 1 --
130 0191 1
131 0192 2 BEGIN
132 0193 2
133 0194 2 LOCAL
134 0195 2     RESULT: VECTOR [2, LONG],           ! D_floating result value
135 0196 2     DESCR: BLOCK [8, BYTE],           ! Descriptor for convert
136 0197 2     FCB: REF $PASS$FCB CONTROL_BLOCK, ! File Control block
137 0198 2     PFV_ADDR: VOLATILE,               ! Enable argument
138 0199 2     UNWIND_ACT: VOLATILE,             ! Enable argument
  
```

```

139      0200      2      ERROR_ADDR: VOLATILE;          . Enable argument
140      0201      2
141      0202      2      BUILTIN
142      0203      2      ACTUALCOUNT;          ! Count of arguments
143      0204      2
144      0205      2      ENABLE
145      0206      2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);    ! Enable error handler
146      0207      2
147      0208      2      !+
148      0209      2      ! Get ERROR parameter, if present.
149      0210      2      !-
150      0211      2
151      0212      2      IF ACTUALCOUNT () GEQU 2
152      0213      2      THEN
153      0214      2      ERROR_ADDR = .ERROR;          ! Set unwind address
154      0215      2
155      0216      2      PFV_ADDR = PFV [PFV$R_PFV];          ! Set PFV address
156      0217      2
157      0218      2      !+
158      0219      2      ! Validate PFV and get PFV.
159      0220      2      !-
160      0221      2
161      0222      2      PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
162      0223      2
163      0224      2      !+
164      0225      2      ! Set unwind action to unlock file.
165      0226      2      !-
166      0227      2
167      0228      2      UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
168      0229      2
169      0230      2      !+
170      0231      2      ! Do common initialization.
171      0232      2      !-
172      0233      2
173      0234      2      PASS$INIT_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
174      0235      2
175      0236      2      RESULT = 0;
176      0237      2
177      0238      2      !+
178      0239      2      ! Set up string descriptor for convert call.
179      0240      2      !-
180      0241      2
181      0242      2      DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
182      0243      2      DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
183      0244      2
184      0245      2      !+
185      0246      2      ! Call utility routine to find a string that looks like an real.
186      0247      2      ! If we can't find one, signal an error.
187      0248      2      !-
188      0249      2
189      0250      2      IF NOT PASS$GET_REAL (PFV [PFV$R_PFV], FCB [FCB$R_FCB];
190      0251      2      DESCR [DSC$A_POINTER], DESCR [DSC$W_LENGTH], FCB)
191      0252      2      THEN
192      0253      2      $PASSIO_ERROR (PASS INVSYNREA, 2, DESCR,
193      0254      2      .FCB [FCB$L_RECORD_NUMBER]); ! Invalid syntax
194      0255      2
195      0256      2      !+

```

```

196      0257 2      ! Call convert routine. If it fails, signal an error.
197      0258 2      !-
198      0259 2
199      0260 2      IF NOT OTSSCVT_T_D (DESCR, RESULT)
200      0261 2      THEN
201      P 0262 2      $PASSIO ERROR (PASS NOTVALTYP,3,DESCR,
202      P 0263 2      UPLIT BYTE (%CHARCOUNT('DOUBLE'),'DOUBLE'),
203      0264 2      .FCB [FCBSL_RECORD_NUMBER]);
204      0265 2
205      0266 2      !+
206      0267 2      ! Do end-of-READ processing.
207      0268 2      !-
208      0269 2
209      0270 2      PASS$END_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
210      0271 2
211      0272 2      !+
212      0273 2      ! Return result.
213      0274 2      !-
214      0275 2
215      0276 2      LO_RESULT = .RESULT [0];
216      0277 2      HI_RESULT = .RESULT [1];
217      0278 2      RETURN;
218      0279 2
219      0280 1      END;
  
```

! End of routine PASS\$READ_REAL_D

				.TITLE	PASS\$READ_REAL_D Read a D_floating value	
				.IDENT	\1-002\	
				.PSECT	_PASS\$CODE,NOWRT, SHR, PIC,2	
45	4C	42	55	4F	06 0000 P.AAA:	.BYTE 6
				44	00001	.ASCII \DOUBLE\
				.EXTRN	PASS\$READ_REAL_D	
				.EXTRN	PASS\$READ_REAL_D	
				.EXTRN	PASS\$IO_HANDLER	
				.EXTRN	PASS\$VALIDATE_PFV	
				.EXTRN	PASS\$INIT_READ, PASS\$GET_REAL	
				.EXTRN	PASS\$SIGNAL, PASS\$INVSYNREA	
				.EXTRN	OTSSCVT_T_D, PASS\$NOTVALTYP	
				.EXTRN	PASS\$END_READ	
				.ENTRY	PASS\$READ_REAL_D, Save R2,R3,R4,R5,R6,R7,R8,-;	0144
	59	00000000G	00	9E	00002	MOVAB PASS\$SIGNAL, R9
	5E		18	C2	00009	SUBL2 #24, SP
		04	7E	D4	0000C	CLRL ERROR_ADDR
			AE	7C	0000E	CLRQ UNWIND_ACT
	6D	007A	CF	DE	00011	MOVAL 4\$, (FP)
	02		6C	91	00016	CMPB (AP), #2
			04	1F	00019	BLSSU 1\$
	6E	08	AC	D0	0001B	MOVL ERROR, ERROR_ADDR
	56	04	AC	D0	0001F	MOVL PFV, R6
08	AE		56	D0	00023	MOVL R6, PFV_ADDR
		00000000G	00	16	00027	JSB PASS\$VALIDATE_PFV
04	AE		01	D0	0002D	MOVL #1, UNWIND_ACT
						0192
						0212
						0214
						0216
						0222
						0228

		00000000G	00	16	00031	JSB	PASS\$INIT_READ	:	0234
		14	AE	D4	00037	CLRL	RESULT	:	0236
0E	AE	010E	8F	B0	0003A	MOVW	#270, DESCR+2	:	0243
		00000000G	00	16	00040	JSB	PASS\$GET_REAL	:	0251
10	AE		54	D0	00046	MOVL	R4, DESCR+4	:	
OC	AE		55	B0	0004A	MOVW	R5, DESCR	:	
	10		50	EB	0004E	BLBS	R0, 2\$:	
		C8	A7	DD	00051	PUSHL	-56(FCB)	:	0254
		10	AE	9F	00054	PUSHAB	DESCR	:	
			02	DD	00057	PUSHL	#2	:	
	7E	00G	8F	9A	00059	MOVZBL	#PASS\$INVSYNREA, -(SP)	:	
	69		04	FB	0005D	CALLS	#4, PASS\$\$SIGNAL	:	
			04		00060	RET		:	
		14	AE	9F	00061	PUSHAB	RESULT	:	0260
		10	AE	9F	00064	PUSHAB	DESCR	:	
00000000G	00		02	FB	00067	CALLS	#2, OTSS\$CVT_T_D	:	
	13		50	EB	0006E	BLBS	R0, 3\$:	
		C8	A7	DD	00071	PUSHL	-56(FCB)	:	0264
		82	AF	9F	00074	PUSHAB	P.AAA	:	
		14	AE	9F	00077	PUSHAB	DESCR	:	
			03	DD	0007A	PUSHL	#3	:	
	7E	00G	8F	9A	0007C	MOVZBL	#PASS\$NOTVALTYP, -(SP)	:	
	69		05	FB	00080	CALLS	#5, PASS\$\$SIGNAL	:	
			04		00083	RET		:	
		00000000G	00	16	00084	JSB	PASS\$END_READ	:	0270
	50		14	AE	7D	MOVQ	RESULT, [O_RESULT	:	0276
			04		0008E	RET		:	0280
			0000		0008F	.WORD	Save nothing	:	0192
	50	08	AC	D0	00091	MOVL	8(AP), R0	:	
	50	04	A0	D0	00095	MOVL	4(R0), R0	:	
		E4	A0	9F	00099	PUSHAB	ERROR_ADDR	:	
		E8	A0	9F	0009C	PUSHAB	UNWIND_ACT	:	
		EC	A0	9F	0009F	PUSHAB	PFV_ADDR	:	
			03	DD	000A2	PUSHL	#3	:	
			5E	DD	000A4	PUSHL	SP	:	
	7E	04	AC	7D	000A6	MOVQ	4(AP), -(SP)	:	
00000000G	00		03	FB	000AA	CALLS	#3, PASS\$IO_HANDLER	:	
			04		000B1	RET		:	

: Routine Size: 178 bytes, Routine Base: _PASS\$CODE + 0007

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


```

: 223 0283 1 %SBTTL 'PASS$READV_REAL_D- Read a D_floating from string'
: 224 0284 1 GLOBAL ROUTINE PASS$READV_REAL_D (
: 225 0285 1     STRING: REF BLOCK [, BYTE],           ! String descriptor
: 226 0286 1     ERROR:                               ! Error unwind address
: 227 0287 1     LO_RESULT,                          ! Low order part of result
: 228 0288 1     HI_RESULT                             ! High order part of result
: 229 0289 1     ) : CALL_VAL01 NOVALUE =
: 230 0290 1
: 231 0291 1 +-
: 232 0292 1 FUNCTIONAL DESCRIPTION:
: 233 0293 1
: 234 0294 1     This function reads a D_floating from the specified string
: 235 0295 1     and returns it as the function value.
: 236 0296 1
: 237 0297 1 CALLING SEQUENCE:
: 238 0298 1
: 239 0299 1     Double.wd.v = PASS$READV_REAL_D (STRING.mt.ds [, ERROR.ja.r])
: 240 0300 1
: 241 0301 1 FORMAL PARAMETERS:
: 242 0302 1
: 243 0303 1     STRING           - The string to read from, passed as a class S
: 244 0304 1                   (assumed) descriptor. The length and pointer
: 245 0305 1                   are updated to reflect the unread string.
: 246 0306 1
: 247 0307 1     ERROR           - Optional. If specified, the address to unwind to
: 248 0308 1                   in case of an error.
: 249 0309 1
: 250 0310 1 IMPLICIT INPUTS:
: 251 0311 1
: 252 0312 1     NONE
: 253 0313 1
: 254 0314 1 IMPLICIT OUTPUTS:
: 255 0315 1
: 256 0316 1     NONE
: 257 0317 1
: 258 0318 1 ROUTINE VALUE:
: 259 0319 1
: 260 0320 1     The value of the D_floating read.
: 261 0321 1
: 262 0322 1 SIDE EFFECTS:
: 263 0323 1
: 264 0324 1     NONE
: 265 0325 1
: 266 0326 1 SIGNALLED ERRORS:
: 267 0327 1
: 268 0328 1     NONE
: 269 0329 1
: 270 0330 1 --
: 271 0331 1
: 272 0332 2 BEGIN
: 273 0333 2
: 274 0334 2 LOCAL
: 275 0335 2     PFV: $PASS$PFV FILE_VARIABLE,           ! Pascal File Variable
: 276 0336 2     ARG_LIST: VECTOR [2, LONG],       ! Argument list
: 277 0337 2     PFV_ADDR: VOLATILE,                 ! Enable argument
: 278 0338 2     UNWIND_ACT: VOLATILE,                 ! Enable argument
: 279 0339 2     ERROR_ADDR: VOLATILE;                 ! Enable argument

```

```

280 0340 2
281 0341 2 BUILTIN
282 0342 2 ACTUALCOUNT; ! Count of arguments
283 0343 2
284 0344 2 ENABLE
285 0345 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR); ! Enable error handler
286 0346 2
287 0347 2 !+
288 0348 2 ! Get ERROR parameter, if present.
289 0349 2 !-
290 0350 2
291 0351 2 IF ACTUALCOUNT () GEQU 2
292 0352 2 THEN
293 0353 2 ERROR_ADDR = .ERROR; ! Set unwind address
294 0354 2
295 0355 2 PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
296 0356 2
297 0357 2 !+
298 0358 2 ! Set up ARG_LIST.
299 0359 2 !-
300 0360 2
301 0361 2 ARG_LIST [0] = 1; ! One argument
302 0362 2 ARG_LIST [1] = PFV [PFV$R_PFV]; ! PFV address
303 0363 2
304 0364 2 !+
305 0365 2 ! Call PASS$DO_READV to do the work, giving it the address of
306 0366 2 ! PASS$READ_REAL_D to call.
307 0367 2 !-
308 0368 2
309 0369 2 PASS$DO_READV (PFV [PFV$R_PFV], .STRING, ARG_LIST, PASS$READ_REAL_D;
310 0370 2 LO_RESULT, HI_RESULT);
311 0371 2
312 0372 2 RETURN;
313 0373 2
314 0374 1 END; ! End of routine PASS$READV_REAL_D

```

```

                                .EXTRN PASS$DO_READV
                                .ENTRY PASS$READV_REAL_D, Save R2,R3,R4,R6
5E                                20 C2 00002
                                7E D4 00005
                                AE 7C 00007
6D                                CF DE 0000A
02                                6C 91 0000F
                                04 1F 00012
08                                6E 08 AC D0 00014
OC                                AE 14 AE 9E 00018 1$:
10                                AE 14 AE 9E 00021
                                54 FF24 CF 9E 00026
                                53 0C AE 9E 0002B
                                56 14 AE 9E 0002F
                                52 04 AC D0 00033
                                00000000G 00 16 00037
                                04 0003D
                                .ENTRY PASS$READV_REAL_D, Save R2,R3,R4,R6
                                SUBL2 #32, SP
                                CLRL ERROR_ADDR
                                CLRQ UNWIND_ACT
                                MOVAL 2$, (FP)
                                CMPB (AP), #2
                                BLSSU 1$
                                MOVL ERROR, ERROR_ADDR
                                MOVAB PFV, PFV_ADDR
                                MOVL #1, ARG_LIST
                                MOVAB PFV, ARG_LIST+4
                                MOVAB PASS$READ_REAL_D, R4
                                MOVAB ARG_LIST, R3
                                MOVAB PFV, R6
                                MOVL STRING, R2
                                JSB PASS$DO_READV
                                RET

```



```

: 318      0377 1 END
: 319      0378 1
: 320      0379 0 ELUDOM
! End of module PASSREAD_REAL_D
    
```

PSECT SUMMARY

```

Name          Bytes          Attributes
_PASS$CODE    282 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	6	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	100	23	33	00:00.4

COMMAND QUALIFIERS

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

```

: Size:          275 code + 7 data bytes
: Run Time:      00:07.3
: Elapsed Time: 00:29.6
: Lines/CPU Min: 3102
: Lexemes/CPU-Min: 13563
: Memory Used:  86 pages
: Compilation Complete
    
```

