

```

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      RRR                TTT                LLL
PPP                AAA          AAA          SSS                RRR                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
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS

PAS

PAS

PAC

PAS

PAS

PAS
PAS

PAS

PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS

```

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 PASSREAD_REAL_H ( %TITLE 'Read an H_floating value'
2 0002 0 -IDENT = '1-002' ! File: PASREAREH.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 an H_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';
: 55      0117 1 ! Externals, linkages, PSECTs, structures
: 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_H: NOVALUE,
: 62      0124 1 PASSREADV_REAL_H: NOVALUE;
: 63      0125 1 ! Read from textfile
: 64      0126 1 ! Read from string
: 65      0127 1
: 66      0128 1 MACROS:
: 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 H - Read an H_floating value from textfile'
83 0144 1 GLOBAL ROUTINE PASSREAD_REAL_H (
84 0145 1     RESULT: REF VECTOR [4, LONG],           ! H_floating result by reference
85 0146 1     PFV: REF $PASSPFV_FILE_VARIABLE,       ! File variable
86 0147 1     ERROR                                     ! Error unwind address
87 0148 1 ) : NOVALUE =
88 0149 1
89 0150 1 ++
90 0151 1 FUNCTIONAL DESCRIPTION:
91 0152 1
92 0153 1     This function reads an H_floating value from the specified textfile
93 0154 1     and returns it as its function value. Because the value is larger
94 0155 1     than 64 bits, it is returned to the first argument passed by reference,
95 0156 1     with all other arguments shifted right by one.
96 0157 1
97 0158 1 CALLING SEQUENCE:
98 0159 1
99 0160 1     CALL PASSREAD_REAL_H (RESULT.wh.r, PFV.mr.r [, ERROR.ja.r])
100 0161 1
101 0162 1 FORMAL PARAMETERS:
102 0163 1
103 0164 1     RESULT          - The address of the H_floating result.
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     ERROR          - Optional. If specified, the address to unwind to
109 0170 1                   in case of an error.
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     The H_floating value of the number read, returned by reference.
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     INVSYNREA - invalid syntax for real value
130 0191 1     NOTVALTYP - "string" is not a value of of type "type"
131 0192 1
132 0193 1 --
133 0194 1
134 0195 2 BEGIN
135 0196 2
136 0197 2 LOCAL
137 0198 2     DESC: BLOCK [8, BYTE], ! Descriptor for convert
138 0199 2     FCB: REF $PASSFCB_CONTROL_BLOCK, ! File Control block
  
```

```

: 139      0200      2      PFV_ADDR: VOLATILE,      ! Enable argument
: 140      0201      2      UNWIND_ACT: VOLATILE,      ! Enable argument
: 141      0202      2      ERROR_ADDR: VOLATILE;      ! Enable argument
: 142      0203      2
: 143      0204      2      BUILTIN
: 144      0205      2      ACTUALCOUNT;      ! Count of arguments
: 145      0206      2
: 146      0207      2      ENABLE
: 147      0208      2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);      ! Enable error handler
: 148      0209      2
: 149      0210      2      !+
: 150      0211      2      ! Get ERROR parameter, if present.
: 151      0212      2      !-
: 152      0213      2
: 153      0214      2      IF ACTUALCOUNT () GEQU 3
: 154      0215      2      THEN
: 155      0216      2      ERROR_ADDR = .ERROR;      ! Set unwind address
: 156      0217      2
: 157      0218      2      PFV_ADDR = PFV [PFV$R_PFV];      ! Set PFV address
: 158      0219      2
: 159      0220      2      !+
: 160      0221      2      ! Validate PFV and get PFV.
: 161      0222      2      !-
: 162      0223      2
: 163      0224      2      PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
: 164      0225      2
: 165      0226      2      !+
: 166      0227      2      ! Set unwind action to unlock file.
: 167      0228      2      !-
: 168      0229      2
: 169      0230      2      UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
: 170      0231      2
: 171      0232      2      !+
: 172      0233      2      ! Do common initialization.
: 173      0234      2      !-
: 174      0235      2
: 175      0236      2      PASS$INIT_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
: 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      $PASS$IO_ERROR (PASS$_INVSYNREA,2,DESCR,.FCB [FCB$L_RECORD_NUMBER]);
: 193      0254      2
: 194      0255      2      !+
: 195      0256      2      ! Call convert routine. If it fails, signal an error.

```


PASSREAD_REAL_H Read an H floating value
1-002

J 11
16-Sep-1984 02:01:28
14-Sep-1984 12:51:50

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

Page 5
(3)

```

196      0257 2      !-
197      0258 2
198      0259 2      IF NOT OTSSCVT_T_H (DESCR, RESULT [0])
199      0260 2      THEN
200      P 0261 2      $PASSIO_ERROR (PASS NOTVALTYP,3,DESCR,
201      P 0262 2      UPLIT BYTE (%CHARCOUNT('QUADRUPLE'),'QUADRUPLE'),
202      0263 2      .FCB [FCB$L_RECORD_NUMBER]);
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 PASSREAD_REAL_H

```

.TITLE PASSREAD_REAL_H Read an H_floating value
.IDENT \1-002\

.PSECT _PASSCODE,NOWRT, SHR, PIC,2

.BYTE 9
.ASCII \QUADRUPLE\

.EXTRN PASSREAD_REAL_H
.EXTRN PASSREADV_REAL_H
.EXTRN PASS$IO_HANDLER
.EXTRN PASS$VALIDATE_PFV
.EXTRN PASS$INIT_READ, PASS$GET_REAL
.EXTRN PASS$SIGNAL, PASS$INVSYNREA
.EXTRN OTSSCVT_T_H, PASS$NOTVALTYP
.EXTRN PASS$END_READ

.ENTRY PASSREAD_REAL_H, Save R2,R3,R4,R5,R6,R7,R8,-; 0144
R9
MOVAB PASS$SIGNAL, R9
SUBL2 #16, SP
CLRL ERROR_ADDR 0195
CLRQ UNWIND_ACT
MOVAL 4$, (FP)
CMPB (AP), #3 0214
BLSSU 1$
MOVL ERROR, ERROR_ADDR 0216
MOVL PFV, R6 0218
MOVL R6, PFV_ADDR
JSB PASS$VALIDATE_PFV 0224
MOVL #1, UNWIND_ACT 0230
JSB PASS$INIT_READ 0236
MOVW #270, DESCR+2 0243
JSB PASS$GET_REAL 0251
MOVL R4, DESCR+4
MOVW R5, DESCR
BLBS R0, 2$
PUSHL -56(FCB) 0253

45 4C 50 55 52 44 41 55 09 00000 P.AAA:
51 00001

03FC 00000
59 00000000G 00 9E 00002
5E 10 C2 00009
7E D4 0000C
04 AE 7C 0000E
6D 0073 CF DE 00011
03 6C 91 00016
04 1F 00019
6E 0C AC D0 0001B
56 08 AC D0 0001F 1$:
08 AE 56 D0 00023
04 AE 00000000G 00 16 00027
0E AE 00000000G 00 16 00031
0E AE 010E 8F B0 00037
10 AE 00000000G 00 16 0003D
0C AE 54 D0 00043
10 AE 55 B0 00047
C8 A7 DD 0004E

```

PASSREAD_REAL_H Read an H floating value
1-002

PASSREAD_REAL_H - Read an H_floating value from

K 11

16-Sep-1984 02:01:28
14-Sep-1984 12:51:50

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

Page 6
(3)

		10	AE	9F	00051		PUSHAB	DESCR	
			02	DD	00054		PUSHL	#2	
7E		00G	8F	9A	00056		MOVZBL	#PASSK_INVSYNREA, -(SP)	
69			04	FB	0005A		CALLS	#4, PASS\$SIGNAL	
				04	0005D		RET		
		04	AC	DD	0005E	2\$:	PUSHL	RESULT	0259
		10	AE	9F	00061		PUSHAB	DESCR	
00000000G	00		02	FB	00064		CALLS	#2, OT\$SCVT_T_H	
	13		50	E8	0006B		BLBS	R0, 3\$	
		C8	A7	DD	0006E		PUSHL	-56(FCB)	0263
		82	AF	9F	00071		PUSHAB	P.AAA	
		14	AE	9F	00074		PUSHAB	DESCR	
			03	DD	00077		PUSHL	#3	
7E		00G	8F	9A	00079		MOVZBL	#PASSK_NOTVALTYP, -(SP)	
69			05	FB	0007D		CALLS	#5, PASS\$SIGNAL	
				04	00080		RET		
		00000000G	00	16	00081	3\$:	JSB	PASS\$END_READ	0269
				04	00087		RET		0273
				0000	00088	4\$:	.WORD	Save nothing	0195
50		08	AC	D0	0008A		MOVL	8(AP), R0	
50		04	A0	D0	0008E		MOVL	4(R0), R0	
		EC	A0	9F	00092		PUSHAB	ERROR_ADDR	
		F0	A0	9F	00095		PUSHAB	UNWIND_ACT	
		F4	A0	9F	00098		PUSHAB	PFV_ADDR	
			03	DD	0009B		PUSHL	#3	
			5E	DD	0009D		PUSHL	SP	
	7E	04	AC	7D	0009F		MOVQ	4(AP), -(SP)	
00000000G	00		03	FB	000A3		CALLS	#3, PASS\$IO_HANDLER	
			04	000AA			RET		

; Routine Size: 171 bytes, Routine Base: _PASS\$CODE + 000A

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


```

216 0276 1 %SBTTL 'PASSREADV_REAL_H - Read an H_floating from string'
217 0277 1 GLOBAL ROUTINE PASSREADV_REAL_H (
218 0278 1     RESULT: REF VECTOR [4, LONG],           ! H floating result by reference
219 0279 1     LINE_DSC: REF VECTOR [, BYTE],         ! String descriptor
220 0280 1     ERROR                                     ! Error unwind address
221 0281 1 ): NOVALUE =
222 0282 1
223 0283 1 ++
224 0284 1 FUNCTIONAL DESCRIPTION:
225 0285 1
226 0286 1     This function reads an H_floating value from the specified string
227 0287 1     and returns it as its function value. Because the value is larger
228 0288 1     than 64 bits, it is returned to the first argument passed by reference,
229 0289 1     with all other arguments shifted right by one.
230 0290 1
231 0291 1 CALLING SEQUENCE:
232 0292 1
233 0293 1     CALL PASSREADV_REAL_H (RESULT.wh.r, LINE_DSC.mq.r [, ERROR.ja.r])
234 0294 1
235 0295 1 FORMAL PARAMETERS:
236 0296 1
237 0297 1     RESULT           - The address of the H_floating result.
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     ERROR            - Optional. If specified, the address to unwind to
244 0304 1                   in case of an error.
245 0305 1
246 0306 1 IMPLICIT INPUTS:
247 0307 1
248 0308 1     NONE
249 0309 1
250 0310 1 IMPLICIT OUTPUTS:
251 0311 1
252 0312 1     NONE
253 0313 1
254 0314 1 ROUTINE VALUE:
255 0315 1
256 0316 1     The value of the H_floating read.
257 0317 1
258 0318 1 SIDE EFFECTS:
259 0319 1
260 0320 1     NONE
261 0321 1
262 0322 1 SIGNALLED ERRORS:
263 0323 1
264 0324 1     NONE
265 0325 1
266 0326 1 --
267 0327 1
268 0328 2 BEGIN
269 0329 2
270 0330 2 LOCAL
271 0331 2     PFV: $PASSPFV_FILE_VARIABLE,      ! Pascal File Variable
272 0332 2     ARG_LIST: VECTOR [5, LONG],     ! Argument list

```

```

: 273      0333 2      PFV_ADDR: VOLATILE,      ! Enable argument
: 274      0334 2      UNWIND_ACT: VOLATILE,    ! Enable argument
: 275      0335 2      ERROR_ADDR: VOLATILE;    ! Enable argument
: 276      0336 2
: 277      0337 2      BUILTIN
: 278      0338 2      ACTUALCOUNT;            ! Count of arguments
: 279      0339 2
: 280      0340 2      ENABLE
: 281      0341 2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);    ! Enable error handler
: 282      0342 2
: 283      0343 2      !+
: 284      0344 2      !- Get ERROR parameter, if present.
: 285      0345 2
: 286      0346 2
: 287      0347 2      IF ACTUALCOUNT () GEQU 3
: 288      0348 2      THEN
: 289      0349 2      ERROR_ADDR = .ERROR;      ! Set unwind address
: 290      0350 2
: 291      0351 2      PFV_ADDR = PFV [PFV$R_PFV]; ! Set PFV address
: 292      0352 2
: 293      0353 2      !+
: 294      0354 2      !- Set up ARG_LIST.
: 295      0355 2
: 296      0356 2
: 297      0357 2      ARG_LIST [0] = 2;          ! Two arguments
: 298      0358 2      ARG_LIST [1] = RESULT [0]; ! Result address
: 299      0359 2      ARG_LIST [2] = PFV [PFV$R_PFV]; ! PFV address
: 300      0360 2
: 301      0361 2      !+
: 302      0362 2      !- Call PASS$DO_READV to do the work, giving it the address of
: 303      0363 2      !- PASS$READ_REAL_H to call.
: 304      0364 2
: 305      0365 2
: 306      0366 2      PASS$DO_READV (PFV [PFV$R_PFV], LINE_DSC [0], ARG_LIST, PASS$READ_REAL_H);
: 307      0367 2
: 308      0368 2      RETURN;
: 309      0369 2
: 310      0370 1      END;

```

! End of routine PASS\$READV_REAL_H

```

                                .EXTRN PASS$DO_READV
                                .ENTRY PASS$READV_REAL_H, Save R2,R3,R4,R6
                                SUBL2 #44, SP
                                CLRL ERROR_ADDR
                                CLRL UNWIND_ACT
                                MOVAL 2$, (FP)
                                CMPB (AP), #3
                                BLSSU 1$
                                MOVL ERROR, ERROR_ADDR
                                MOVAB PFV, PFV_ADDR
                                MOVL #2, ARG_LIST
                                MOVL RESULT, ARG_LIST+4
                                MOVAB PFV, ARG_LIST+8
                                MOVAB PASS$READ_REAL_H, R4
                                MOVAB ARG_LIST, R3
                                : 0277
                                : 0328
                                : 0347
                                : 0349
                                : 0351
                                : 0357
                                : 0358
                                : 0359
                                : 0366

```


56	20	AE	9E	00034	MOVAB	PFV, R6
52	08	AC	D0	00038	MOVL	LINE_DSC, R2
	00000000G	00	16	0003C	JSB	PASS\$DO_READV
			04	00042	RET	
			0000	00043	.WORD	Save nothing
50	08	AC	D0	00045	MOVL	8(AP), R0
50	04	A0	D0	00049	MOVL	4(R0), R0
	D0	A0	9F	0004D	PUSHAB	ERROR_ADDR
	D4	A0	9F	00050	PUSHAB	UNWIND_ACT
	D8	A0	9F	00053	PUSHAB	PFV_ADDR
		03	DD	00056	PUSHL	#3
		5E	DD	00058	PUSHL	SP
	7E	04	AC	7D	MOVQ	4(AP), -(SP)
00000000G	00	03	FB	0005E	CALLS	#3, PASS\$IO_HANDLER
			04	00065	RET	

0370
 0328

; Routine Size: 102 bytes, Routine Base: _PASS\$CODE + 00B5

; 311 0371 1
 ; 312 0372 1 !<BLF/PAGE>

PASS\$READ_REAL_H

1-002

Read an H_floating value

PASS\$READV_REAL_H - Read an H_floating from stri

B 12

16-Sep-1984 02:01:28

VAX-11 Bliss-32 V4.0-742

Page 10

(5)

14-Sep-1984 12:51:50

[PASRTL.SRC]PAS\$REAREH.B32;1

! End of module PASS\$READ_REAL_H

314

0373 1 END

315

0374 1

316

0375 0 ELUDOM

PSECT SUMMARY

Name

Bytes

Attributes

_PASS\$CODE

283

NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File

Symbols

Pages

Processing

Total

Loaded

Percent

Mapped

Time

\$255\$DUA28:[SYSLIB]STARLET.L32;1

9776

6

0

581

00:00.9

\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1

427

99

23

33

00:00.4

COMMAND QUALIFIERS

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

Size: 273 code + 10 data bytes

Run Time: 00:07.2

Elapsed Time: 00:25.2

Lines/CPU Min: 3133

Lexemes/CPU-Min: 14147

Memory Used: 83 pages

Compilation Complete

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

