```
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  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     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
PPP              AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
PPP              AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
PPP              AAAAAAAAAAAAAAA              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        LLLLLLLLLLLLLLL
PPP              AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
PPP              AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
PPPPPPPP      AAAAAA        SSSSSSSS  RRRRRRRR    EEEEEEEEEE    AAAAAA      RRRRRRRR    EEEEEEEEEE  FFFFFFFFFF
PPPPPPPP      AAAAAA        SSSSSSSS  RRRRRRRR    EEEEEEEEEE    AAAAAA      RRRRRRRR    EEEEEEEEEE  FFFFFFFFFF
PP      PP  AA      AA  SS          RR      RR  EE          AA      AA  RR      RR  EE          FF
PP      PP  AA      AA  SS          RR      RR  EE          AA      AA  RR      RR  EE          FF
PP      PP  AA      AA  SS          RR      RR  EE          AA      AA  RR      RR  EE          FF
PP      PP  AA      AA  SS          RR      RR  EE          AA      AA  RR      RR  EE          FF
PPPPPPPP    AA      AA    SSSSSS    RRRRRRRR    EEEEEEEE    AA      AA  RRRRRRRR    EEEEEEEE    FFFFFFFF
PPPPPPPP    AA      AA    SSSSSS    RRRRRRRR    EEEEEEEE    AA      AA  RRRRRRRR    EEEEEEEE    FFFFFFFF
PP          AAAAAAAAAA          SS  RR  RR      EE          AAAAAAAAAA  RR  RR      EE          FF
PP          AAAAAAAAAA          SS  RR  RR      EE          AAAAAAAAAA  RR  RR      EE          FF
PP          AA      AA          SS  RR    RR    EE          AA      AA  RR    RR    EE          FF              ....
PP          AA      AA          SS  RR    RR    EE          AA      AA  RR    RR    EE          FF              ....
PP          AA      AA  SSSSSSSS    RR      RR  EEEEEEEEEE  AA      AA  RR      RR  EEEEEEEEEE  FF              ....
PP          AA      AA  SSSSSSSS    RR      RR  EEEEEEEEEE  AA      AA  RR      RR  EEEEEEEEEE  FF              ....


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
```

```
0001  0 MODULE PAS$READ_REAL_F ( %TITLE 'Read an F_floating value'
0002  0                 IDENT = '1-002'                ! File: PASREAREF.B32 Edit: SBL1002
0003  0                 ) =
0004  1 BEGIN
0005  1 !
0006  1 !****************************************************************************
0007  1 !*                                                                          *
0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0010  1 !*  ALL RIGHTS RESERVED.                                                    *
0011  1 !*                                                                          *
0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0017  1 !*  TRANSFERRED.                                                            *
0018  1 !*                                                                          *
0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0021  1 !*  CORPORATION.                                                            *
0022  1 !*                                                                          *
0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0025  1 !*                                                                          *
0026  1 !*                                                                          *
0027  1 !****************************************************************************
0028  1 !
0029  1
0030  1 !++
0031  1 ! FACILITY:     Pascal Language Support
0032  1 !
0033  1 ! ABSTRACT:
0034  1 !
0035  1 !       This module contain, procedures which read an F_floating value
0036  1 !       from a textfile or a string.
0037  1 !
0038  1 ! ENVIRONMENT:  User mode - AST reentrant
0039  1 !
0040  1 ! AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
0041  1 !
0042  1 ! MODIFIED BY:
0043  1 !
0044  1 ! 1-001 - Original.  SBL 1-April-1981
0045  1 ! 1-002 - Use PAS$$END_READ.  SBL 26-May-1982
0046  1 !--
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     PAS$READ_REAL_F,                        ! Read from textfile
;   62          0124  1     PAS$READV_REAL_F;                       ! 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 'PAS$READ_REAL_F - Read an F_floating value from textfile'
;   83          0144  1 GLOBAL ROUTINE PAS$READ_REAL_F (
;   84          0145  1         PFV: REF $PAS$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 an F_floating value from the specified textfile
;   92          0153  1 !       and returns it as the function value.
;   93          0154  1 !
;   94          0155  1 ! CALLING SEQUENCE:
;   95          0156  1 !
;   96          0157  1 !       Single.wf.v = PAS$READ_REAL_F (PFV.mr.r [, ERROR.ja.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 PASPFV.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 F_floating value of the number read.
;  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 !       INVSYNREA - invalid syntax for real value
;  125          0186  1 !       NOTVALTYP- "string" is not a value of type "type"
;  126          0187  1 !
;  127          0188  1 !--
;  128          0189  1
;  129          0190  2     BEGIN
;  130          0191  2
;  131          0192  2     LOCAL
;  132          0193  2         RESULT,                         ! F_floating result value
;  133          0194  2         DESCR: BLOCK [8, BYTE],         ! Descriptor for convert
;  134          0195  2         FCB: REF $PAS$FCB_CONTROL_BLOCK,! File Control block
;  135          0196  2         PFV_ADDR: VOLATILE,             ! Enable argument
;  136          0197  2         UNWIND_ACT: VOLATILE,           ! Enable argument
;  137          0198  2         ERROR_ADDR: VOLATILE;           ! Enable argument
;  138          0199  2
```

```
BUILTIN
    ACTUALCOUNT;                        ! Count of arguments

ENABLE
    PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler

!+
! Get ERROR parameter, if present.
!-

IF ACTUALCOUNT () GEQU 2
THEN
    ERROR_ADDR = .ERROR;                ! Set unwind address

PFV_ADDR = PFV [PFV$R_PFV];             ! Set PFV address

!+
! Validate PFV and get PFV.
!-

PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);

!+
! Set unwind action to unlock file.
!-

UNWIND_ACT = PAS$K_UNWIND_UNLOCK;

!+
! Do common initialization.
!-

PAS$$INIT_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);

!+
! Set up string descriptor for convert call.
!-

DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;

!+
! Call utility routine to find a string that looks like an real.
! If we can't find one, signal an error.
!-

IF NOT PAS$$GET_REAL (PFV [PFV$R_PFV], FCB [FCB$R_FCB];
    DESCR [DSC$A_POINTER], DESCR [DSC$W_LENGTH], FCB)
THEN
    $PAS$IO_ERROR (PAS$_INVSYNREA,2,DESCR,.FCB [FCB$L_RECORD_NUMBER]);

!+
! Call convert routine.  If it fails, signal an error.
!-

IF NOT OTS$CVT_T_F (DESCR, RESULT)
THEN
```

```
;  196        P 0257  2          $PAS$IO_ERROR (PAS$_NOTVALTYP,3,DESCR,
;  197        P 0258  2               UPLIT BYTE (%CHARCOUNT('SINGLE'),'SINGLE'),
;  198          0259  2               .FCB [FCB$L_RECORD_NUMBER]);
;  199          0260  2
;  200          0261  2      !+
;  201          0262  2      ! Do end-of-READ processing.
;  202          0263  2      !-
;  203          0264  2
;  204          0265  2      PAS$$END_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
;  205          0266  2
;  206          0267  2      RETURN .RESULT;
;  207          0268  2
;  208          0269  1      END;                                          ! End of routine PAS$READ_REAL_F


                                                            .TITLE  PAS$READ_REAL_F Read an F_floating value
                                                            .IDENT  \1-002\

                                                            .PSECT  _PAS$CODE,NOWRT,  SHR,  PIC,2

                                    06  00000 P.AAA:        .BYTE   6                                                ;
                 45  4C  47  4E  49  53  00001              .ASCII  \SINGLE\                                         ;

                                                            .EXTRN  PAS$READ_REAL_F
                                                            .EXTRN  PAS$READV_REAL_F
                                                            .EXTRN  PAS$$IO_HANDLER
                                                            .EXTRN  PAS$$VALIDATE_PFV
                                                            .EXTRN  PAS$$INIT_READ, PAS$$GET_REAL
                                                            .EXTRN  PAS$$SIGNAL, PAS$K_INVSYNREA
                                                            .EXTRN  OTS$CVT_T_F, PAS$K_NOTVALTYP
                                                            .EXTRN  PAS$$END_READ

                                  03FC  00000              .ENTRY  PAS$READ_REAL_F, Save R2,R3,R4,R5,R6,R7,R8,-; 0144
                                                                    R9                                               ;
                 59 00000000G  00  9E  00002              MOVAB   PAS$$SIGNAL, R9                                  ;
                 5E            18  C2  00009              SUBL2   #24, SP                                          ;
                         04    AE  7C  0000C              CLRQ    ERROR_ADDR                                       ; 0190
                         0C    AE  D4  0000F              CLRL    PFV_ADDR                                         ;
                 6D      007B  CF  DE  00012              MOVAL   5$, (FP)                                         ;
                 02            6C  91  00017              CMPB    (AP), #2                                         ; 0210
                               05  1F  0001A              BLSSU   1$                                               ;
           04    AE      08    AC  D0  0001C              MOVL    ERROR, ERROR_ADDR                                ; 0212
                 56      04    AC  D0  00021 1$:          MOVL    PFV, R6                                          ; 0214
           0C    AE            56  D0  00025              MOVL    R6, PFV_ADDR                                     ;
                    00000000G  00  16  00029              JSB     PAS$$VALIDATE_PFV                                ; 0220
           08    AE            01  D0  0002F              MOVL    #1, UNWIND_ACT                                   ; 0226
                    00000000G  00  16  00033              JSB     PAS$$INIT_READ                                   ; 0232
           12    AE      010E  8F  B0  00039              MOVW    #270, DESCR+2                                    ; 0239
                    00000000G  00  16  0003F              JSB     PAS$$GET_REAL                                    ; 0247
           14    AE            54  D0  00045              MOVL    R4, DESCR+4                                      ;
           10    AE            55  B0  00049              MOVW    R5, DESCR                                        ;
                 11            50  E8  0004D              BLBS    R0, 2$                                           ;
                         C8    A7  DD  00050              PUSHL   -56(FCB)                                         ; 0249
                         14    AE  9F  00053              PUSHAB  DESCR                                            ;
                               02  DD  00056              PUSHL   #2                                               ;
                 7E       00G  8F  9A  00058              MOVZBL  #PAS$K_INVSYNREA, -(SP)                          ;
                 69            04  FB  0005C              CALLS   #4, PAS$$SIGNAL                                  ;
```

```
                                    2D  11 0005F          BRB     4$                                  ;
                                    5E  DD 00061 2$:      PUSHL   SP                                  ; 0255
                              14    AE  9F 00063          PUSHAB  DESCR                               ;
               00000000G  00        02  FB 00066          CALLS   #2, OTS$CVT_T_F                     ;
                          14        50  E8 0006D          BLBS    R0, 3$                              ;
                              C8    A7  DD 00070          PUSHL   -56(FCB)                            ; 0259
                              83    AF  9F 00073          PUSHAB  P.AAA                               ;
                              18    AE  9F 00076          PUSHAB  DESCR                               ;
                                    03  DD 00079          PUSHL   #3                                  ;
                          7E   00G  8F  9A 0007B          MOVZBL  #PAS$K_NOTVALTYP, -(SP)             ;
                          69        05  FB 0007F          CALLS   #5, PAS$$SIGNAL                     ;
                                    0A  11 00082          BRB     4$                                  ;
                     00000000G  00  16 00084 3$:      JSB     PAS$$END_READ                       ; 0265
                          50        6E  D0 0008A          MOVL    RESULT, R0                          ; 0267
                                        04 0008D          RET                                         ;
                                    50  D4 0008E 4$:      CLRL    R0                                  ; 0269
                                        04 00090          RET                                         ;
                                     0000 00091 5$:      .WORD   Save nothing                        ; 0190
                          50  08    AC  D0 00093          MOVL    8(AP), R0                           ;
                          50  04    A0  D0 00097          MOVL    4(R0), R0                           ;
                              EC    A0  9F 0009B          PUSHAB  ERROR_ADDR                          ;
                              F0    A0  9F 0009E          PUSHAB  UNWIND_ACT                          ;
                              F4    A0  9F 000A1          PUSHAB  PFV_ADDR                            ;
                                    03  DD 000A4          PUSHL   #3                                  ;
                                    5E  DD 000A6          PUSHL   SP                                  ;
                          7E  04    AC  7D 000A8          MOVQ    4(AP), -(SP)                        ;
               00000000G  00        03  FB 000AC          CALLS   #3, PAS$$IO_HANDLER                 ;
                                        04 000B3          RET                                         ;

; Routine Size:  180 bytes,    Routine Base:  _PAS$CODE + 0007


;  209          0270  1
;  210          0271  1 !<BLF/PAGE>
```

```
1 %SBTTL 'PAS$READV_REAL_F - Read an F_floating from string'
1 GLOBAL ROUTINE PAS$READV_REAL_F (
1         STRING: REF BLOCK [, BYTE],                          ! String descriptor
1         ERROR                                                ! Error unwind address
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This function reads an F_floating from the specified string
1 !       and returns it as the function value.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       Single.wf.v = PAS$READV_REAL_F (STRING.mt.ds [, ERROR.ja.r])
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       STRING          - The string to read from, passed as a class S
1 !                         (assumed) descriptor.  The length and pointer
1 !                         are updated to reflect the unread string.
1 !
1 !       ERROR           - Optional.  If specified, the address to unwind to
1 !                         in case of an error.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       The value of the F_floating read.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         PFV: $PAS$PFV_FILE_VARIABLE,    ! Pascal File Variable
2         RESULT,                         ! Result value
2         ARG_LIST: VECTOR [2, LONG],     ! Argument list
2         PFV_ADDR: VOLATILE,             ! Enable argument
2         UNWIND_ACT: VOLATILE,           ! Enable argument
2         ERROR_ADDR: VOLATILE;           ! Enable argument
2
```

```
;  269          0329  2       BUILTIN
;  270          0330  2           ACTUALCOUNT;                        ! Count of arguments
;  271          0331  2
;  272          0332  2       ENABLE
;  273          0333  2           PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  274          0334  2
;  275          0335  2       !+
;  276          0336  2       ! Get ERROR parameter, if present.
;  277          0337  2       !-
;  278          0338  2
;  279          0339  2       IF ACTUALCOUNT () GEQU 2
;  280          0340  2       THEN
;  281          0341  2           ERROR_ADDR = .ERROR;                ! Set unwind address
;  282          0342  2
;  283          0343  2       PFV_ADDR = PFV [PFV$R_PFV];             ! Set PFV address
;  284          0344  2
;  285          0345  2       !+
;  286          0346  2       ! Set up ARG_LIST.
;  287          0347  2       !-
;  288          0348  2
;  289          0349  2       ARG_LIST [0] = 1;                       ! One argument
;  290          0350  2       ARG_LIST [1] = PFV [PFV$R_PFV];         ! PFV address
;  291          0351  2
;  292          0352  2       !+
;  293          0353  2       ! Call PAS$$DO_READV to do the work, giving it the address of
;  294          0354  2       ! PAS$READ_REAL_F to call.
;  295          0355  2       !-
;  296          0356  2
;  297          0357  2       PAS$$DO_READV (PFV [PFV$R_PFV], .STRING, ARG_LIST, PAS$READ_REAL_F;
;  298          0358  2           RESULT);
;  299          0359  2
;  300          0360  2       RETURN .RESULT;
;  301          0361  2
;  302          0362  1       END;                                            ! End of routine PAS$READV_REAL_F



                                                      .EXTRN  PAS$$DO_READV

                                   005C 00000         .ENTRY  PAS$READV_REAL_F, Save R2,R3,R4,R6            ; 0273
                5E              20  C2 00002          SUBL2   #32, SP                                       ;
                                7E  D4 00005          CLRL    ERROR_ADDR                                    ; 0319
                      04        AE  7C 00007          CLRQ    UNWIND_ACT                                    ;
                6D    0030      CF  DE 0000A          MOVAL   2$, (FP)                                      ;
                02              6C  91 0000F          CMPB    (AP), #2                                      ; 0339
                                04  1F 00012          BLSSU   1$                                            ;
                6E      08      AC  D0 00014          MOVL    ERROR, ERROR_ADDR                             ; 0341
           08   AE      14      AE  9E 00018 1$:      MOVAB   PFV, PFV_ADDR                                 ; 0343
           0C   AE              01  D0 0001D          MOVL    #1, ARG_LIST                                  ; 0349
           10   AE      14      AE  9E 00021          MOVAB   PFV, ARG_LIST+4                               ; 0350
                54    FF22      CF  9E 00026          MOVAB   PAS$READ_REAL_F, R4                           ; 0357
                53      0C      AE  9E 0002B          MOVAB   ARG_LIST, R3                                  ;
                56      14      AE  9E 0002F          MOVAB   PFV, R6                                       ;
                52      04      AC  D0 00033          MOVL    STRING, R2                                    ;
                   00000000G    00  16 00037          JSB     PAS$$DO_READV                                 ;
                                    04 0003D          RET                                                   ; 0362
                              0000 0003E 2$:          .WORD   Save nothing                                  ; 0319
                              50        08   AC   D0 00040          MOVL    8(AP), R0                          ;
                              50        04   A0   D0 00044          MOVL    4(R0), R0                          ;
                                        DC   A0   9F 00048          PUSHAB  ERROR_ADDR                         ;
                                        E0   A0   9F 0004B          PUSHAB  UNWIND_ACT                         ;
                                        E4   A0   9F 0004E          PUSHAB  PFV_ADDR                           ;
                                             03   DD 00051          PUSHL   #3                                 ;
                                             5E   DD 00053          PUSHL   SP                                 ;
                              7E        04   AC   7D 00055          MOVQ    4(AP), -(SP)                       ;
                     00000000G  00           03   FB 00059          CALLS   #3, PAS$$IO_HANDLER                ;
                                                  04 00060          RET                                        ;

; Routine Size:  97 bytes,    Routine Base:  _PAS$CODE + 00BB


;  303          0363  1
;  304          0364  1 !<BLF/PAGE>
;  306          0365  1 END                                              ! End of module PAS$READ_REAL_F
;  307          0366  1
;  308          0367  0 ELUDOM



;                                  PSECT SUMMARY
;
;       Name                        Bytes                        Attributes
;
;    _PAS$CODE                          284  NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)



;                            Library Statistics
;
;                                            -------- Symbols --------       Pages       Processing
;       F le                                  Total    Loaded   Percent      Mapped      Time
;
;    _$255$DUA28:[SYSLIB]STARLET.L32;1         9776         6         0         581        00:01.0
;    _$255$DUA28:[PASRTL.OBJ]PASLIB.L32;1       427        99        23          33        00:00.4



;                                  COMMAND QUALIFIERS

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

; Size:          277 code + 7 data bytes
; Run Time:         00:07.2
; Elapsed Time:     00:31.7
; Lines/CPU Min:      3049
; Lexemes/CPU-Min: 12806
; Memory Used:  84 pages
; Compilation Complete
```

0296 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY