```
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     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     RRRRRRRRRRR        TTT        LLL
PPPPPPPPPPPP       AAA         AAA     SSSSSSSSS     RRRRRRRRRRR        TTT        LLL
PPPPPPPPPPPP       AAA         AAA     SSSSSSSSS     RRRRRRRRRRR        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  SSSSSSSSSSS   RRR         RRR     TTT        LLLLLLLLLLLLLLL
PPP                AAA         AAA  SSSSSSSSSSS   RRR         RRR     TTT        LLLLLLLLLLLLLLL
PPP                AAA         AAA  SSSSSSSSSSS   RRR         RRR     TTT        LLLLLLLLLLLLLLL
PPP                AAA         AAA  SSSSSSSSSSS   RRR         RRR     TTT        LLLLLLLLLLLLLLL
```

FILEID**PASREAREG

```
PPPPPPPP      AAAAAA      SSSSSSSS  RRRRRRRR    EEEEEEEEEE    AAAAAA    RRRRRRRR    EEEEEEEEEE    GGGGGGGG
PPPPPPPP      AAAAAA      SSSSSSSS  RRRRRRRR    EEEEEEEEEE    AAAAAA    RRRRRRRR    EEEEEEEEEE    GGGGGGGG
PP      PP   AA      AA   SS        RR      RR  EE            AA      AA  RR      RR  EE            GG
PP      PP   AA      AA   SS        RR      RR  EE            AA      AA  RR      RR  EE            GG
PP      PP   AA      AA   SS        RR      RR  EE            AA      AA  RR      RR  EE            GG
PP      PP   AA      AA   SS        RR      RR  EE            AA      AA  RR      RR  EE            GG
PPPPPPPP     AA      AA     SSSSSS  RRRRRRRR    EEEEEEEE      AA      AA  RRRRRRRR    EEEEEEEE      GG
PPPPPPPP     AA      AA     SSSSSS  RRRRRRRR    EEEEEEEE      AA      AA  RRRRRRRR    EEEEEEEE      GG
PP           AAAAAAAAAA           SS  RR  RR      EE          AAAAAAAAAA  RR  RR      EE            GG  GGGGGG
PP           AAAAAAAAAA           SS  RR  RR      EE          AAAAAAAAAA  RR  RR      EE            GG  GGGGGG
PP           AA      AA           SS  RR    RR    EE          AA      AA  RR    RR    EE            GG      GG      ....
PP           AA      AA           SS  RR    RR    EE          AA      AA  RR    RR    EE            GG      GG      ....
PP           AA      AA   SSSSSSSS  RR      RR  EEEEEEEEEE  AA      AA  RR      RR  EEEEEEEEEE    GGGGGG          ....
PP           AA      AA   SSSSSSSS  RR      RR  EEEEEEEEEE  AA      AA  RR      RR  EEEEEEEEEE    GGGGGG          ....

LL             IIIIII      SSSSSSSS
LL             IIIIII      SSSSSSSS
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 PAS$READ_REAL_G ( %TITLE 'Read a G_floating value'
;     2          0002  0                  IDENT = '1-002'                    ! File: PASREAREG.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 G_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 PAS$$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     PAS$READ_REAL_G: CALL_VAL01 NOVALUE,       ! Read from textfile
;    62          0124  1     PAS$READV_REAL_G: 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 'PAS$READ_REAL_G - Read a G_floating value from textfile'
;    83          0144  1 GLOBAL ROUTINE PAS$READ_REAL_G (
;    84          0145  1         PFV: REF $PAS$PFV_FILE_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_VALO1 NOVALUE =
;    89          0150  1
;    90          0151  1 !++
;    91          0152  1 ! FUNCTIONAL DESCRIPTION:
;    92          0153  1 !
;    93          0154  1 !       This function reads a G_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.wg.v = PAS$READ_REAL_G (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 G_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 !       OUTRANTYP - "string" is outside the range 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 $PAS$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          PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  146          0207  2
;  147          0208  2      !+
;  148          0209  2      ! Get ERROR parameter, if present.
;  149          0210  2      !-
;  150          0211  2
;  151          0212  2      IF ACTUALCOUNT () GEQU 2
;  152          0213  2      THEN
;  153          0214  2          ERROR_ADDR = .ERROR;            ! Set unwind address
;  154          0215  2
;  155          0216  2      PFV_ADDR = PFV [PFV$R_PFV];         ! Set PFV address
;  156          0217  2
;  157          0218  2      !+
;  158          0219  2      ! Validate PFV and get PFV.
;  159          0220  2      !-
;  160          0221  2
;  161          0222  2      PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
;  162          0223  2
;  163          0224  2      !+
;  164          0225  2      ! Set unwind action to unlock file.
;  165          0226  2      !-
;  166          0227  2
;  167          0228  2      UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
;  168          0229  2
;  169          0230  2      !+
;  170          0231  2      ! Do common initialization.
;  171          0232  2      !-
;  172          0233  2
;  173          0234  2      PAS$$INIT_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
;  174          0235  2
;  175          0236  2      !+
;  176          0237  2      ! Set up string descriptor for convert call.
;  177          0238  2      !-
;  178          0239  2
;  179          0240  2      DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
;  180          0241  2      DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  181          0242  2
;  182          0243  2      !+
;  183          0244  2      ! Call utility routine to find a string that looks like an real.
;  184          0245  2      ! If we can't find one, signal an error.
;  185          0246  2      !-
;  186          0247  2
;  187          0248  2      IF NOT PAS$$GET_REAL (PFV [PFV$R_PFV], FCB [FCB$R_FCB];
;  188          0249  2          DESCR [DSC$A_POINTER], DESCR [DSC$W_LENGTH], FCB)
;  189          0250  2      THEN
;  190          0251  2          $PAS$IO_ERROR (PAS$_INVSYNREA,2,DESCR,.FCB [FCB$L_RECORD_NUMBER]);
;  191          0252  2
;  192          0253  2      !+
;  193          0254  2      ! Call convert routine.  If it fails, signal an error.
;  194          0255  2      !-
;  195          0256  2
```

```
;  196          0257  2     IF NOT OTS$CVT_T_G (DESCR, RESULT)
;  197          0258  2     THEN
;  198        P 0259  2         $PASSIO_ERROR (PASS_NOTVALTYP,3,DESCR,
;  199        P 0260  2             UPLIT BYTE (%CHARCOUNT('DOUBLE'),'DOUBLE'),
;  200          0261  2             .FCB [FCB$L_RECORD_NUMBER]);
;  201          0262  2
;  202          0263  2     !+
;  203          0264  2     ! Do end-of-READ processing.
;  204          0265  2     !-
;  205          0266  2
;  206          0267  2     PASS$END_READ (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
;  207          0268  2
;  208          0269  2     !+
;  209          0270  2     ! Return result.
;  210          0271  2     !-
;  211          0272  2
;  212          0273  2     LO_RESULT = .RESULT [0];
;  213          0274  2     HI_RESULT = .RESULT [1];
;  214          0275  2     RETURN;
;  215          0276  2
;  216          0277  1     END;                                    ! End of routine PAS$READ_REAL_G

                                                                    .TITLE  PAS$READ_REAL_G Read a G_floating value
                                                                    .IDENT  \1-002\

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

                                                 06  00000 P.AAA:  .BYTE   6                                          ;
                                    45  4C  42  55  4F  44  00001   .ASCII  \DOUBLE\                                     ;

                                                                    .EXTRN  PAS$READ_REAL_G
                                                                    .EXTRN  PAS$READV_REAL_G
                                                                    .EXTRN  PAS$$IO_HANDLER
                                                                    .EXTRN  PAS$$VALIDATE_PFV
                                                                    .EXTRN  PAS$$INIT_READ, PAS$$GET_REAL
                                                                    .EXTRN  PAS$$SIGNAL, PAS$K_INVSYNREA
                                                                    .EXTRN  OTS$CVT_T_G, PAS$K_NOTVALTYP
                                                                    .EXTRN  PAS$$END_READ

                                                 03FC 00000         .ENTRY  PAS$READ_REAL_G, Save R2,R3,R4,R5,R6,R7,R8,-; 0144
                                                                            R9                                          ;
                                59 00000000G     00   9E 00002     MOVAB   PAS$$SIGNAL, R9                             ;
                                5E               18   C2 00009     SUBL2   #24, SP                                     ;
                                                 7E   D4 0000C     CLRL    ERROR_ADDR                                  ; 0192
                                         04      AE   7C 0000E     CLRQ    UNWIND_ACT                                  ;
                                6D       0077    CF   DE 00011     MOVAL   4$, (FP)                                    ;
                                02               6C   91 00016     CMPB    (AP), #2                                    ; 0212
                                                 04   1F 00019     BLSSU   1$                                          ;
                                6E          08   AC   D0 0001B     MOVL    ERROR, ERROR_ADDR                           ; 0214
                                56          04   AC   D0 0001F 1$:  MOVL    PFV, R6                                     ; 0216
                          08    AE               56   D0 00023     MOVL    R6, PFV_ADDR                                ;
                                00000000G        00   16 00027     JSB     PAS$$VALIDATE_PFV                           ; 0222
                          04    AE               01   D0 0002D     MOVL    #1, UNWIND_ACT                              ; 0228
                                00000000G        00   16 00031     JSB     PAS$$INIT_READ                              ; 0234
                          0E    AE         010E  8F   B0 00037     MOVW    #270, DESCR+2                               ; 0241
                                00000000G        00   16 0003D     JSB     PAS$$GET_REAL                               ; 0249
```

```
                          10   AE              54   D0 00043          MOVL    R4, DESCR+4                                    ;
                          0C   AE              55   B0 00047          MOVW    R5, DESCR                                      ;
                               10              50   E8 0004B          BLBS    R0, 2$                                         ;
                                         C8    A7   DD 0004E          PUSHL   -56(FCB)                                       ; 0251
                                         10    AE   9F 00051          PUSHAB  DESCR                                          ;
                                               02   DD 00054          PUSHL   #2                                             ;
                               7E        00G   8F   9A 00056          MOVZBL  #PAS$K_INVSYNREA, -(SP)                        ;
                               69              04   FB 0005A          CALLS   #4, PAS$$SIGNAL                                ;
                                               04   00 0005D          RET                                                    ;
                                         14    AE   9F 0005E 2$:      PUSHAB  RESULT                                         ; 0257
                                         10    AE   9F 00061          PUSHAB  DESCR                                          ;
                          00000000G      00        02   FB 00064          CALLS   #2, OTS$CVT_T_G                                ;
                               13              50   E8 0006B          BLBS    R0, 3$                                         ;
                                         C8    A7   DD 0006E          PUSHL   -56(FCB)                                       ; 0261
                                         85    AF   9F 00071          PUSHAB  P.AAA                                          ;
                                         14    AE   9F 00074          PUSHAB  DESCR                                          ;
                                               03   DD 00077          PUSHL   #3                                             ;
                               7E        00G   8F   9A 00079          MOVZBL  #PAS$K_NOTVALTYP, -(SP)                        ;
                               69              05   FB 0007D          CALLS   #5, PAS$$SIGNAL                                ;
                                               04   00 00080          RET                                                    ;
                                    00000000G  00   16   00081 3$:      JSB     PAS$$END_READ                                  ; 0267
                               50        14    AE   7D 00087          MOVQ    RESULT, [0_RESULT                              ; 0273
                                               04   00 0008B          RET                                                    ; 0277
                                                  0000 0008C 4$:      .WORD   Save nothing                                   ; 0192
                               50        08    AC   D0 0008E          MOVL    8(AP), R0                                      ;
                               50        04    A0   D0 00092          MOVL    4(R0), R0                                      ;
                                         E4    A0   9F 00096          PUSHAB  ERROR_ADDR                                     ;
                                         E8    A0   9F 00099          PUSHAB  UNWIND_ACT                                     ;
                                         EC    A0   9F 0009C          PUSHAB  PFV_ADDR                                       ;
                                               03   DD 0009F          PUSHL   #3                                             ;
                                               5E   DD 000A1          PUSHL   SP                                             ;
                               7E        04    AC   7D 000A3          MOVQ    4(AP), -(SP)                                   ;
                          00000000G      00        03   FB 000A7          CALLS   #3, PAS$$IO_HANDLER                            ;
                                               04   00 000AE          RET                                                    ;

; Routine Size:  175 bytes,    Routine Base:  _PAS$CODE + 0007

;  217          0278  1
;  218          0279  1 !<BLF/PAGE>
```

```
;  220     0280  1 %SBTTL 'PAS$READV_REAL_G- Read a G_floating from string'
;  221     0281  1 GLOBAL ROUTINE PAS$READV_REAL_G (
;  222     0282  1         LINE_DSC: REF VECTOR [, BYTE],                  ! Line descriptor
;  223     0283  1         ERROR;                                          ! Error unwind address
;  224     0284  1         LO_RESULT,                                      ! Low order part of result
;  225     0285  1         HI_RESULT                                       ! High order part of result
;  226     0286  1     ) : CALL_VALO1 NOVALUE =
;  227     0287  1
;  228     0288  1 !++
;  229     0289  1 ! FUNCTIONAL DESCRIPTION:
;  230     0290  1 !
;  231     0291  1 !       This function reads a G_floating from the specified string
;  232     0292  1 !       and returns it as the function value.
;  233     0293  1 !
;  234     0294  1 ! CALLING SEQUENCE:
;  235     0295  1 !
;  236     0296  1 !       Double.wg.v = PAS$READV_REAL_G (LINE_DSC.mq.r [, ERROR.ja.r])
;  237     0297  1 !
;  238     0298  1 ! FORMAL PARAMETERS:
;  239     0299  1 !
;  240     0300  1 !       LINE_DSC        - The string to read from, passed as a class S
;  241     0301  1 !                         (assumed) descriptor.  The length and pointer
;  242     0302  1 !                         are updated to reflect the unread string.
;  243     0303  1 !
;  244     0304  1 !       ERROR           - Optional.  If specified, the address to unwind to
;  245     0305  1 !                         in case of an error.
;  246     0306  1 !
;  247     0307  1 ! IMPLICIT INPUTS:
;  248     0308  1 !
;  249     0309  1 !       NONE
;  250     0310  1 !
;  251     0311  1 ! IMPLICIT OUTPUTS:
;  252     0312  1 !
;  253     0313  1 !       NONE
;  254     0314  1 !
;  255     0315  1 ! ROUTINE VALUE:
;  256     0316  1 !
;  257     0317  1 !       The value of the G_floating read.
;  258     0318  1 !
;  259     0319  1 ! SIDE EFFECTS:
;  260     0320  1 !
;  261     0321  1 !       NONE
;  262     0322  1 !
;  263     0323  1 ! SIGNALLED ERRORS:
;  264     0324  1 !
;  265     0325  1 !       NONE
;  266     0326  1 !--
;  267     0327  1
;  268     0328  1
;  269     0329  2     BEGIN
;  270     0330  2
;  271     0331  2     LOCAL
;  272     0332  2         PFV: $PAS$PFV_FILE_VARIABLE,    ! Pascal File Variable
;  273     0333  2         ARG_LIST: VECTOR [2, LONG],     ! Argument list
;  274     0334  2         PFV_ADDR: VOLATILE,             ! Enable argument
;  275     0335  2         UNWIND_ACT: VOLATILE,           ! Enable argument
;  276     0336  2         ERROR_ADDR: VOLATILE;           ! Enable argument
```

```
;  277     0337  2
;  278     0338  2      BUILTIN
;  279     0339  2          ACTUALCOUNT;                        ! Count of arguments
;  280     0340  2
;  281     0341  2      ENABLE
;  282     0342  2          PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  283     0343  2
;  284     0344  2      !+
;  285     0345  2      ! Get ERROR parameter, if present.
;  286     0346  2      !-
;  287     0347  2
;  288     0348  2      IF ACTUALCOUNT () GEQU 2
;  289     0349  2      THEN
;  290     0350  2          ERROR_ADDR = .ERROR;            ! Set unwind address
;  291     0351  2
;  292     0352  2      PFV_ADDR = PFV [PFV$R_PFV];         ! Set PFV address
;  293     0353  2
;  294     0354  2      !+
;  295     0355  2      ! Set up ARG_LIST.
;  296     0356  2      !-
;  297     0357  2
;  298     0358  2      ARG_LIST [0] = 1;                   ! One argument
;  299     0359  2      ARG_LIST [1] = PFV [PFV$R_PFV];     ! PFV address
;  300     0360  2
;  301     0361  2      !+
;  302     0362  2      ! Call PAS$$DO_READV to do the work, giving it the address of
;  303     0363  2      ! PAS$READ_REAL_G to call.
;  304     0364  2      !-
;  305     0365  2
;  306     0366  2      PAS$$DO_READV (PFV [PFV$R_PFV], LINE_DSC [0], ARG_LIST, PAS$READ_REAL_G;
;  307     0367  2          LO_RESULT, HI_RESULT);
;  308     0368  2
;  309     0369  2      RETURN;
;  310     0370  2
;  311     0371  1      END;                                   ! End of routine PAS$READV_REAL_G


                                                .EXTRN  PAS$$DO_READV

                                    005C 00000  .ENTRY  PAS$READV_REAL_G, Save R2,R3,R4,R6      ; 0281
                5E                  20  C2 00002  SUBL2  #32, SP
                                    7E  D4 00005  CLRL   ERROR_ADDR                             ; 0329
                          04        AE  7C 00007  CLRQ   UNWIND_ACT
                6D        0030      CF  DE 0000A  MOVAL  2$, (FP)
                02                  6C  91 0000F  CMPB   (AP), #2                               ; 0348
                                    04  1F 00012  BLSSU  1$
                6E          08      AC  D0 00014  MOVL   ERROR, ERROR_ADDR                      ; 0350
          08    AE          14      AE  9E 00018  1$:    MOVAB  PFV, PFV_ADDR                   ; 0352
          0C    AE                  01  D0 0001D  MOVL   #1, ARG_LIST                          ; 0358
          10    AE          14      AE  9E 00021  MOVAB  PFV, ARG_LIST+4                       ; 0359
                54        FF27      CF  9E 00026  MOVAB  PAS$READ_REAL_G, R4                   ; 0366
                53          0C      AE  9E 0002B  MOVAB  ARG_LIST, R3
                56          14      AE  9E 0002F  MOVAB  PFV, R6
                52          04      AC  D0 00033  MOVL   LINE_DSC, R2
                    00000000G       00  16 00037  JSB    PAS$$DO_READV
                                        04 0003D  RET                                           ; 0371
```

```
                                        0000 0003E 2$:      .WORD   Save nothing                             ; 0329
                                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 + 00B6

;  312          0372  1
;  313          0373  1 !<BLF/PAGE>
```

```
;  315          0374  1 END                                   ! End of module PAS$READ_REAL_G
;  316          0375  1
;  317          0376  0 ELUDOM
```

```
;                                PSECT SUMMARY
;
;      Name                      Bytes                         Attributes
;
;  _PAS$CODE                        279  NOVEC,NOWRT,  RD    EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
```

```
;                              Library Statistics
;
;                                        --------- Symbols ---------      Pages       Processing
;      File                              Total    Loaded   Percent       Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776         6         0          581        00:01.1
;  _$255$DUA28:[PASRTL.OBJ]PASLIB.L32;1    427       100        23           33        00:00.4
```

```
;                                COMMAND QUALIFIERS

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

; Size:         272 code + 7 data bytes
; Run Time:        00:07.3
; Elapsed Time:    00:29.0
; Lines/CPU Min:    3077
; Lexemes/CPU-Min: 13751
; Memory Used:  86 pages
; Compilation Complete
```

PASREABOO
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

PASREAINT
LIS

PASREASTR
LIS