```
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
```

```
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR     IIIIII    RRRRRRRR    EEEEEEEEEE  FFFFFFFFFF
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR     IIIIII    RRRRRRRR    EEEEEEEEEE  FFFFFFFFFF
PP      PP   AA      AA  SS        WW      WW  RR      RR     II      RR      RR  EE          FF
PP      PP   AA      AA  SS        WW      WW  RR      RR     II      RR      RR  EE          FF
PP      PP   AA      AA  SS        WW      WW  RR      RR     II      RR      RR  EE          FF
PP      PP   AA      AA  SS        WW      WW  RR      RR     II      RR      RR  EE          FF
PPPPPPPP     AA      AA    SSSSSS  WW      WW  RRRRRRRR       II      RRRRRRRR    EEEEEEEE    FFFFFFF
PPPPPPPP     AA      AA    SSSSSS  WW      WW  RRRRRRRR       II      RRRRRRRR    EEEEEEEE    FFFFFFF
PP           AAAAAAAAAA          SS  WW  WW  WW  RR  RR       II      RR  RR      EE          FF
PP           AAAAAAAAAA          SS  WW  WW  WW  RR  RR       II      RR  RR      EE          FF
PP           AA      AA          SS  WWWW  WWWW  RR    RR     II      RR    RR    EE          FF
PP           AA      AA          SS  WWWW  WWWW  RR    RR     II      RR    RR    EE          FF
PP           AA      AA  SSSSSSSS    WW      WW  RR      RR   IIIIII  RR      RR  EEEEEEEEEE  FF
PP           AA      AA  SSSSSSSS    WW      WW  RR      RR   IIIIII  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
```

```
;     1          0001  0 MODULE PAS$WRITE_REALE_F ( %TITLE 'Write an F_floating in E format'
;     2          0002  0                       IDENT = '1-002'                     ! File: PASWRIREF.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 a procedure which writes an F_floating in
;    36          0036  1 !       exponential format to a textfile.
;    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 - Make total-width a longword.  SBL 30-June-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$WRITE_REALE_F: NOVALUE,               ! Write to textfile
;    62          0124  1     PAS$WRITEV_REALE_F: NOVALUE;              ! Write to 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$WRITE_REALE_F - Write F_floating in E format to textfile'
;    83          0144  1 GLOBAL ROUTINE PAS$WRITE_REALE_F (
;    84          0145  1         PFV: REF $PAS$PFV_FILE_VARIABLE,                ! File variable
;    85          0146  1         VALUE,                                          ! Value to write
;    86          0147  1         TOTAL_WIDTH: SIGNED,                            ! Total field width
;    87          0148  1         ERROR                                           ! Error unwind address
;    88          0149  1     ): NOVALUE =
;    89          0150  1
;    90          0151  1 !++
;    91          0152  1 ! FUNCTIONAL DESCRIPTION:
;    92          0153  1 !
;    93          0154  1 !       This procedure writes an F_floating value in exponential notation
;    94          0155  1 !       to the specified textfile.
;    95          0156  1 !
;    96          0157  1 ! CALLING SEQUENCE:
;    97          0158  1 !
;    98          0159  1 !       CALL PAS$WRITE_REALE_F (PFV.mr.r, VALUE.rf.v, TOTAL_WIDTH.rl.v
;    99          0160  1 !                               [ERROR.j.r]])
;   100          0161  1 !
;   101          0162  1 ! FORMAL PARAMETERS:
;   102          0163  1 !
;   103          0164  1 !       PFV             - The Pascal File Variable (PFV) passed by reference.
;   104          0165  1 !                         The structure of the PFV is defined in PASPFV.REQ.
;   105          0166  1 !
;   106          0167  1 !       VALUE           - The F_floating value to write.
;   107          0168  1 !
;   108          0169  1 !       TOTAL_WIDTH     - Total field width.
;   109          0170  1 !
;   110          0171  1 !       ERROR           - Optional.  Address to unwind to if an error occurs.
;   111          0172  1 !
;   112          0173  1 ! IMPLICIT INPUTS:
;   113          0174  1 !
;   114          0175  1 !       NONE
;   115          0176  1 !
;   116          0177  1 ! IMPLICIT OUTPUTS:
;   117          0178  1 !
;   118          0179  1 !       NONE
;   119          0180  1 !
;   120          0181  1 ! ROUTINE VALUE:
;   121          0182  1 !
;   122          0183  1 !       NONE
;   123          0184  1 !
;   124          0185  1 ! SIDE EFFECTS:
;   125          0186  1 !
;   126          0187  1 !       If the file is the standard file OUTPUT, it is implicitly opened.
;   127          0188  1 !
;   128          0189  1 ! SIGNALLED ERRORS:
;   129          0190  1 !
;   130          0191  1 !       LINTOOLON - line too long
;   131          0192  1 !       NEGWIDDIG - negative Width or Digits specification is not allowed
;   132          0193  1 !
;   133          0194  1 !--
;   134          0195  1
;   135          0196  2     BEGIN
;   136          0197  2
;   137          0198  2     LOCAL
;   138          0199  2         FCB: REF $PAS$FCB_CONTROL_BLOCK,                ! File control block
```

```
;  139          0200  2          FIELD_WIDTH: SIGNED,                        ! Minimum width
;  140          0201  2          REMAINING_WIDTH,                            ! Width remaining on line
;  141          0202  2          PFV_ADDR: VOLATILE,                         ! Enable argument
;  142          0203  2          UNWIND_ACT: VOLATILE,                       ! Enable argument
;  143          0204  2          ERROR_ADDR: VOLATILE;                       ! Enable argument
;  144          0205  2
;  145          0206  2      BUILTIN
;  146          0207  2          ACTUALCOUNT;
;  147          0208  2
;  148          0209  2      ENABLE
;  149          0210  2          PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  150          0211  2
;  151          0212  2      !+
;  152          0213  2      ! Get ERROR parameter, if present.
;  153          0214  2      !-
;  154          0215  2
;  155          0216  2      IF ACTUALCOUNT () GEQU 4
;  156          0217  2      THEN
;  157          0218  2          ERROR_ADDR = .ERROR;            ! Set unwind address
;  158          0219  2
;  159          0220  2      PFV_ADDR = PFV [PFV$R_PFV];         ! Set PFV address
;  160          0221  2
;  161          0222  2      !+
;  162          0223  2      ! Validate PFV and get PFV.
;  163          0224  2      !-
;  164          0225  2
;  165          0226  2      PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
;  166          0227  2
;  167          0228  2      !+
;  168          0229  2      ! Set unwind action to unlock file.
;  169          0230  2      !-
;  170          0231  2
;  171          0232  2      UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
;  172          0233  2
;  173          0234  2      !+
;  174          0235  2      ! Do common initialization.
;  175          0236  2      !-
;  176          0237  2
;  177          0238  2      PAS$$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
;  178          0239  2
;  179          0240  2      !+
;  180          0241  2      ! Get field width and remaining width.  Check for valid field width.
;  181          0242  2      !-
;  182          0243  2
;  183          0244  2      FIELD_WIDTH = .TOTAL_WIDTH;
;  184          0245  2      IF .FIELD_WIDTH LSS 0
;  185          0246  2      THEN
;  186          0247  2          $PAS$IO_ERROR (PAS$_NEGWIDDIG,0);
;  187          0248  2      REMAINING_WIDTH = .FCB [FCB$A_RECORD_END] - .FCB [FCB$A_RECORD_CUR];
;  188          0249  2
;  189          0250  2      !+
;  190          0251  2      ! Convert to text and put in record buffer.  If failure, signal an error.
;  191          0252  2      ! FIELD_WIDTH will be modified to have the actual width used.
;  192          0253  2      !-
;  193          0254  2
;  194          0255  2      IF NOT PAS$CVT_F_T (VALUE,                  ! Value to convert
;  195          0256  2                          .FCB [FCB$A_RECORD_CUR],! Destination
```

```
;  196          0257  2                          FIELD_WIDTH,            ! Minimum/actual width
;  197          0258  2                          .REMAINING_WIDTH)       ! Width remaining on line
;  198          0259  2      THEN
;  199          0260  2          $PASSIO_ERROR (PASS_LINTOOLON,1,(.FIELD_WIDTH-.REMAINING_WIDTH));
;  200          0261  2
;  201          0262  2      !+
;  202          0263  2      ! Advance the record pointer.
;  203          0264  2      !-
;  204          0265  2
;  205          0266  2      FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_CUR] + .FIELD_WIDTH;
;  206          0267  2
;  207          0268  2      !+
;  208          0269  2      ! Call WRITE epilogue routine to move the last character written to the
;  209          0270  2      ! user's buffer and to unlock the file variable.
;  210          0271  2      !-
;  211          0272  2
;  212          0273  2      PASS$END_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
;  213          0274  2
;  214          0275  2      RETURN;
;  215          0276  2
;  216          0277  1      END;                                        ! End of routine PAS$WRITE_REALE_F


                                                                         .TITLE  PAS$WRITE_REALE_F Write an F_floating in E form
;                                                                                                  at
                                                                         .IDENT  \1-002\

                                                                         .EXTRN  PAS$WRITE_REALE_F
                                                                         .EXTRN  PAS$WRITEV_REALE_F
                                                                         .EXTRN  PAS$$IO_HANDLER
                                                                         .EXTRN  PAS$$VALIDATE_PFV
                                                                         .EXTRN  PAS$$INIT_WRITE
                                                                         .EXTRN  PAS$$SIGNAL, PAS$K_NEGWIDDIG
                                                                         .EXTRN  PAS$CVT_F_T, PAS$K_LINTOOLON
                                                                         .EXTRN  PAS$$END_WRITE

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

                                                01FC 00000               .ENTRY  PAS$WRITE_REALE_F, Save R2,R3,R4,R5,R6,R7,- ; 0144
                                                                                 R8
                58 00000000G   00    9E 00002               MOVAB   PAS$$SIGNAL, R8                                     ;
                5E             10    C2 00009               SUBL2   #16, SP                                             ;
                         04    AE    7C 0000C               CLRQ    ERROR_ADDR                                          ; 0196
                         0C    AE    D4 0000F               CLRL    PFV_ADDR                                            ;
                6D       0067  CF    DE 00012               MOVAL   4$, (FP)                                            ;
                04             6C    91 00017               CMPB    (AP), #4                                            ; 0216
                               05    1F 0001A               BLSSU   1$                                                  ;
04              AE       10    AC    D0 0001C               MOVL    ERROR, ERROR_ADDR                                   ; 0218
                56       04    AC    D0 00021 1$:           MOVL    PFV, R6                                             ; 0220
0C              AE             56    D0 00025               MOVL    R6, PFV_ADDR                                        ;
                   00000000G   00    16 00029               JSB     PAS$$VALIDATE_PFV                                   ; 0226
08              AE             01    D0 0002F               MOVL    #1, UNWIND_ACT                                      ; 0232
                   00000000G   00    16 00033               JSB     PAS$$INIT_WRITE                                     ; 0238
                6E       0C    AC    D0 00039               MOVL    TOTAL_WIDTH, FIELD_WIDTH                            ; 0244
                               0A    18 0003D               BGEQ    2$                                                  ; 0245
                               7E    D4 0003F               CLRL    -(SP)                                               ; 0247
                7E       00G   8F    9A 00041               MOVZBL  #PAS$K_NEGWIDDIG, -(SP)                             ;
```

```
                                 68              02  FB 00045          CALLS   #2, PAS$$SIGNAL                                    ;
                                                 04  00048          RET                                                         ;
                          52     F0   A7     EC  A7  C3 00049 2$:    SUBL3   -20(FCB), -16(FCB), REMAINING_WIDTH                 ; 0248
                                                 52  DD 0004F          PUSHL   REMAINING_WIDTH                                     ; 0258
                                              04 AE  9F 00051          PUSHAB  FIELD_WIDTH                                         ; 0255
                                              EC A7  DD 00054          PUSHL   -20(FCB)                                            ; 0256
                                              08 AC  9F 00057          PUSHAB  VALUE                                               ; 0255
                             00000000G        00     04  FB 0005A          CALLS   #4, PAS$CVT_F_T                                 ;
                                              0E     50  E8 00061          BLBS    R0, 3$                                          ;
                          7E                  6E     52  C3 00064          SUBL3   REMAINING_WIDTH, FIELD_WIDTH, -(SP)             ; 0260
                                                     01  DD 00068          PUSHL   #1                                              ;
                                              7E     00G 8F  9A 0006A          MOVZBL  #PAS$K_LINTOOLON, -(SP)                     ;
                                              68         03  FB 0006E          CALLS   #3, PAS$$SIGNAL                             ;
                                                         04  00071          RET                                                 ;
                                        EC    A7                 6E  C0 00072 3$:     ADDL2   FIELD_WIDTH, -20(FCB)           ; 0266
                                                    00000000G    00  16 00076          JSB     PAS$$END_WRITE                  ; 0273
                                                                 04  0007C          RET                                     ; 0277
                                                               0000  0007D 4$:        .WORD   Save nothing                    ; 0196
                                              50     08 AC  D0 0007F          MOVL    8(AP), R0                                   ;
                                              50     04 A0  D0 00083          MOVL    4(R0), R0                                   ;
                                              F4 A0  9F 00087          PUSHAB  ERROR_ADDR                                          ;
                                              F8 A0  9F 0008A          PUSHAB  UNWIND_ACT                                          ;
                                              FC A0  9F 0008D          PUSHAB  PFV_ADDR                                            ;
                                                 03  DD 00090          PUSHL   #3                                                  ;
                                                 5E  DD 00092          PUSHL   SP                                                  ;
                                              7E     04 AC  7D 00094          MOVQ    4(AP), -(SP)                                ;
                             00000000G        00         03  FB 00098          CALLS   #3, PAS$$IO_HANDLER                         ;
                                                         04  0009F          RET                                                 ;

; Routine Size:  160 bytes,    Routine Base:  _PAS$CODE + 0000

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

```
;  220          0280  1 %SBTTL 'PAS$WRITEV_REALE_F - Write F_floating in E format to string'
;  221          0281  1 GLOBAL ROUTINE PAS$WRITEV_REALE_F (
;  222          0282  1         MAX_LENGTH: WORD,                               ! Maximum length of string
;  223          0283  1         STRING_LINE: REF VECTOR [, WORD],               ! String to write to
;  224          0284  1         VALUE,                                          ! Value to write
;  225          0285  1         TOTAL_WIDTH: SIGNED,                            ! Total field width
;  226          0286  1         ERROR                                           ! Error unwind address
;  227          0287  1     ) : NOVALUE =
;  228          0288  1
;  229          0289  1 !++
;  230          0290  1 ! FUNCTIONAL DESCRIPTION:
;  231          0291  1 !
;  232          0292  1 !       This procedure writes an F_floating in E format
;  233          0293  1 !       to the specified string.
;  234          0294  1 !
;  235          0295  1 ! CALLING SEQUENCE:
;  236          0296  1 !
;  237          0297  1 !       CALL PAS$WRITEV_REALE_F (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
;  238          0298  1 !          VALUE.rf.v, TOTAL_WIDTH.r[.v [, ERROR.j.r])
;  239          0299  1 !
;  240          0300  1 ! FORMAL PARAMETERS:
;  241          0301  1 !
;  242          0302  1 !       MAX_LENGTH      - The maximum length of STRING_LINE.
;  243          0303  1 !
;  244          0304  1 !       STRING_LINE     - A varying string to which the output will be appended.
;  245          0305  1 !
;  246          0306  1 !       VALUE           - The value to write.                                    ;
;  247          0307  1 !
;  248          0308  1 !       TOTAL_WIDTH     - The width of the field to write.
;  249          0309  1 !
;  250          0310  1 !       ERROR           - Optional.  If specified, the address to unwind to      ;
;  251          0311  1 !                         in case of an error.
;  252          0312  1 !
;  253          0313  1 ! IMPLICIT INPUTS:
;  254          0314  1 !
;  255          0315  1 !       NONE
;  256          0316  1 !
;  257          0317  1 ! IMPLICIT OUTPUTS:
;  258          0318  1 !
;  259          0319  1 !       NONE
;  260          0320  1 !
;  261          0321  1 ! ROUTINE VALUE:
;  262          0322  1 !
;  263          0323  1 !       NONE
;  264          0324  1 !
;  265          0325  1 ! SIDE EFFECTS:
;  266          0326  1 !
;  267          0327  1 !       NONE
;  268          0328  1 !
;  269          0329  1 ! SIGNALLED ERRORS:
;  270          0330  1 !
;  271          0331  1 !       See PAS$WRITE_REALE_F
;  272          0332  1 !
;  273          0333  1 !--
;  274          0334  1
;  275          0335  2     BEGIN
;  276          0336  2
```

```
;  277          0337  2     LOCAL
;  278          0338  2         PFV: $PAS$PFV_FILE_VARIABLE,     ! Pascal File Variable
;  279          0339  2         ARG_LIST: VECTOR [4, LONG],      ! Argument list
;  280          0340  2         PFV_ADDR: VOLATILE,              ! Enable argument
;  281          0341  2         UNWIND_ACT: VOLATILE,            ! Enable argument
;  282          0342  2         ERROR_ADDR: VOLATILE;            ! Enable argument
;  283          0343  2
;  284          0344  2     BUILTIN
;  285          0345  2         ACTUALCOUNT;                     ! Count of arguments
;  286          0346  2
;  287          0347  2     ENABLE
;  288          0348  2         PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  289          0349  2
;  290          0350  2     !+
;  291          0351  2     ! Get ERROR parameter, if present.
;  292          0352  2     !-
;  293          0353  2
;  294          0354  2     IF ACTUALCOUNT () GEQU 5
;  295          0355  2     THEN
;  296          0356  2         ERROR_ADDR = .ERROR;             ! Set unwind address
;  297          0357  2
;  298          0358  2     PFV_ADDR = PFV [PFV$R_PFV];          ! Set PFV address
;  299          0359  2
;  300          0360  2     !+
;  301          0361  2     ! Set up ARG_LIST.
;  302          0362  2     !-
;  303          0363  2
;  304          0364  2     ARG_LIST [0] = 3;                    ! Three arguments
;  305          0365  2     ARG_LIST [1] = PFV [PFV$R_PFV];      ! PFV address
;  306          0366  2     ARG_LIST [2] = .VALUE;               ! Value to write
;  307          0367  2     ARG_LIST [3] = .TOTAL_WIDTH;         ! Field width
;  308          0368  2
;  309          0369  2     !+
;  310          0370  2     ! Call PAS$$DO_WRITEV to do the work, giving it the address of
;  311          0371  2     ! PAS$WRITE_REALE_F to call.
;  312          0372  2     !-
;  313          0373  2
;  314          0374  2     PAS$$DO_WRITEV (PFV [PFV$R_PFV], .MAX_LENGTH, STRING_LINE [0], ARG_LIST,
;  315          0375  2         PAS$WRITE_REALE_F);
;  316          0376  2
;  317          0377  2     RETURN;
;  318          0378  2
;  319          0379  1     END;                                 ! End of routine PAS$WRITEV_REALE_F


                                                 .EXTRN  PAS$$DO_WRITEV

                                    007C 00000   .ENTRY  PAS$WRITEV_REALE_F, Save R2,R3,R4,R5,R6      ; 0281
                              5E          28 C2 00002   SUBL2  #40, SP                                ;
                                          7E D4 00005   CLRL   ERROR_ADDR                             ; 0335
                                    04    AE 7C 00007   CLRQ   UNWIND_ACT                             ;
                              6D  0039    CF DE 0000A   MOVAL  2$, (FP)                               ;
                              05          6C 91 0000F   CMPB   (AP), #5                               ; 0354
                                          04 1F 00012   BLSSU  1$                                     ;
                              6E    14    AC D0 00014   MOVL   ERROR, ERROR_ADDR                      ; 0356
                        08    AE    1C    AE 9E 00018 1$: MOVAB  PFV, PFV_ADDR                        ; 0358
```

```
                            0C   AE            03   DO 0001D          MOVL    #3, ARG_LIST                          ; 0364
                            10   AE       1C   AE   9E 00021          MOVAB   PFV, ARG_LIST+4                       ; 0365
                            14   AE       0C   AC   7D 00026          MOVQ    VALUE, ARG_LIST+8                     ; 0366
                                 55     FF31   CF   9E 0002B          MOVAB   PAS$WRITE_REALE_F, R5                 ; 0374
                                 54       0C   AE   9E 00030          MOVAB   ARG_LIST, R4                          ;
                                 56       1C   AE   9E 00034          MOVAB   PFV, R6                               ;
                                 53       08   AC   D0 00038          MOVL    STRING_LINE, R3                       ;
                                 52       04   AC   3C 0003C          MOVZWL  MAX_LENGTH, R2                        ;
                                    00000000G   00   16 00040          JSB     PAS$$DO_WRITEV                        ;
                                                04 00046              RET                                           ; 0379
                                              0000 00047 2$:          .WORD   Save nothing                          ; 0335
                                 50       08   AC   D0 00049          MOVL    8(AP), R0                             ;
                                 50       04   A0   D0 0004D          MOVL    4(R0), R0                             ;
                                          D4   A0   9F 00051          PUSHAB  ERROR_ADDR                            ;
                                          D8   A0   9F 00054          PUSHAB  UNWIND_ACT                            ;
                                          DC   A0   9F 00057          PUSHAB  PFV_ADDR                              ;
                                               03   DD 0005A          PUSHL   #3                                    ;
                                               5E   DD 0005C          PUSHL   SP                                    ;
                                 7E       04   AC   7D 0005E          MOVQ    4(AP), -(SP)                          ;
                         00000000G   00        03   FB 00062          CALLS   #3, PAS$$IO_HANDLER                   ;
                                               04 00069              RET                                           ;

; Routine Size:  106 bytes,    Routine Base:  _PAS$CODE + 00A0

;  320          0380  1
;  321          0381  1 !<BLF/PAGE>
```

```
;  323          0382  1 END                                   ! End of module PAS$WRITE_REALE_F
;  324          0383  1
;  325          0384  0 ELUDOM
```

```
;                                   PSECT SUMMARY

;       Name                    Bytes                           Attributes

;   _PAS$CODE                       266  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         0         0         581      00:01.0
;   _$255$DUA28:[PASRTL.OBJ]PASLIB.L32;1     427        97        22          33      00:00.4
```

```
;                                   COMMAND QUALIFIERS

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

;  326          0385  0
; Size:         266 code + 0 data bytes
; Run Time:       00:07.0
; Elapsed Time:   00:18.1
; Lines/CPU Min:   3285
; Lexemes/CPU-Min: 12751
; Memory Used:  80 pages
; Compilation Complete
```

PASWRIREH LIS

PASWRITEL LIS

PASWRIREG LIS

PATDEF MDL

PASWRISTR LIS

PATCH

PASWRIREF LIS

PATCH MAP

SRMDEF MDL

PASWRIRFH LIS

PASWRIVAR LIS

PASWRIRED LIS

PASWRIUNS LIS

PASWRIINT LIS

PASWRIOCT LIS

PASWRIRFG LIS

BSTRUC REQ

PASWRIRFF LIS

CHRKEY REQ

PASWRIRFD LIS