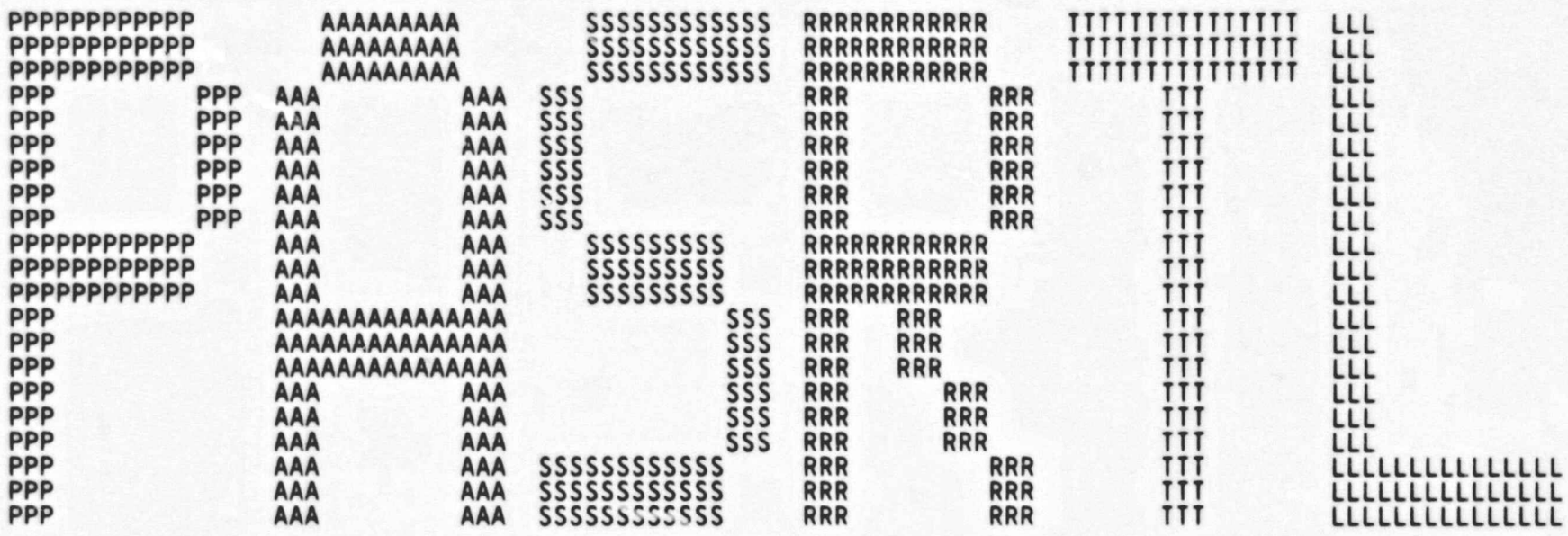
PPPPPPPPPPPP AAAAAAAAA SSSSSSSSSSS RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
PPP PPP AAA AAA SSS RRR RRR TTT LLL
PPPPPPPPPPPP AAAAAAAAAAAAAAA SSS RRR RRR TTT LLL
PPP AAA AAA SSSSSSSSS RRRRRRRRRRR TTT LLL
PPP AAAAAAAAAAAAAAA SSS RRR RRR TTT LLL
PPP AAA AAA SSS RRR RRR TTT LLL
PPP AAA AAA SSSSSSSSSSS RRR RRR TTT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

**FILE**ID**PASWRIRFF

```
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR     IIIIII    RRRRRRRR    FFFFFFFFFF  FFFFFFFFFF
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR     IIIIII    RRRRRRRR    FFFFFFFFFF  FFFFFFFFFF
PP       PP  AA      AA  SS        WW      WW  RR      RR     II      RR      RR  FF          FF
PP       PP  AA      AA  SS        WW      WW  RR      RR     II      RR      RR  FF          FF
PP       PP  AA      AA  SS        WW      WW  RR      RR     II      RR      RR  FF          FF
PP       PP  AA      AA  SS        WW      WW  RR      RR     II      RR      RR  FF          FF
PPPPPPPP     AA      AA    SSSSSS  WW      WW  RRRRRRRR       II      RRRRRRRR    FFFFFFFF    FFFFFFFF
PPPPPPPP     AA      AA    SSSSSS  WW      WW  RRRRRRRR       II      RRRRRRRR    FFFFFFFF    FFFFFFFF
PP           AAAAAAAAAA          SS  WW  WW  WW  RR  RR        II      RR  RR      FF          FF
PP           AAAAAAAAAA          SS  WW  WW  WW  RR  RR        II      RR  RR      FF          FF
PP           AA      AA          SS  WWWW  WWWW  RR    RR      II      RR    RR    FF          FF
PP           AA      AA          SS  WWWW  WWWW  RR    RR      II      RR    RR    FF          FF
PP           AA      AA  SSSSSSSS    WW      WW  RR      RR  IIIIII    RR      RR  FF          FF
PP           AA      AA  SSSSSSSS    WW      WW  RR      RR  IIIIII    RR      RR  FF          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_REALF_F ( %TITLE 'Write an F_floating in F format'
;     2          0002  0                 IDENT = '1-002'                 ! File: PASWRIRFF.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 !       fixed-point notation 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$WRITE_REALF_F: NOVALUE,              ! Write to textfile
;    62          0124  1     PAS$WRITEV_REALF_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$WRITE_REALF_F - Write F_floating in F format to textfile'
;    83          0144  1 GLOBAL ROUTINE PAS$WRITE_REALF_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           FRAC_DIGITS: SIGNED,                            ! Digits in fraction
;    88          0149  1           ERROR                                           ! Error unwind address
;    89          0150  1       ): NOVALUE =
;    90          0151  1
;    91          0152  1 !++
;    92          0153  1 ! FUNCTIONAL DESCRIPTION:
;    93          0154  1 !
;    94          0155  1 !       This procedure writes an F_floating value in fixed-point notation
;    95          0156  1 !       to the specified textfile.
;    96          0157  1 !
;    97          0158  1 ! CALLING SEQUENCE:
;    98          0159  1 !
;    99          0160  1 !       CALL PAS$WRITE_REALF_F (PFV.mr.r, VALUE.rf.v , TOTAL_WIDTH.rl.v,
;   100          0161  1 !                               FRAC_DIGITS.rl.v [, ERROR.ja.r]])
;   101          0162  1 !
;   102          0163  1 ! FORMAL PARAMETERS:
;   103          0164  1 !
;   104          0165  1 !       PFV             - The Pascal File Variable (PFV) passed by reference.
;   105          0166  1 !                         The structure of the PFV is defined in PASPFV.REQ.
;   106          0167  1 !
;   107          0168  1 !       VALUE           - The F_floating value to write.
;   108          0169  1 !
;   109          0170  1 !       TOTAL_WIDTH     - Total field width.
;   110          0171  1 !
;   111          0172  1 !       FRAC_DIGITS     - Number of digits in fraction.
;   112          0173  1 !
;   113          0174  1 !       ERROR           - Optional.  Address to unwind to if an error occurs.
;   114          0175  1 !
;   115          0176  1 ! IMPLICIT INPUTS:
;   116          0177  1 !
;   117          0178  1 !       NONE
;   118          0179  1 !
;   119          0180  1 ! IMPLICIT OUTPUTS:
;   120          0181  1 !
;   121          0182  1 !       NONE
;   122          0183  1 !
;   123          0184  1 ! ROUTINE VALUE:
;   124          0185  1 !
;   125          0186  1 !       NONE
;   126          0187  1 !
;   127          0188  1 ! SIDE EFFECTS:
;   128          0189  1 !
;   129          0190  1 !       If the file is the standard file OUTPUT, it is implicitly opened.
;   130          0191  1 !
;   131          0192  1 ! SIGNALLED ERRORS:
;   132          0193  1 !
;   133          0194  1 !       LINTOOLON - line too long
;   134          0195  1 !       NEGWIDDIG - negative Width or Digits specification is not allowed
;   135          0196  1 !
;   136          0197  1 !--
;   137          0198  1
;   138          0199  2     BEGIN
```

```
;  139     0200  2
;  140     0201  2      BUILTIN
;  141     0202  2          ACTUALCOUNT;
;  142     0203  2
;  143     0204  2      LOCAL
;  144     0205  2          FCB: REF $PAS$FCB_CONTROL_BLOCK,                    ! File control block
;  145     0206  2          FIELD_WIDTH,                                        ! Minimum/actual field width
;  146     0207  2          REMAINING_WIDTH,                                    ! Width remaining on line
;  147     0208  2          PFV_ADDR: VOLATILE,                                 ! Enable argument
;  148     0209  2          UNWIND_ACT: VOLATILE,                               ! Enable argument
;  149     0210  2          ERROR_ADDR: VOLATILE;                               ! Enable argument
;  150     0211  2
;  151     0212  2      ENABLE
;  152     0213  2          PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  153     0214  2
;  154     0215  2      !+
;  155     0216  2      ! Get ERROR parameter, if present.
;  156     0217  2      !-
;  157     0218  2
;  158     0219  2      IF ACTUALCOUNT () GEQU 5
;  159     0220  2      THEN
;  160     0221  2          ERROR_ADDR = .ERROR;            ! Set unwind address
;  161     0222  2
;  162     0223  2      PFV_ADDR = PFV [PFV$R_PFV];         ! Set PFV address
;  163     0224  2
;  164     0225  2      !+
;  165     0226  2      ! Validate PFV and get PFV.
;  166     0227  2      !-
;  167     0228  2
;  168     0229  2      PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
;  169     0230  2
;  170     0231  2      !+
;  171     0232  2      ! Set unwind action to unlock file.
;  172     0233  2      !-
;  173     0234  2
;  174     0235  2      UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
;  175     0236  2
;  176     0237  2      !+
;  177     0238  2      ! Do common initialization.
;  178     0239  2      !-
;  179     0240  2
;  180     0241  2      PAS$$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
;  181     0242  2
;  182     0243  2      !+
;  183     0244  2      ! Get minimum and maximum field widths.  Check for valid field width
;  184     0245  2      ! and number of digits.
;  185     0246  2      !-
;  186     0247  2
;  187     0248  2      FIELD_WIDTH = .TOTAL_WIDTH;
;  188     0249  3      IF (.FIELD_WIDTH LSS 0) OR (.FRAC_DIGITS LSS 0)
;  189     0250  2      THEN
;  190     0251  2          $PAS$IO_ERROR (PAS$_NEGWIDDIG,0);
;  191     0252  2      REMAINING_WIDTH = .FCB [FCB$A_RECORD_END] - .FCB [FCB$A_RECORD_CUR];
;  192     0253  2
;  193     0254  2      !+
;  194     0255  2      ! Do the convert.  If it fails, signal an error.
;  195     0256  2      !-
```

```
;  196          0257  2
;  197          0258  2      IF NOT PAS$CVT_F_T (VALUE,                      ! Value to convert
;  198          0259  2                         .FCB [FCB$A_RECORD_CUR],    ! Destination
;  199          0260  2                         FIELD_WIDTH,                ! Minimum/actual width
;  200          0261  2                         .REMAINING_WIDTH,           ! Maximum width
;  201          0262  2                         .FRAC_DIGITS)               ! Fraction digits
;  202          0263  2      THEN
;  203          0264  2          $PAS$IO_ERROR (PAS$_LINTOOLON,1,(.FIELD_WIDTH-.REMAINING_WIDTH));
;  204          0265  2
;  205          0266  2      !+
;  206          0267  2      ! Update buffer pointer.
;  207          0268  2      !-
;  208          0269  2
;  209          0270  2      FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_CUR] + .FIELD_WIDTH;
;  210          0271  2
;  211          0272  2      !+
;  212          0273  2      ! Call WRITE epilogue routine to move the last character written to the
;  213          0274  2      ! user's buffer and to unlock the file variable.
;  214          0275  2      !-
;  215          0276  2
;  216          0277  2      PAS$$END_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
;  217          0278  2
;  218          0279  2      RETURN;
;  219          0280  2
;  220          0281  1      END;                                           ! End of routine PAS$WRITE_REALF_F


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

                                                                .EXTRN  PAS$WRITE_REALF_F
                                                                .EXTRN  PAS$WRITEV_REALF_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_REALF_F, Save R2,R3,R4,R5,R6,R7,- ; 0144
                                                                        R8
                58 00000000G    00  9E 00002                  MOVAB   PAS$$SIGNAL, R8                             ;
                5E                  10  C2 00009              SUBL2   #16, SP                                     ;
                        04      AE  7C 0000C                  CLRQ    ERROR_ADDR                                  ; 0199
                        0C      AE  D4 0000F                  CLRL    PFV_ADDR                                    ;
                6D          006F    CF  DE 00012              MOVAL   5$, (FP)                                    ;
                05                  6C  91 00017              CMPB    (AP), #5                                    ; 0219
                                    05  1F 0001A              BLSSU   1$                                          ;
04      AE              14      AC  D0 0001C                  MOVL    ERROR, ERROR_ADDR                           ; 0221
56                      04      AC  D0 00021  1$:             MOVL    PFV, R6                                     ; 0223
0C      AE                      56  D0 00025                  MOVL    R6, PFV_ADDR                                ;
            00000000G           00  16 00029                  JSB     PAS$$VALIDATE_PFV                           ; 0229
08      AE                      01  D0 0002F                  MOVL    #1, UNWIND_ACT                              ; 0235
            00000000G           00  16 00033                  JSB     PAS$$INIT_WRITE                             ; 0241
```

```
                        6E      0C  AC  D0 00039          MOVL    TOTAL_WIDTH, FIELD_WIDTH                ; 0248
                                05  19  0003D          BLSS    2$                                      ; 0249
                                10  AC  D5 0003F          TSTL    FRAC_DIGITS
                                0A  18  00042          BGEQ    3$
                                7E  D4  00044 2$:      CLRL    -(SP)                                   ; 0251
                        7E      00G 8F  9A  00046      MOVZBL  #PAS$K_NEGWIDDIG, -(SP)
                        68          02  FB  0004A      CALLS   #2, PAS$$SIGNAL
                                    04  0004D          RET
                52      F0  A7      EC  A7  C3 0004E 3$:   SUBL3   -20(FCB), -16(FCB), REMAINING_WIDTH      ; 0252
                                10  AC  DD  00054      PUSHL   FRAC_DIGITS                             ; 0262
                                52  DD  00057          PUSHL   REMAINING_WIDTH                         ; 0261
                                08  AE  9F  00059      PUSHAB  FIELD_WIDTH                             ; 0258
                                EC  A7  DD  0005C      PUSHL   -20(FCB)                                ; 0259
                                08  AC  9F  0005F      PUSHAB  VALUE                                   ; 0258
                    00000000G   00      05  FB  00062  CALLS   #5, PAS$CVT_F_T
                                0E      50  E8  00069  BLBS    R0, 4$
                7E              6E      52  C3  0006C  SUBL3   REMAINING_WIDTH, FIELD_WIDTH, -(SP)       ; 0264
                                    01  DD  00070      PUSHL   #1
                                7E      00G 8F  9A  00072  MOVZBL  #PAS$K_LINTOOLON, -(SP)
                                68          03  FB  00076  CALLS   #3, PAS$$SIGNAL
                                            04  00079  RET
                            EC  A7          6E  C0  0007A 4$:  ADDL2   FIELD_WIDTH, -20(FCB)               ; 0270
                                    00000000G   00  16  0007E  JSB     PAS$$END_WRITE                      ; 0277
                                                04  00084  RET                                            ; 0281
                                                0000 00085 5$:  .WORD   Save nothing                       ; 0199
                                50      08  AC  D0  00087  MOVL    8(AP), R0
                                50      04  A0  D0  0008B  MOVL    4(R0), R0
                                        F4  A0  9F  0008F  PUSHAB  ERROR_ADDR
                                        F8  A0  9F  00092  PUSHAB  UNWIND_ACT
                                        FC  A0  9F  00095  PUSHAB  PFV_ADDR
                                            03  DD  00098  PUSHL   #3
                                            5E  DD  0009A  PUSHL   SP
                                    7E      04  AC  7D  0009C  MOVQ    4(AP), -(SP)
                            00000000G   00      03  FB  000A0  CALLS   #3, PAS$$IO_HANDLER
                                                04  000A7  RET

; Routine Size:  168 bytes,    Routine Base:  _PAS$CODE + 0000

;  221          0282  1
;  222          0283  1 !<BLF/PAGE>
```

```
;  224     0284  1 %SBTTL 'PAS$WRITEV_REALF_F - Write F_floating in F format to string'
;  225     0285  1 GLOBAL ROUTINE PAS$WRITEV_REALF_F (
;  226     0286  1         MAX_LENGTH: WORD,                               ! Maximum length of string
;  227     0287  1         STRING_LINE: REF VECTOR [, WORD],               ! String to write to
;  228     0288  1         VALUE,                                          ! Value to write
;  229     0289  1         TOTAL_WIDTH: WORD SIGNED,                       ! Total field width
;  230     0290  1         FRAC_DIGITS: SIGNED,                            ! Number of fraction digits
;  231     0291  1         ERROR                                           ! Error unwind address
;  232     0292  1     ) : NOVALUE =
;  233     0293  1
;  234     0294  1 !++
;  235     0295  1 ! FUNCTIONAL DESCRIPTION:
;  236     0296  1 !
;  237     0297  1 !       This procedure writes an F_floating in fixed-point format
;  238     0298  1 !       to the specified string.
;  239     0299  1 !
;  240     0300  1 ! CALLING SEQUENCE:
;  241     0301  1 !
;  242     0302  1 !       CALL PAS$WRITEV_REALF_F (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
;  243     0303  1 !          VALUE.rf.v, TOTAL_WIDTH.rw.v, FRAC_DIGITS.rl.v [, ERROR.j.r])
;  244     0304  1 !
;  245     0305  1 ! FORMAL PARAMETERS:
;  246     0306  1 !
;  247     0307  1 !       MAX_LENGTH      - The maximum length of STRING_LINE.
;  248     0308  1 !
;  249     0309  1 !       STRING_LINE     - A varying string to which the output will be appended.
;  250     0310  1 !
;  251     0311  1 !       VALUE           - The value to write.
;  252     0312  1 !
;  253     0313  1 !       TOTAL_WIDTH     - The width of the field to write.
;  254     0314  1 !
;  255     0315  1 !       FRAC_DIGITS     - The number of digits in the fraction field.
;  256     0316  1 !
;  257     0317  1 !       ERROR           - Optional.  If specified, the address to unwind to
;  258     0318  1 !                         in case of an error.
;  259     0319  1 !
;  260     0320  1 ! IMPLICIT INPUTS:
;  261     0321  1 !
;  262     0322  1 !       NONE
;  263     0323  1 !
;  264     0324  1 ! IMPLICIT OUTPUTS:
;  265     0325  1 !
;  266     0326  1 !       NONE
;  267     0327  1 !
;  268     0328  1 ! ROUTINE VALUE:
;  269     0329  1 !
;  270     0330  1 !       NONE
;  271     0331  1 !
;  272     0332  1 ! SIDE EFFECTS:
;  273     0333  1 !
;  274     0334  1 !       NONE
;  275     0335  1 !
;  276     0336  1 ! SIGNALLED ERRORS:
;  277     0337  1 !
;  278     0338  1 !       See PAS$WRITE_REALF_F
;  279     0339  1 !
;  280     0340  1 !--
```

```
;  281          0341  1
;  282          0342  2      BEGIN
;  283          0343  2
;  284          0344  2      LOCAL
;  285          0345  2          PFV: $PAS$PFV_FILE_VARIABLE,     ! Pascal File Variable
;  286          0346  2          ARG_LIST: VECTOR [5, LONG],      ! Argument list
;  287          0347  2          PFV_ADDR: VOLATILE,              ! Enable argument
;  288          0348  2          UNWIND_ACT: VOLATILE,            ! Enable argument
;  289          0349  2          ERROR_ADDR: VOLATILE;            ! Enable argument
;  290          0350  2
;  291          0351  2      BUILTIN
;  292          0352  2          ACTUALCOUNT;                     ! Count of arguments
;  293          0353  2
;  294          0354  2      ENABLE
;  295          0355  2          PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  296          0356  2
;  297          0357  2      !+
;  298          0358  2      ! Get ERROR parameter, if present.
;  299          0359  2      !-
;  300          0360  2
;  301          0361  2      IF ACTUALCOUNT () GEQU 6
;  302          0362  2      THEN
;  303          0363  2          ERROR_ADDR = .ERROR;             ! Set unwind address
;  304          0364  2
;  305          0365  2      PFV_ADDR = PFV [PFV$R_PFV];          ! Set PFV address
;  306          0366  2
;  307          0367  2      !+
;  308          0368  2      ! Set up ARG_LIST.
;  309          0369  2      !-
;  310          0370  2
;  311          0371  2      ARG_LIST [0] = 4;                    ! Four arguments
;  312          0372  2      ARG_LIST [1] = PFV [PFV$R_PFV];      ! PFV address
;  313          0373  2      ARG_LIST [2] = .VALUE;               ! Value to write
;  314          0374  2      ARG_LIST [3] = .TOTAL_WIDTH;         ! Width of field
;  315          0375  2      ARG_LIST [4] = .FRAC_DIGITS;         ! Fraction digits
;  316          0376  2
;  317          0377  2      !+
;  318          0378  2      ! Call PAS$$DO_WRITEV to do the work, giving it the address of
;  319          0379  2      ! PAS$WRITE_REALF_F to call.
;  320          0380  2      !-
;  321          0381  2
;  322          0382  2      PAS$$DO_WRITEV (PFV [PFV$R_PFV], .MAX_LENGTH, STRING_LINE [0], ARG_LIST,
;  323          0383  2          PAS$WRITE_REALF_F);
;  324          0384  2
;  325          0385  2      RETURN;
;  326          0386  2
;  327          0387  1      END;                                 ! End of routine PAS$WRITEV_REALF_F


                                                  .EXTRN  PAS$$DO_WRITEV

                                     007C 00000   .ENTRY  PAS$WRITEV_REALF_F, Save R2,R3,R4,R5,R6     ; 0285
                              5E           2C  C2 00002   SUBL2  #44, SP                                ;
                                           7E  D4 00005   CLRL   ERROR_ADDR                             ; 0342
                                        04 AE  7C 00007   CLRQ   UNWIND_ACT                             ;
                              6D      0043  CF  DE 0000A   MOVAL  2$, (FP)                              ;
```

```
                                 06              6C  91 0000F          CMPB    (AP), #6                                ; 0361
                                                 04  1F 00012          BLSSU   1$                                      ;
                                 6E        18    AC  D0 00014          MOVL    ERROR, ERROR_ADDR                       ; 0363
                           08    AE        20    AE  9E 00018 1$:      MOVAB   PFV, PFV_ADDR                           ; 0365
                           0C    AE              04  D0 0001D          MOVL    #4, ARG_LIST                            ; 0371
                           10    AE        20    AE  9E 00021          MOVAB   PFV, ARG_LIST+4                         ; 0372
                           14    AE        0C    AC  D0 00026          MOVL    VALUE, ARG_LIST+8                       ; 0373
                           18    AE        10    AC  32 0002B          CVTWL   TOTAL_WIDTH, ARG_LIST+12                ; 0374
                           1C    AE        14    AC  D0 00030          MOVL    FRAC_DIGITS, ARG_LIST+16                ; 0375
                                 55      FF1F    CF  9E 00035          MOVAB   PAS$WRITE_REALF_F, R5                   ; 0382
                                 54        0C    AE  9E 0003A          MOVAB   ARG_LIST, R4                            ;
                                 56        20    AE  9E 0003E          MOVAB   PFV, R6                                 ;
                                 53        08    AC  D0 00042          MOVL    STRING_LINE, R3                         ;
                                 52        04    AC  3C 00046          MOVZWL  MAX_LENGTH, R2                          ;
                                    00000000G    00  16 0004A          JSB     PAS$$DO_WRITEV                          ;
                                                 04  00050          RET                                             ; 0387
                                              0000  00051 2$:       .WORD   Save nothing                            ; 0342
                                 50        08    AC  D0 00053          MOVL    8(AP), R0                               ;
                                 50        04    A0  D0 00057          MOVL    4(R0), R0                               ;
                                           D0    A0  9F 0005B          PUSHAB  ERROR_ADDR                              ;
                                           D4    A0  9F 0005E          PUSHAB  UNWIND_ACT                              ;
                                           D8    A0  9F 00061          PUSHAB  PFV_ADDR                                ;
                                                 03  DD 00064          PUSHL   #3                                      ;
                                                 5E  DD 00066          PUSHL   SP                                      ;
                                 7E        04    AC  7D 00068          MOVQ    4(AP), -(SP)                            ;
                        00000000G  00              03  FB 0006C          CALLS   #3, PAS$$IO_HANDLER                     ;
                                                 04  00073          RET                                             ;

; Routine Size:  116 bytes,    Routine Base:  _PAS$CODE + 00A8

;  328          0388  1
;  329          0389  1 !<BLF/PAGE>
```

```
;  331          0390  1 END                                          ! End of module PAS$WRITE_REALF_F
;  332          0391  1
;  333          0392  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
;       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$:PASWRIRFF/OBJ=OBJ$:PASWRIRFF MSRC$:PASWRIRFF/UPDATE=(ENH$:PASWRIRFF
;       )

; Size:          284 code + 0 data bytes
; Run Time:          00:07.4
; Elapsed Time:      00:17.4
; Lines/CPU Min:      3178
; Lexemes/CPU-Min: 12624
; Memory Used:   82 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

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