G 6

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

```
_$2

Sym
---

PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS

PAS

PAS

PAS

PAS
PAS
PAS
PAS
PAS

PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
```

```
**FILE**ID**PASWRIRFF                                   K 7
```

```
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
```

```
0 MODULE PAS$WRITE_REALF_F ( %TITLE 'Write an F_floating in F format'
0                IDENT = '1-002'                     ! File: PASWRIRFF.B32 Edit: SBL1002
0                ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
1 !*  ALL RIGHTS RESERVED.                                                     *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     Pascal Language Support
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains a procedure which writes an F_floating in
1 !       fixed-point notation to a textfile.
1 !
1 ! ENVIRONMENT:  User mode - AST reentrant
1 !
1 ! AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  SBL 1-April-1981
1 ! 1-002 - Make total-width a longword.  SBL 30-June-1982
1 !--
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 !
```

```
%SBTTL 'PAS$WRITE_REALF_F - Write F_floating in F format to textfile'
GLOBAL ROUTINE PAS$WRITE_REALF_F (
        PFV: REF $PAS$PFV_FILE_VARIABLE,                ! File variable
        VALUE,                                          ! Value to write
        TOTAL_WIDTH: SIGNED,                            ! Total field width
        FRAC_DIGITS: SIGNED,                            ! Digits in fraction
        ERROR                                           ! Error unwind address
    ): NOVALUE =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This procedure writes an F_floating value in fixed-point notation
!       to the specified textfile.
!
! CALLING SEQUENCE:
!
!       CALL PAS$WRITE_REALF_F (PFV.mr.r, VALUE.rf.v , TOTAL_WIDTH.rl.v,
!                               FRAC_DIGITS.rl.v [, ERROR.ja.r]])
!
! FORMAL PARAMETERS:
!
!       PFV             - The Pascal File Variable (PFV) passed by reference.
!                         The structure of the PFV is defined in PASPFV.REQ.
!
!       VALUE           - The F_floating value to write.
!
!       TOTAL_WIDTH     - Total field width.
!
!       FRAC_DIGITS     - Number of digits in fraction.
!
!       ERROR           - Optional.  Address to unwind to if an error occurs.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
!
!       NONE
!
! SIDE EFFECTS:
!
!       If the file is the standard file OUTPUT, it is implicitly opened.
!
! SIGNALLED ERRORS:
!
!       LINTOOLON - line too long
!       NEGWIDDIG - negative Width or Digits specification is not allowed
!
!--

    BEGIN
```

```
2
2       BUILTIN
2           ACTUALCOUNT;
2
2       LOCAL
2           FCB: REF $PAS$FCB_CONTROL_BLOCK,                ! File control block
2           FIELD_WIDTH,                                    ! Minimum/actual field width
2           REMAINING_WIDTH,                                ! Width remaining on line
2           PFV_ADDR: VOLATILE,                             ! Enable argument
2           UNWIND_ACT: VOLATILE,                           ! Enable argument
2           ERROR_ADDR: VOLATILE;                           ! Enable argument
2
2       ENABLE
2           PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
2
2       !+
2       ! Get ERROR parameter, if present.
2       !-
2
2       IF ACTUALCOUNT () GEQU 5
2       THEN
2           ERROR_ADDR = .ERROR;                ! Set unwind address
2
2       PFV_ADDR = PFV [PFV$R_PFV];             ! Set PFV address
2
2       !+
2       ! Validate PFV and get PFV.
2       !-
2
2       PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
2
2       !+
2       ! Set unwind action to unlock file.
2       !-
2
2       UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
2
2       !+
2       ! Do common initialization.
2       !-
2
2       PAS$$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
2
2       !+
2       ! Get minimum and maximum field widths.  Check for valid field width
2       ! and number of digits.
2       !-
2
2       FIELD_WIDTH = .TOTAL_WIDTH;
3       IF (.FIELD_WIDTH LSS 0) OR (.FRAC_DIGITS LSS 0)
2       THEN
2           $PAS$IO_ERROR (PAS$_NEGWIDDIG,0);
2       REMAINING_WIDTH = .FCB [FCB$A_RECORD_END] - .FCB [FCB$A_RECORD_CUR];
2
2       !+
2       ! Do the convert.  If it fails, signal an error.
2       !-
```

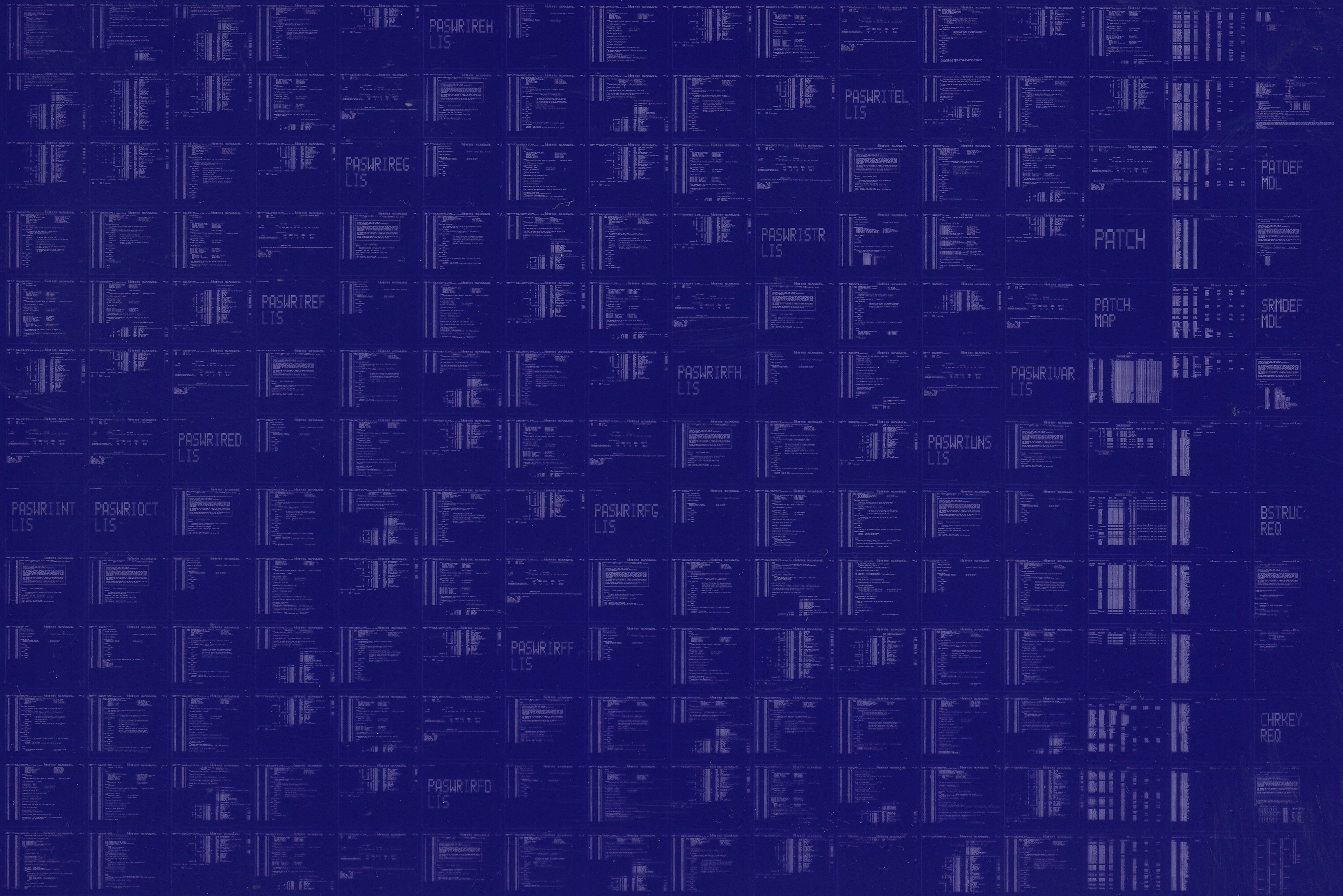
```
;  196          0257  2
;  197          0258  2      IF NOT PAS$CVT_F_T (VALUE,                   ! Value to convert
;  198          0259  2                          .FCB [FCB$A_RECORD_CUR],! Destination
;  199          0260  2                          FIELD_WIDTH,            ! Minimum/actual width
;  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 000C2          MOVAB   PAS$$SIGNAL, R8                             ;
                5E            10   C2 00009          SUBL2   #16, SP                                     ;
                        04    AE   7C 000OC          CLRQ    ERROR_ADDR                                  ; 0199
                        0C    AE   D4 0000F          CLRL    PFV_ADDR                                    ;
                6D      006F  CF   DE 00012          MOVAL   5$, (FP)                                    ;
                05            6C   91 00017          CMPB    (AP), #5                                    ; 0219
                              05   1F 0001A          BLSSU   1$                                          ;
           04   AE      14    AC   D0 0001C          MOVL    ERROR, ERROR_ADDR                           ; 0221
                56      04    AC   D0 00021 1$:      MOVL    PFV, R6                                     ; 0223
           0C   AE            56   D0 00025          MOVL    R6, PFV_ADDR                                ;
                   00000000G  00   16 00029          JSB     PAS$$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>
;  1 %SBTTL 'PAS$WRITEV_REALF_F - Write F_floating in F format to string'
;  1 GLOBAL ROUTINE PAS$WRITEV_REALF_F (
;  1         MAX_LENGTH: WORD,                           ! Maximum length of string
;  1         STRING_LINE: REF VECTOR [, WORD],           ! String to write to
;  1         VALUE,                                      ! Value to write
;  1         TOTAL_WIDTH: WORD SIGNED,                   ! Total field width
;  1         FRAC_DIGITS: SIGNED,                        ! Number of fraction digits
;  1         ERROR                                       ! Error unwind address
;  1     ) : NOVALUE =
;  1
;  1 !++
;  1 ! FUNCTIONAL DESCRIPTION:
;  1 !
;  1 !       This procedure writes an F_floating in fixed-point format
;  1 !       to the specified string.
;  1 !
;  1 ! CALLING SEQUENCE:
;  1 !
;  1 !       CALL PAS$WRITEV_REALF_F (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
;  1 !          VALUE.rf.v, TOTAL_WIDTH.rw.v, FRAC_DIGITS.rl.v [, ERROR.j.r])
;  1 !
;  1 ! FORMAL PARAMETERS:
;  1 !
;  1 !       MAX_LENGTH      - The maximum length of STRING_LINE.
;  1 !
;  1 !       STRING_LINE     - A varying string to which the output will be appended.
;  1 !
;  1 !       VALUE           - The value to write.
;  1 !
;  1 !       TOTAL_WIDTH     - The width of the field to write.
;  1 !
;  1 !       FRAC_DIGITS     - The number of digits in the fraction field.
;  1 !
;  1 !       ERROR           - Optional.  If specified, the address to unwind to
;  1 !                         in case of an error.
;  1 !
;  1 ! IMPLICIT INPUTS:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! IMPLICIT OUTPUTS:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! ROUTINE VALUE:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! SIDE EFFECTS:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! SIGNALLED ERRORS:
;  1 !
;  1 !       See PAS$WRITE_REALF_F
;  1 !
;  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
```

0298 AH-BT13A-SE
VAX/VMS V4.0

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