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
```

```
                                                                G 9
**FILE**ID**PASWRIRFH
```

```
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR      IIIIII    RRRRRRRR    FFFFFFFFFF  HH      HH
PPPPPPPP      AAAAAA      SSSSSSSS  WW      WW  RRRRRRRR      IIIIII    RRRRRRRR    FFFFFFFFFF  HH      HH
PP      PP  AA      AA  SS          WW      WW  RR      RR      II      RR      RR  FF          HH      HH
PP      PP  AA      AA  SS          WW      WW  RR      RR      II      RR      RR  FF          HH      HH
PP      PP  AA      AA  SS          WW      WW  RR      RR      II      RR      RR  FF          HH      HH
PP      PP  AA      AA  SS          WW      WW  RR      RR      II      RR      RR  FF          HH      HH
PPPPPPPP    AA      AA    SSSSSS    WW      WW  RRRRRRRR        II      RRRRRRRR    FFFFFFFF    HHHHHHHHHH
PPPPPPPP    AA      AA    SSSSSS    WW      WW  RRRRRRRR        II      RRRRRRRR    FFFFFFFF    HHHHHHHHHH
PP          AAAAAAAAAA          SS  WW  WW  WW  RR  RR          II      RR  RR      FF          HH      HH
PP          AAAAAAAAAA          SS  WW  WW  WW  RR  RR          II      RR  RR      FF          HH      HH
PP          AA      AA          SS  WWWW  WWWW  RR    RR        II      RR    RR    FF          HH      HH      ....
PP          AA      AA          SS  WWWW  WWWW  RR    RR        II      RR    RR    FF          HH      HH      ....
PP          AA      AA  SSSSSSSS    WW      WW  RR      RR    IIIIII    RR      RR  FF          HH      HH      ....
PP          AA      AA  SSSSSSSS    WW      WW  RR      RR    IIIIII    RR      RR  FF          HH      HH      ....

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_H ( %TITLE 'Write an H_floating in F format'
0                   IDENT = '1-002'                    ! File: PASWRIRFH.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 procedures which write an H_floating
1 !       in fixed-point notation to a textfile or string.
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
```

```
1 %SBTTL 'Declarations'
1 !
1 ! PROLOGUE DEFINITIONS:
1 !
1
1 REQUIRE 'RTLIN:PASPROLOG';                    ! Externals, linkages, PSECTs, structures
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     PAS$WRITE_REALF_H: NOVALUE,               ! Write to textfile
1     PAS$WRITEV_REALF_H: NOVALUE;              ! Write to string
1
1 !
1 ! MACROS:
1 !
1 !       NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1 !       NONE
1 !
1 ! FIELDS:
1 !
1 !       NONE
1 !
1 ! OWN STORAGE:
1 !
1 !       NONE
1 !
```

```
1 %SBTTL 'PAS$WRITE_REALF_H - Write H_floating in F format'
1 GLOBAL ROUTINE PAS$WRITE_REALF_H (
1         PFV: REF $PAS$PFV_FILE_VARIABLE,              ! File variable
1         VALUE_0,VALUE_1,VALUE_2,VALUE_3,              ! Value to write
1         TOTAL_WIDTH: SIGNED,                          ! Total field width
1         FRAC_DIGITS: SIGNED,                          ! Digits in fraction
1         ERROR                                         ! Error unwind address
1     ): NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This procedure writes an H_floating value in fixed-point notation
1 !       to the specified textfile.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       CALL PAS$WRITE_REALF_H (PFV.mr.r, VALUE.rh.v , TOTAL_WIDTH.rl.v
1 !                               , FRAC_DIGITS.rl.v [, ERROR.ja.r]])
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       PFV             - The Pascal File Variable (PFV) passed by reference.
1 !                         The structure of the PFV is defined in PASPFV.REQ.
1 !
1 !       VALUE           - The H_floating value to write, by immediate value.
1 !                         Note that this takes up four argument list positions.
1 !
1 !       TOTAL_WIDTH     - Total field width.
1 !
1 !       FRAC_DIGITS     - Number of digits in fraction.
1 !
1 !       ERROR           - Optional.  Address to unwind to if an error occurs.
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 !       If the file is the standard file OUTPUT, it is implicitly opened.
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       LINTOOLON - line too long
1 !       NEGWIDDIG - negative Width or Digits specification is not allowed
1 !
1 !--
1
```

```
;  139          0200  2       BEGIN
;  140          0201  2
;  141          0202  2       BUILTIN
;  142          0203  2           ACTUALCOUNT;
;  143          0204  2
;  144          0205  2       LOCAL
;  145          0206  2           FCB: REF $PAS$FCB_CONTROL_BLOCK,                ! File control block
;  146          0207  2           FIELD_WIDTH: SIGNED,                            ! Minimum/actual width
;  147          0208  2           REMAINING_WIDTH,                                ! Maximum width
;  148          0209  2           PFV_ADDR: VOLATILE,                             ! Enable argument
;  149          0210  2           UNWIND_ACT: VOLATILE,                           ! Enable argument
;  150          0211  2           ERROR_ADDR: VOLATILE;                           ! Enable argument
;  151          0212  2
;  152          0213  2       ENABLE
;  153          0214  2           PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
;  154          0215  2
;  155          0216  2       !+
;  156          0217  2       ! Get ERROR parameter, if present.
;  157          0218  2       !-
;  158          0219  2
;  159          0220  2       IF ACTUALCOUNT () GEQU 8
;  160          0221  2       THEN
;  161          0222  2           ERROR_ADDR = .ERROR;             ! Set unwind address
;  162          0223  2
;  163          0224  2       PFV_ADDR = PFV [PFV$R_PFV];          ! Set PFV address
;  164          0225  2
;  165          0226  2       !+
;  166          0227  2       ! Validate PFV and get PFV.
;  167          0228  2       !-
;  168          0229  2
;  169          0230  2       PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
;  170          0231  2
;  171          0232  2       !+
;  172          0233  2       ! Set unwind action to unlock file.
;  173          0234  2       !-
;  174          0235  2
;  175          0236  2       UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
;  176          0237  2
;  177          0238  2       !+
;  178          0239  2       ! Do common initialization.
;  179          0240  2       !-
;  180          0241  2
;  181          0242  2       PAS$$INIT_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
;  182          0243  2
;  183          0244  2       !+
;  184          0245  2       ! Get minimum and maximum widths.  Check for invalid width or digits.
;  185          0246  2       !-
;  186          0247  2
;  187          0248  2       FIELD_WIDTH = .TOTAL_WIDTH;
;  188          0249  3       IF (.FIELD_WIDTH LSS 0) OR (.FRAC_DIGITS LSS 0)
;  189          0250  2       THEN
;  190          0251  2           $PAS$IO_ERROR (PAS$_NEGWIDDIG,0);
;  191          0252  2       REMAINING_WIDTH = .FCB [FCB$A_RECORD_END] - .FCB [FCB$A_RECORD_CUR];
;  192          0253  2
;  193          0254  2       !+
;  194          0255  2       ! Do the conversion.  If it fails, signal an error.
;  195          0256  2       !-
;  196          0257  2     IF NOT PAS$CVT_H_T (VALUE_0,                   ! Value to convert
;  197          0258  2                 .FCB [FCB$A_RECORD_CUR],            ! Destination
;  198          0259  2                 FIELD_WIDTH,                        ! Minimum/actual width
;  199          0260  2                 .REMAINING_WIDTH,                   ! Maximum width
;  200          0261  2                 .FRAC_DIGITS)                       ! Fraction digits
;  201          0262  2     THEN
;  202          0263  2         $PAS$IO_ERROR (PAS$_LINTOOLON,1,(.FIELD_WIDTH-.FRAC_DIGITS));
;  203          0264  2
;  204          0265  2     !+
;  205          0266  2     ! Advance buffer pointer.
;  206          0267  2     !-
;  207          0268  2
;  208          0269  2     FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_CUR] + .FIELD_WIDTH;
;  209          0270  2
;  210          0271  2     !+
;  211          0272  2     ! Call WRITE epilogue routine to move the last character written to the
;  212          0273  2     ! user's buffer and to unlock the file variable.
;  213          0274  2     !-
;  214          0275  2
;  215          0276  2     PAS$$END_WRITE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
;  216          0277  2
;  217          0278  2     RETURN;
;  218          0279  2
;  219          0280  2
;  220          0281  1     END;                                        ! End of routine PAS$WRITE_REALF_H

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

                                                                    .EXTRN  PAS$WRITE_REALF_H
                                                                    .EXTRN  PAS$WRITEV_REALF_H
                                                                    .EXTRN  PAS$$IO_HANDLER
                                                                    .EXTRN  PAS$$VALIDATE_PFV
                                                                    .EXTRN  PAS$$INIT_WRITE
                                                                    .EXTRN  PAS$$SIGNAL, PAS$K_NEGWIDDIG
                                                                    .EXTRN  PAS$CVT_H_T, PAS$K_LINTOOLON
                                                                    .EXTRN  PAS$$END_WRITE

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

                                         01FC 00000                 .ENTRY  PAS$WRITE_REALF_H, Save R2,R3,R4,R5,R6,R7,- ; 0144
                                                                            R8                                          ;
                        58 00000000G  00   9E 00002                 MOVAB   PAS$$SIGNAL, R8                             ;
                        5E            10   C2 00009                 SUBL2   #16, SP                                     ;
                                04    AE   7C 0000C                 CLRQ    ERROR_ADDR                                  ; 0200
                                0C    AE   D4 0000F                 CLRL    PFV_ADDR                                    ;
                        6D      0070  CF   DE 00012                 MOVAL   5$, (FP)                                    ;
                        08            6C   91 00017                 CMPB    (AP), #8                                    ; 0220
                                      05   1F 0001A                 BLSSU   1$                                          ;
                   04   AE        20  AC   D0 0001C                 MOVL    ERROR, ERROR_ADDR                           ; 0222
                        56        04  AC   D0 00021 1$:             MOVL    PFV, R6                                     ; 0224
                   0C   AE            56   D0 00025                 MOVL    R6, PFV_ADDR                                ;
                           00000000G  00   16 00029                 JSB     PAS$$VALIDATE_PFV                           ; 0230
                   08   AE            01   D0 0002F                 MOVL    #1, UNWIND_ACT                              ; 0236
                           00000000G  00   16 00033                 JSB     PAS$$INIT_WRITE                             ; 0242
```

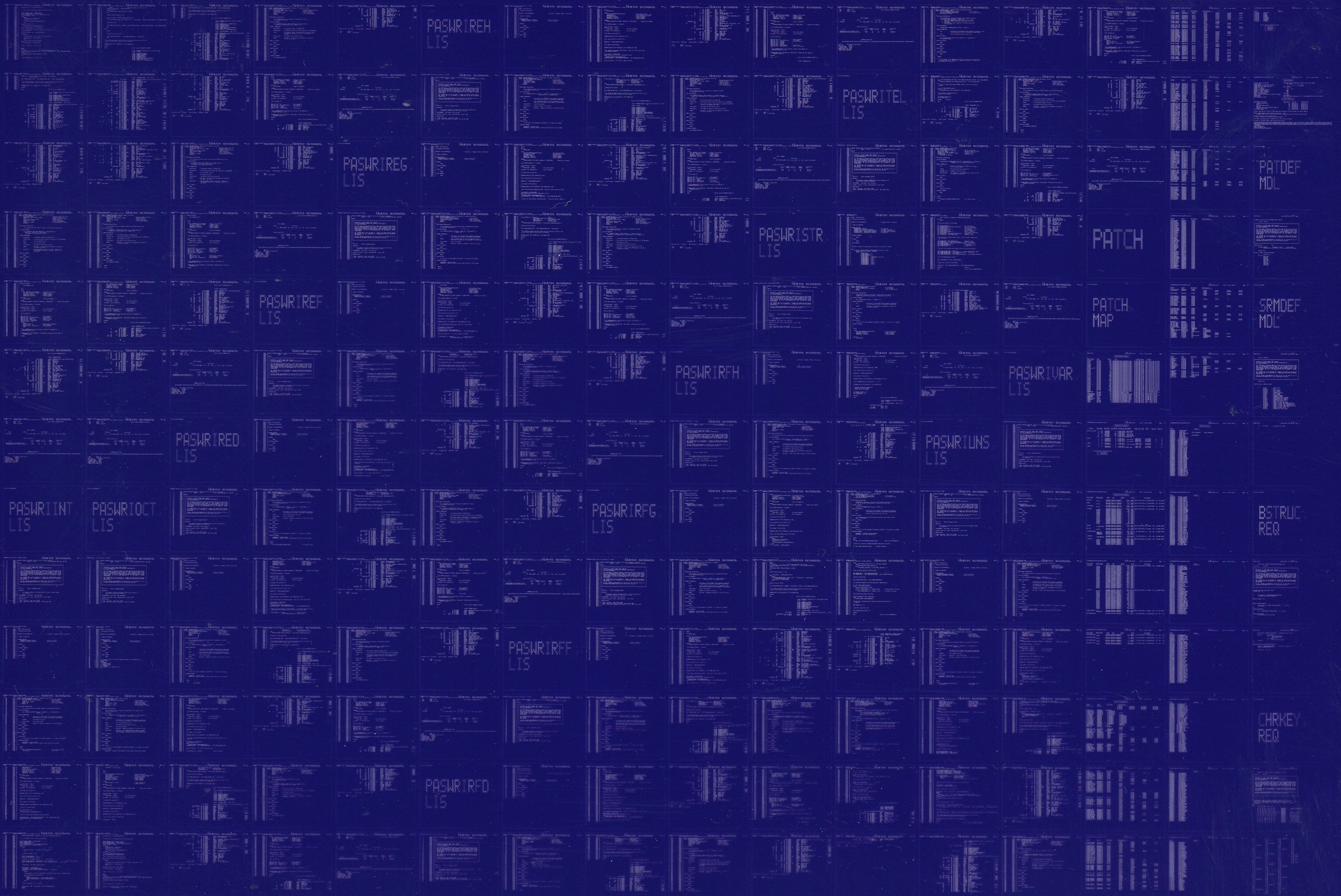
```
                                       6E       18     AC D0 00039         MOVL    TOTAL_WIDTH, FIELD_WIDTH                     ; 0248
                                                       05 19 0003D         BLSS    2$                                           ; 0249
                                                1C     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$$SIGNAL                              ;
                                                          04 0004D         RET                                                  ;
                         50       F0   A7       EC     A7 C3 0004E 3$:     SUBL3   -20(FCB), -16(FCB), REMAINING_WIDTH          ; 0252
                                                1C     AC DD 00054         PUSHL   FRAC_DIGITS                                  ; 0262
                                                       50 DD 00057         PUSHL   REMAINING_WIDTH                              ; 0261
                                                08     AE 9F 00059         PUSHAB  FIELD_WIDTH                                  ; 0258
                                                EC     A7 DD 0005C         PUSHL   -20(FCB)                                     ; 0259
                                                08     AC 9F 0005F         PUSHAB  VALUE_0                                      ; 0258
                            00000000G  00              05 FB 00062         CALLS   #5, PAS$CVT_H_T                              ;
                                       0F              50 E8 00069         BLBS    R0, 4$                                       ;
                         7E            6E       1C     AC C3 0006C         SUBL3   FRAC_DIGITS, FIELD_WIDTH, -(SP)              ; 0264
                                                       01 DD 00071         PUSHL   #1                                           ;
                                       7E       00G    8F 9A 00073         MOVZBL  #PAS$K_LINTuOLON, -(SP)                      ;
                                       68              03 FB 00077         CALLS   #3, PAS$$SIGNAL                              ;
                                                          04 0007A         RET                                                  ;
                                  EC   A7              6E C0 0007B 4$:     ADDL2   FIELD_WIDTH, -20(FCB)                        ; 0270
                                          00000000G    00 16 0007F         JSB     PAS$$END_WRITE                               ; 0277
                                                          04 00085         RET                                                  ; 0281
                                                       0000 00086 5$:     .WORD   Save nothing                                 ; 0200
                                       50       08     AC D0 00088         MOVL    8(AP), R0                                    ;
                                       50       04     A0 D0 0008C         MOVL    4(R0), R0                                    ;
                                                F4     A0 9F 00090         PUSHAB  ERROR_ADDR                                   ;
                                                F8     A0 9F 00093         PUSHAB  UNWIND_A                                     ;
                                                FC     A0 9F 00096         PUSHAB  PFV_ADD                                      ;
                                                       03 DD 00099         PUSHL   #3                                           ;
                                                       5E DD 0009B         PUSHL   SP                                           ;
                                       7E       04     AC 7D 0009D         MOVQ    4(AP), -(SP)                                 ;
                            00000000G  00              03 FB 000A1         CALLS   #3, PAS$$IO_HANDLER                          ;
                                                          04 000A8         RET                                                  ;

; Routine Size:  169 bytes,    Routine Base:  _PAS$CODE + 0000


;  221         0282  1
;  222         0283  1 !<BLF/PAGE>
;  224          0284    1 %SBTTL 'PAS$WRITEV_REALF_H - Write H_floating in F format to string'
;  225          0285    1 GLOBAL ROUTINE PAS$WRITEV_REALF_H (
;  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         VALUE0,VALUE1,VALUE2,VALUE3,                    ! Value to write
;  229          0289    1         TOTAL_WIDTH: SIGNED,                            ! Total field width
;  230          0290    1         FRAC_DIGITS: SIGNED,                            ! Digits in fraction
;  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 H_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$WRITEV_REALF_H (MAX_LENGTH.rw.v, STRING_LINE.wvt.r,
;  243          0303    1 !           VALUE.rh.v, TOTAL_WIDTH.rl.v, FRAC_DIGITS.rl.v [, ERROR.j.r])
;  244          0304    1 !
;  245          0305    1 ! FORMAL PARAMETERS:
;  246          0306    1 !
;  247          0307    1 !       MAX_LENGTH       - The maximum length of STRING_LINE.
;  248          0308    1 !
;  249          0309    1 !       STRING_LINE      - A varying string to which the output will be appended.
;  250          0310    1 !
;  251          0311    1 !       VALUE            - The value to write.  Note that the H_floating value
;  252          0312    1 !                          is passed by immediate value in four argument list
;  253          0313    1 !                          positions.
;  254          0314    1 !
;  255          0315    1 !       TOTAL_WIDTH      - The width of the field to write.
;  256          0316    1 !
;  257          0317    1 !       FRAC_DIGITS      - The number of fraction digits.
;  258          0318    1 !
;  259          0319    1 !       ERROR            - Optional.  If specified, the address to unwind to
;  260          0320    1 !                          in case of an error.
;  261          0321    1 !
;  262          0322    1 ! IMPLICIT INPUTS:
;  263          0323    1 !
;  264          0324    1 !       NONE
;  265          0325    1 !
;  266          0326    1 ! IMPLICIT OUTPUTS:
;  267          0327    1 !
;  268          0328    1 !       NONE
;  269          0329    1 !
;  270          0330    1 ! ROUTINE VALUE:
;  271          0331    1 !
;  272          0332    1 !       NONE
;  273          0333    1 !
;  274          0334    1 ! SIDE EFFECTS:
;  275          0335    1 !
;  276          0336    1 !       NONE
;  277          0337    1 !
;  278          0338    1 ! SIGNALLED ERRORS:
;  279          0339    1 !
;  280          0340    1 !       See PAS$WRITE_REALF_H
```

```
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         PFV: $PAS$PFV_FILE_VARIABLE,     ! Pascal File Variable
2         ARG_LIST: VECTOR [8, LONG],      ! Argument list
2         PFV_ADDR: VOLATILE,              ! Enable argument
2         UNWIND_ACT: VOLATILE,            ! Enable argument
2         ERROR_ADDR: VOLATILE;            ! Enable argument
2
2     BUILTIN
2         ACTUALCOUNT;                     ! Count of arguments
2
2     ENABLE
2         PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);     ! Enable error handler
2
2     !+
2     ! Get ERROR parameter, if present.
2     !-
2
2     IF ACTUALCOUNT () GEQU 9
2     THEN
2         ERROR_ADDR = .ERROR;             ! Set unwind address
2
2     PFV_ADDR = PFV [PFV$R_PFV];          ! Set PFV address
2
2     !+
2     ! Set up ARG_LIST.
2     !-
2
2     ARG_LIST [0] = 7;                    ! Seven arguments
2     ARG_LIST [1] = PFV [PFV$R_PFV];      ! PFV address
2     ARG_LIST [2] = .VALUE0;              ! Value to write
2     ARG_LIST [3] = .VALUE1;              !
2     ARG_LIST [4] = .VALUE2;              !
2     ARG_LIST [5] = .VALUE3;              !
2     ARG_LIST [6] = .TOTAL_WIDTH;         ! Field width
2     ARG_LIST [7] = .FRAC_DIGITS;         ! Digits in fraction
2
2     !+
2     ! Call PAS$$DO_WRITEV to do the work, giving it the address of
2     ! PAS$WRITE_REALF_H to call.
2     !-
2
2     PAS$$DO_WRITEV (PFV [PFV$R_PFV], .MAX_LENGTH, STRING_LINE [0], ARG_LIST,
2         PAS$WRITE_REALF_H);
2
2     RETURN;
2
1     END;                                         ! End of routine PAS$WRITEV_REALF_H


                                                    .EXTRN  PAS$$DO_WRITEV
```

```
                                    007C 00000         .ENTRY  PA,$WRITEV_REALF_H, Save R2,R3,R4,R5,R6     ; 0285
              5E               38   C2 00002           SUBL2   #56, SP                                     :
                               7E   D4 00005           CLRL    ERROR_ADDR                                  ; 0344
                        04     AE   7C 00007           CLRQ    UNWIND_ACT                                  :
              6D      0043     CF   DE 0000A           MOVAL   2$, (FP)                                    :
              09               6C   91 0000F           CMPB    (AP), #9                                    ; 0363
                               04   1F 00012           BLSSU   1$                                          :
              6E        24     AC   D0 00014           MOVL    ERROR, ERROR_ADDR                           ; 0365
        08    AE        2C     AE   9E 00018 1$:       MOVAB   PFV, PFV_ADDR                               ; 0367
        0C    AE               07   D0 0001D           MOVL    #7, ARG_LIST                                ; 0373
        10    AE        2C     AE   9E 00021           MOVAB   PFV, ARG_LIST+4                             ; 0374
        1'    AE        0C     AC   7D 00026           MOVQ    VALUE0, ARG_LIST+8                          ; 0375
        1C    AE        14     AC   7D 0002B           MOVQ    VALUE2, ARG_LIST+16                         ; 0377
        24    AE        1C     AC   7D 00030           MOVQ    TOTAL_WIDTH, ARG_LIST+24                    ; 0379
              55      FF1E     CF   9E 00035           MOVAB   PAS$WRITE_REALF_H, R5                       ; 0387
              54        0C     AE   9E 0003A           MOVAB   ARG_LIST, R4                                :
              56        2C     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$$DO_WRITEV                              :
                                    04 00050           RET                                                 ; 0392
                                  0000 00051 2$:       .WORD   Save nothing                                ; 0344
              50        08     AC   D0 00053           MOVL    8(AP), R0                                   :
              50        04     A0   D0 00057           MOVL    4(R0), R0                                   :
                        C4     A0   9F 0005B           PUSHAB  ERROR_ADDR                                  :
                        C8     A0   9F 0005E           PUSHAB  UNWIND_ACT                                  :
                        CC     A0   9F 00061           PUSHAB  PFV_ADDR                                    :
                               03   DD 00064           PUSHL   #3                                          :
                               5E   DD 00066           PUSHL   SP                                          :
              7E        04     AC   7D 00068           MOVQ    4(AP), -(SP)                                :
   00000000G  00               03   FB 0006C           CALLS   #3, PAS$$IO_HANDLER                         :
                                    04 00073           RET                                                 :

; Routine Size:  116 bytes,    Routine Base:  _PAS$CODE + 00A9


;  333          0393  1
;  334          0394  1 !<BLF/PAGE>
;  336          0395  1 END                                                   ! End of module PAS$WRITE_REALF_H
;  337          0396  1
;  338          0397  0 ELUDOM



;                                          PSECT SUMMARY
;
;       Name                          Bytes                       Attributes
;
;  _PAS$CODE                               285  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$:PASWRIRFH/OBJ=OBJ$:PASWRIRFH MSRC$:PASWRIRFH/UPDATE=(ENH$:PASWRIRFH
;       )

; Size:         285 code + 0 data bytes
; Run Time:        00:07.3
; Elapsed Time:    00:19.0
; Lines/CPU Min:     3258
; Lexemes/CPU-Min: 13756
; Memory Used:  83 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