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

FILEID**PASFINDK

```
PPPPPPPP      AAAAAA      SSSSSSSS  FFFFFFFFFF    IIIIII    NN      NN  DDDDDDDD    KK      KK
PPPPPPPP      AAAAAA      SSSSSSSS  FFFFFFFFFF    IIIIII    NN      NN  DDDDDDDD    KK      KK
PP      PP   AA      AA  SS        FF              II      NN      NN  DD      DD  KK      KK
PP      PP   AA      AA  SS        FF              II      NN      NN  DD      DD  KK      KK
PP      PP   AA      AA  SS        FF              II      NNNN    NN  DD      DD  KK    KK
PP      PP   AA      AA  SS        FF              II      NNNN    NN  DD      DD  KK    KK
PPPPPPPP     AA      AA    SSSSSS  FFFFFFFF        II      NN  NN  NN  DD      DD  KKKKKK
PPPPPPPP     AA      AA    SSSSSS  FFFFFFFF        II      NN  NN  NN  DD      DD  KKKKKK
PP           AAAAAAAAAA        SS  FF              II      NN    NNNN  DD      DD  KK    KK
PP           AAAAAAAAAA        SS  FF              II      NN    NNNN  DD      DD  KK    KK
PP           AA      AA        SS  FF              II      NN      NN  DD      DD  KK      KK      ....
PP           AA      AA        SS  FF              II      NN      NN  DD      DD  KK      KK      ....
PP           AA      AA  SSSSSSSS  FF            IIIIII    NN      NN  DDDDDDDD    KK      KK      ....
PP           AA      AA  SSSSSSSS  FF            IIIIII    NN      NN  DDDDDDDD    KK      KK      ....

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$FINDK ( %TITLE 'FINDK procedure'
;     2          0002  0                 IDENT = '1-003'         ! File: PASFINDK.B32 Edit: SBL1003
;     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 PAS$FINDK, which implements the
;    36          0036  1 !       VAX-11 Pascal FINDK procedure.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  User mode - AST reentrant
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: Steven B. Lionel, CREATION DATE: 24-February-1982
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.  SBL 24-February-1982
;    45          0045  1 ! 1-002 - Store length word for FILE OF VARYING.  FT3 QAR 006  SBL 31-Aug-1982
;    46          0046  1 ! 1-003 - Turn off RAB$V_NLK.  SPR 11-55690 SBL 8-Apr-1983
;    47          0047  1 !--
;    48          0048  1
```

```
;    50          0049  1 %SBTTL 'Declarations'
;    51          0050  1 !
;    52          0051  1 ! PROLOGUE DEFINITIONS:
;    53          0052  1 !
;    54          0053  1
;    55          0054  1 REQUIRE 'RTLIN:PASPROLOG';                    ! Externals, linkages, PSECTs, structures
;    56          0118  1
;    57          0119  1 !
;    58          0120  1 ! TABLE OF CONTENTS:
;    59          0121  1 !
;    60          0122  1
;    61          0123  1 FORWARD ROUTINE
;    62          0124  1     PAS$FINDK: NOVALUE;                      ! Do a FINDK
;    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$FINDK - FINDK procedure'
;    83          0144  1 GLOBAL ROUTINE PAS$FINDK (
;    84          0145  1         PFV: REF $PAS$PFV_FILE_VARIABLE,                ! File variable
;    85          0146  1         KEY_NUMBER,                                     ! Key number
;    86          0147  1         KEY_VALUE: REF BLOCK [, BYTE],                  ! Key value
;    87          0148  1         MATCH_TYPE: BYTE,                               ! Match type
;    88          0149  1         ERROR                                           ! Unwind address if error
;    89          0150  1     ): NOVALUE =
;    90          0151  1
;    91          0152  1 !++
;    92          0153  1 ! FUNCTIONAL DESCRIPTION:
;    93          0154  1 !
;    94          0155  1 !       PAS$FINDK implements the VAX-11 Pascal FINDK procedure.  It
;    95          0156  1 !       finds a record in an indexed file by key and does a GET.
;    96          0157  1 !
;    97          0158  1 ! CALLING SEQUENCE:
;    98          0159  1 !
;    99          0160  1 !       CALL PAS$FINDK (PFV.mr.r, KEY_NUMBER.rl.v,
;   100          0161  1 !               KEY_VALUE.rx.ds, MATCH_TYPE.rbu.v [, ERROR.j.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 !       KEY_NUMBER      - The number of the key to be searched.
;   108          0169  1 !
;   109          0170  1 !       KEY_VALUE       - The value of the key to look for, passed by
;   110          0171  1 !                         descriptor.  Valid datatypes are BU, WU, L
;   111          0172  1 !                         LU, T and Z. (Z is equivalent to T).
;   112          0173  1 !
;   113          0174  1 !       MATCH_TYPE      - Denotes the match type for the search.  The values
;   114          0175  1 !                         are:
;   115          0176  1 !                               0 - Match on equal only
;   116          0177  1 !                               1 - Match on greater than or equal to
;   117          0178  1 !                               2 - Match on greater than only
;   118          0179  1 !
;   119          0180  1 !       ERROR           - Optional.  If specified, the address to unwind to
;   120          0181  1 !                         if an error occurs.
;   121          0182  1 !
;   122          0183  1 ! IMPLICIT INPUTS:
;   123          0184  1 !
;   124          0185  1 !       NONE
;   125          0186  1 !
;   126          0187  1 ! IMPLICIT OUTPUTS:
;   127          0188  1 !
;   128          0189  1 !       NONE
;   129          0190  1 !
;   130          0191  1 ! ROUTINE VALUE:
;   131          0192  1 !
;   132          0193  1 !       NONE
;   133          0194  1 !
;   134          0195  1 ! SIDE EFFECTS:
;   135          0196  1 !
;   136          0197  1 !       Places file in Inspection mode
;   137          0198  1 !
;   138          0199  1 ! SIGNALLED ERRORS:
```

```
;  139          0200  1 !
;  140          0201  1 !       FILNOTOPE - File not open
;  141          0202  1 !       FILNOTKEY - File not opened for keyed access
;  142          0203  1 !       KEYNOTDEF - Key "n" is not defined for this file
;  143          0204  1 !       KEYVALINC - Key value is incompatible
;  144          0205  1 !       FAIGETLOC - Failed to get locked record
;  145          0206  1 !       ERRDURFIN - Error during FIND or FINDK
;  146          0207  1 !
;  147          0208  1 !
;  148          0209  1 !--
;  149          0210  1
;  150          0211  2     BEGIN
;  151          0212  2
;  152          0213  2     LOCAL
;  153          0214  2         FCB: REF $PAS$FCB_CONTROL_BLOCK,            ! File control block
;  154          0215  2         PFV_ADDR: VOLATILE,                         ! Enable argument
;  155          0216  2         UNWIND_ACT: VOLATILE,                       ! Enable argument
;  156          0217  2         ERROR_ADDR: VOLATILE,                       ! Enable argument
;  157          0218  2         KEYTYPES_VECTOR: REF VECTOR [, WORD],       ! Vector of keytype masks and lengths
;  158          0219  2         KEYTYPES_MASK: BITVECTOR [16],              ! Mask of allowable key types
;  159          0220  2         KEY_SIZE: WORD,                             ! Key size in bytes
;  160          0221  2         STATUS,                                     ! Status from $GET
;  161          0222  2         LOCAL_KEY;                                  ! Local key value
;  162          0223  2
;  163          0224  2     BIND
;  164          0225  2         RAB = FCB: REF BLOCK [, BYTE];              ! RMS RAB
;  165          0226  2
;  166          0227  2     BUILTIN
;  167          0228  2         ACTUALCOUNT;
;  168          0229  2
;  169          0230  2     LITERAL
;  170          0231  2         K_EQL = 0,                                  ! Match on equal
;  171          0232  2         K_GEQ = 1,                                  ! Match on greater or equal
;  172          0233  2         K_GTR = 2;                                  ! Match on greater
;  173          0234  2
;  174          0235  2     ENABLE
;  175          0236  2         PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);
;  176          0237  2
;  177          0238  2     IF ACTUALCOUNT () GEQU 5
;  178          0239  2     THEN
;  179          0240  2         ERROR_ADDR = .ERROR;                        ! Set unwind address
;  180          0241  2
;  181          0242  2     !+
;  182          0243  2     ! Set PFV address enable argument.
;  183          0244  2     !-
;  184          0245  2
;  185          0246  2     PFV_ADDR = PFV [PFV$R_PFV];
;  186          0247  2
;  187          0248  2     !+
;  188          0249  2     ! Validate and lock PFV
;  189          0250  2     !-
;  190          0251  2
;  191          0252  2     PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
;  192          0253  2
;  193          0254  2     !+
;  194          0255  2     ! Set unwind action to unlock file.
;  195          0256  2     !-
```

```
;  196          0257  2
;  197          0258  2     UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
;  198          0259  2
;  199          0260  2     !+
;  200          0261  2     ! Open file if it should be implicitly opened.
;  201          0262  2     !-
;  202          0263  2
;  203          0264  2     IF NOT .PFV [PFV$V_VALID]
;  204          0265  2     THEN
;  205          0266  2         PAS$$OPEN_IMPLICIT (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
;  206          0267  2
;  207          0268  2     !+
;  208          0269  2     ! Verify that the file is open.
;  209          0270  2     !-
;  210          0271  2
;  211          0272  2     IF NOT .PFV [PFV$V_OPEN]
;  212          0273  2     THEN
;  213          0274  2         $PAS$IO_ERROR (PAS$_FILNOTOPE,0);
;  214          0275  2
;  215          0276  2     !+
;  216          0277  2     ! Verify that the file is opened for keyed access
;  217          0278  2     !-
;  218          0279  2
;  219          0280  2     IF NOT .FCB [FCB$V_KEYED]
;  220          0281  2     THEN
;  221          0282  2         $PAS$IO_ERROR (PAS$_FILNOTKEY,0);
;  222          0283  2
;  223          0284  2     !+
;  224          0285  2     ! Verify that the key number is valid for the file and
;  225          0286  2     ! set the key number.  Set access mode to keyed.
;  226          0287  2     !-
;  227          0288  2
;  228          0289  2     IF .KEY_NUMBER GEQU .FCB [FCB$L_NKEYS]
;  229          0290  2     THEN
;  230          0291  2         $PAS$IO_ERROR (PAS$_KEYNOTDEF,1,.KEY_NUMBER);
;  231          0292  2     RAB [RAB$B_KRF] = .KEY_NUMBER;              ! Set key number
;  232          0293  2     RAB [RAB$B_RAC] = RAB$C_KEY;                ! Keyed access
;  233          0294  2
;  234          0295  2     !+
;  235          0296  2     ! Get mask of allowable datatypes for this key.  KEYTYPES_VECTOR is
;  236          0297  2     ! a vector of longwords, one for each key in the file.  The first word of
;  237          0298  2     ! each longword is a bit mask, with each bit corresponding to a datatype
;  238          0299  2     ! code (0-15).  If a bit is set, that key datatype may be used to access
;  239          0300  2     ! the key.  The second word is the length of the key in bytes.
;  240          0301  2     !-
;  241          0302  2
;  242          0303  2     KEYTYPES_VECTOR = .FCB [FCB$A_KEY_TYPES];
;  243          0304  2     KEYTYPES_MASK = .KEYTYPES_VECTOR [.KEY_NUMBER*2];
;  244          0305  2
;  245          0306  2     !+
;  246          0307  2     ! Check to see if caller key type is allowed for this key.
;  247          0308  2     !-
;  248          0309  2
;  249          0310  2     IF .KEY_VALUE [DSC$B_DTYPE] GTRU 15 ! No datatype greater than 15 is allowed
;  250          0311  2     THEN
;  251          0312  2         $PAS$IO_ERROR (PAS$_KEYVALINC,1,.KEY_NUMBER);
;  252          0313  2     IF NOT .KEYTYPES_MASK [.KEY_VALUE [DSC$B_DTYPE]]
```

```
;  253          0314  2     THEN
;  254          0315  2         $PAS$IO_ERROR (PAS$_KEYVALINC,1,.KEY_NUMBER);
;  255          0316  2
;  256          0317  2     !+
;  257          0318  2     ! Get key value.
;  258          0319  2     !-
;  259          0320  2
;  260          0321  2     IF  .KEY_VALUE [DSC$B_DTYPE] EQL DSC$K_DTYPE_T OR
;  261          0322  2         .KEY_VALUE [DSC$B_DTYPE] EQL DSC$K_DTYPE_Z
;  262          0323  2     THEN
;  263          0324  3         BEGIN
;  264          0325  3         RAB [RAB$L_KBF] = .KEY_VALUE [DSC$A_POINTER];   ! Key address
;  265          0326  3         IF .KEY_VALUE [DSC$W_LENGTH] GTRU 255
;  266          0327  3         THEN
;  267          0328  3             $PAS$IO_ERROR (PAS$_KEYVALINC,1,.KEY_NUMBER);
;  268          0329  3         RAB [RAB$B_KSZ] = .KEY_VALUE [DSC$W_LENGTH];    ! String size
;  269          0330  3         END
;  270          0331  2     ELSE
;  271          0332  3         BEGIN
;  272          0333  3         !+
;  273          0334  3         ! Key value is numeric.  Expand it to a longword and store in
;  274          0335  3         ! LOCAL_KEY.
;  275          0336  3         !-
;  276          0337  3
;  277          0338  3         LOCAL
;  278          0339  3             KEYVAL: REF BLOCK [, BYTE];
;  279          0340  3
;  280          0341  3         KEYVAL = .KEY_VALUE [DSC$A_POINTER];    ! Get key value address
;  281          0342  3
;  282          0343  3         CASE .KEY_VALUE [DSC$B_DTYPE] FROM DSC$K_DTYPE_BU TO DSC$K_DTYPE_L OF
;  283          0344  3             SET
;  284          0345  3
;  285          0346  3             [DSC$K_DTYPE_BU]:   LOCAL_KEY = .KEYVAL [0,0,8,0];
;  286          0347  3             [DSC$K_DTYPE_WU]:   LOCAL_KEY = .KEYVAL [0,0,16,0];
;  287          0348  3             [DSC$K_DTYPE_L, DSC$K_DTYPE_LU]:
;  288          0349  3                                 LOCAL_KEY = .KEYVAL [0,0,32,0];
;  289          0350  3             [INRANGE, OUTRANGE]:
;  290          0351  3                 $PAS$IO_ERROR (PAS$_INVARGPAS,0);
;  291          0352  3
;  292          0353  3             TES;
;  293          0354  3
;  294          0355  3         !+
;  295          0356  3         ! Check that key value is in proper range for file's key.
;  296          0357  3         !-
;  297          0358  3
;  298          0359  3         KEY_SIZE = .KEYTYPES_VECTOR [(2*.KEY_NUMBER)+1];
;  299          0360  3         IF .KEY_SIZE EQL 1
;  300          0361  3         THEN
;  301          0362  4             BEGIN
;  302          0363  4             IF .LOCAL_KEY GTRU 255
;  303          0364  4             THEN
;  304          0365  4                 $PAS$IO_ERROR (PAS$_KEYVALINC,1,.KEY_NUMBER);
;  305          0366  4             END
;  306          0367  3         ELSE IF .KEY_SIZE EQL 2
;  307          0368  3         THEN
;  308          0369  4             BEGIN
;  309          0370  4             IF .LOCAL_KEY GTRU 65535
```

```
;  310     0371  4              THEN
;  311     0372  4                  $PAS$IO_ERROR (PAS$_KEYVALINC,1,.KEY_NUMBER);
;  312     0373  4              END
;  313     0374  3          ELSE IF .KEY_SIZE NEQ 4
;  314     0375  3          THEN
;  315     0376  3              $PAS$IO_ERROR (PAS$_INVARGPAS,0);
;  316     0377  3
;  317     0378  3          !+
;  318     0379  3          ! Set key buffer address and size.
;  319     0380  3          !-
;  320     0381  3
;  321     0382  3          RAB [RAB$L_KBF] = LOCAL_KEY;
;  322     0383  3          RAB [RAB$B_KSZ] = .KEY_SIZE;
;  323     0384  2          END;
;  324     0385  2
;  325     0386  2      !+
;  326     0387  2      ! Set match type.
;  327     0388  2      !-
;  328     0389  2
;  329     0390  2      RAB [RAB$V_KGE] = 0;
;  330     0391  2      RAB [RAB$V_KGT] = 0;
;  331     0392  2
;  332     0393  2      CASE .MATCH_TYPE FROM K_EQL TO K_GTR OF
;  333     0394  2          SET
;  334     0395  2
;  335     0396  2          [K_EQL]: ;                          ! Set equal
;  336     0397  2          [K_GEQ]: RAB [RAB$V_KGE] = 1;       ! Set greater or equal
;  337     0398  2          [K_GTR]: RAB [RAB$V_KGT] = 1;       ! Set greater than
;  338     0399  2          [OUTRANGE]:
;  339     0400  2              $PAS$IO_ERROR (PAS$_INVARGPAS,0);
;  340     0401  2
;  341     0402  2          TES;
;  342     0403  2
;  343     0404  2      PFV [PFV$V_DFB] = 0;                    ! Undefine file buffer
;  344     0405  2      PFV [PFV$V_EOF_DEFINED] = 0;            ! EOF(f) not defined
;  345     0406  2      FCB [FCB$V_EOF] = 0;                    ! Not at EOF
;  346     0407  2      RAB [RAB$V_NLK] = 0;                    ! Turn off NLK bit
;  347     0408  2
;  348     0409  2      !+
;  349     0410  2      ! Get the record.
;  350     0411  2      !-
;  351     0412  2
;  352     0413  2      STATUS = $PAS$RMS_OP ($GET (RAB=.RAB));
;  353     0414  2
;  354     0415  2      !+
;  355     0416  2      ! If varying, put component length in first word of user buffer.
;  356     0417  2      ! Do this before we check for errors!
;  357     0418  2      !-
;  358     0419  2
;  359     0420  2      IF .FCB [FCB$V_VARYING]
;  360     0421  2      THEN
;  361     0422  2          (.PFV [PFV$A_BUFFER])<0,16,0> = .RAB [RAB$W_RSZ];
;  362     0423  2
;  363     0424  2      IF NOT .STATUS
;  364     0425  2      THEN
;  365     0426  3          BEGIN
;  366     0427  3          !+
```

```
;  367          0428  3          ! If the status is "record not found", simply leave DFB clear.
;  368          0429  3          ! Otherwise, signal the appropriate error.
;  369          0430  3          !-
;  370          0431  3
;  371          0432  3          IF .STATUS NEQ RMS$_RNF
;  372          0433  3          THEN
;  373          0434  3              IF .STATUS EQL RMS$_RLK     ! Record locked?
;  374          0435  3              THEN
;  375          0436  4                  $PAS$IO_ERROR (PAS$_FAIGETLOC)  ! Failed to get loc ed record
;  376          0437  3              ELSE
;  377          0438  3                  $PAS$IO_ERROR (PAS$_ERRDURFIN); ! Error during FIND or FINDK
;  378          0439  3              END
;  379          0440  2      ELSE
;  380          0441  2          PFV [PFV$V_DFB] = 1;
;  381          0442  2
;  382          0443  2      !+
;  383          0444  2      ! Set Inspection mode
;  384          0445  2      ! Indicate successful completion
;  385          0446  2      ! Unlock the file
;  386          0447  2      !-
;  387          0448  2
;  388          0449  2      FCB [FCB$V_INSPECTION] = 1;
;  389          0450  2      FCB [FCB$V_GENERATION] = 0;
;  390          0451  2      FCB [FCB$L_STATUS] = 0;
;  391          0452  2      PFV [PFV$V_LOCK] = 0;
;  392          0453  2
;  393          0454  2      RETURN;
;  394          0455  2
;  395          0456  1      END;                                   ! End of routine PAS$FINDK


                                                .TITLE  PAS$FINDK FINDK procedure
                                                .IDENT  \1-003\

                                                .EXTRN  PAS$FINDK, PAS$$IO_HANDLER
                                                .EXTRN  PAS$$VALIDATE_PFV
                                                .EXTRN  PAS$$OPEN_IMPLICIT
                                                .EXTRN  PAS$$SIGNAL, PAS$K_FILNOTOPE
                                                .EXTRN  PAS$K_FILNOTKEY
                                                .EXTRN  PAS$K_KEYNOTDEF
                                                .EXTRN  PAS$K_KEYVALINC
                                                .EXTRN  PAS$K_INVARGPAS
                                                .EXTRN  SYS$GET, PAS$K_FAIGETLOC
                                                .EXTRN  PAS$K_ERRDURFIN

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

                                  0FFC 00000    .ENTRY  PAS$FINDK, Save R2,R3,R4,R5,R6,R7,R8,R9,-    ; 0144
                                                        R10,R11                                      ;
                5E                10  C2 00002         SUBL2   #16, SP                              ;
                          04      AE  7C 00005         CLRQ    ERROR_ADDR                           ; 0211
                          0C      AE  D4 00008         CLRL    PFV_ADDR                             ;
                6D      0198      CF  DE 0000B         MOVAL   34$, (FP)                            ;
                05                6C  91 00010         CMPB    (AP), #5                             ; 0256
                                  05  1F 00013         BLSSU   1$                                   ;
          04    AE        14      AC  D0 00015         MOVL    ERROR, ERROR_ADDR                    ; 0240
                56        04      AC  D0 0001A 1$:     MOVL    PFV, R6                              ; 0246
```

```
                 0C   AE          56   D0 0001E          MOVL    R6, PFV_ADDR                                    ;
                      00000000G   00   16 00022          JSB     PAS$$VALIDATE_PFV                               ; 0252
                 08   AE          01   D0 00028          MOVL    #1, UNWIND_ACT                                  ; 0258
                      5A    04    A6   9E 0002C          MOVAB   4(R6), R10                                      ; 0264
                      06    02    AA   E8 00030          BLBS    2(R10), 2$                                      ;
                      00000000G   00   16 00034          JSB     PAS$$OPEN_IMPLICIT                              ; 0266
           08         6A          1D   E0 0003A 2$:      BBS     #29, (R10), 3$                                  ; 0272
                                  7E   D4 0003E          CLRL    -(SP)                                           ; 0274
                      7E    00G   8F   9A 00040          MOVZBL  #PAS$K_FILNOTOPE, -(SP)                         ;
                                  0E   11 00044          BRB     4$                                              ;
                      5B    FC    A7   9E 00046 3$:      MOVAB   -4(FCB), R11                                    ; 0280
           09         6B          02   E0 0004A          BBS     #2, (R11), 5$                                   ;
                                  7E   D4 0004E          CLRL    -(SP)                                           ; 0282
                      7E    00G   8F   9A 00050          MOVZBL  #PAS$K_FILNOTKEY, -(SP)                         ;
                                00D2   31 00054 4$:      BRW     23$                                             ;
                      55    08    AC   D0 00057 5$:      MOVL    KEY_NUMBER, R5                                  ; 0289
                 D0   A7          55   D1 0005B          CMPL    R5, -48(FCB)                                    ;
                                  0B   1F 0005F          BLSSU   6$                                              ;
                                  55   DD 00061          PUSHL   R5                                              ; 0291
                                  01   DD 00063          PUSHL   #1                                              ;
                      7E    00G   8F   9A 00065          MOVZBL  #PAS$K_KEYNOTDEF, -(SP)                         ;
                                008E   31 00069          BRW     17$                                             ;
                 35   A7          55   90 0006C 6$:      MOVB    R5, 53(FCB)                                     ; 0292
                 1E   A7          01   90 00070          MOVB    #1, 30(FCB)                                     ; 0293
                      58    CC    A7   D0 00074          MOVL    -52(FCB), KEYTYPES_VECTOR                       ; 0303
           52         55          01   78 00078          ASHL    #1, R5, R2                                      ; 0304
                      59        6842   B0 0007C          MOVW    (KEYTYPES_VECTOR)[R2], KEYTYPES_MASK            ;
                      53    0C    AC   D0 00080          MOVL    KEY_VALUE, R3                                   ; 0310
                      54    02    A3   9A 00084          MOVZBL  2(R3), R4                                       ;
                      0F          54   91 00088          CMPB    R4, #15                                         ;
                                  65   1A 0008B          BGTRU   16$                                             ;
           61         59          54   E1 0008D          BBC     R4, KEYTYPES_MASK, 16$                          ; 0313
                      0E          54   91 00091          CMPB    R4, #14                                         ; 0321
                                  04   13 00094          BEQL    7$                                              ;
                                  54   D5 00096          TSTL    R4                                              ; 0322
                                  12   12 00098          BNEQ    8$                                              ;
                 30   A7    04    A3   D0 0009A 7$:      MOVL    4(R3), 48(FCB)                                  ; 0325
              00FF    8F          63   B1 0009F          CMPW    (R3), #255                                      ; 0326
                                  4C   1A 000A4          BGTRU   16$                                             ;
                 34   A7          63   90 000A6          MOVB    (R3), 52(FCB)                                   ; 0329
                                  63   11 000AA          BRB     20$                                             ; 0321
                      53    04    A3   D0 000AC 8$:      MOVL    4(R3), KEYVAL                                   ; 0341
                 06               54   8F 000B0          CASEB   R4, #2, #6                                      ; 0343
      006F  001A          0015        0010       000B4 9$:      .WORD   10$-9$,-                                        ;
            001A          006F        006F       000BC                  11$-9$,-                                 ;
                                                                        12$-9$,-                                 ;
                                                                        22$-9$,-                                 ;
                                                                        22$-9$,-                                 ;
                                                                        22$-9$,-                                 ;
                                                                        12$-9$                                   ;
                                  5F   11 000C2          BRB     22$                                             ; 0351
                      6E          63   9A 000C4 10$:     MOVZBL  (KEYVAL), LOCAL_KEY                             ; 0346
                                  08   11 000C7          BRB     13$                                             ;
                      6E          63   3C 000C9 11$:     MOVZWL  (KEYVAL), LOCAL_KEY                             ; 0347
                                  03   11 000CC          BRB     13$                                             ;
                      6E          63   D0 000CE 12$:     MOVL    (KEYVAL), LOCAL_KEY                             ; 0349
                      53    02 A842   B0 000D1 13$:     MOVW    2(KEYTYPES_VECTOR)[R2], KEY_SIZE                ; 0359
```

```
                                01          53  B1 000D6          CMPW    KEY_SIZE, #1                                    ; 0360
                                            09  12 000D9          BNEQ    14$
                          000000FF  8F      6E  D1 000DB          CMPL    LOCAL_KEY, #255                                 ; 0363
                                            0C  11 000E2          BRB     15$
                                02          53  B1 000E4 14$:     CMPW    KEY_SIZE, #2                                    ; 0367
                                            19  12 000E7          BNEQ    18$
                          0000FFFF  8F      6E  D1 000E9          CMPL    LOCAL_KEY, #65535                               ; 0370
                                            15  1B 000F0 15$:     BLEQU   19$
                                            55  DD 000F2 16$:     PUSHL   R5                                              ; 0372
                                            01  DD 000F4          PUSHL   #1
                                7E      00G 8F  9A 000F6          MOVZBL  #PAS$K_KEYVALINC, -(SP)
                          00000000G     00      03  FB 000FA 17$:     CALLS   #3, PAS$$SIGNAL
                                            04  00101          RET
                                04          53  B1 00102 18$:     CMPW    KEY_SIZE, #4                                    ; 0374
                                            1C  12 00105          BNEQ    22$
                              30  A7        6E  9E 00107 19$:     MOVAB   LOCAL_KEY, 48(FCB)                              ; 0382
                              34  A7        53  90 0010B          MOVB    KEY_SIZE, 52(FCB)                               ; 0383
                                53      04  A7  9E 0010F 20$:     MOVAB   4(FCB), R3                                      ; 0390
                                02  A3  60  8F  8A 00113          BICB2   #96, 2(R3)                                      ; 0391
                            02      00  10  AC  8F 00118          CASEB   MATCH_TYPE, #0, #2                              ; 0393
                        001A        0014    001F    0011D 21$:     .WORD   26$-21$,-
                                                                            24$-21$,-
                                                                            25$-21$
                                            7E  D4 00123 22$:     CLRL    -(SP)                                           ; 0400
                                7E      00G 8F  9A 00125          MOVZBL  #PAS$K_INVARGPAS, -(SP)
                          00000000G     00      02  FB 00129 23$:     CALLS   #2, PAS$$SIGNAL
                                            04  00130          RET
                                02  A3          20  88 00131 24$:     BISB2   #32, 2(R3)                                      ; 0397
                                            05  11 00135          BRB     26$
                                02  A3  40  8F  88 00137 25$:     BISB2   #64, 2(R3)                                      ; 0398
                                02  AA      06  8A 0013C 26$:     BICB2   #6, 2(R10)                                      ; 0405
                                01  AB          20  8A 00140          BICB2   #32, 1(R11)                                     ; 0406
                                02  A3          10  8A 00144          BICB2   #16, 2(R3)                                      ; 0407
                                            57  DD 00148 27$:     PUSHL   FCB                                             ; 0413
                          00000000G     00      01  FB 0014A          CALLS   #1, SYS$GET
                                        0D      50  E8 00151          BLBS    $$STATUS, 28$
                          0001825A  8F          50  D1 00154          CMPL    $$STATUS, #98906
                                            04  12 0015B          BNEQ    28$
                                E7      03  AB  E8 0015D          BLBS    3(R11), 27$
                        05      F8  A7          02  E1 00161 28$:     BBC     #2, -8(FCB), 29$                        ; 0420
                                00  B6  22  A7  B0 00166          MOVW    34(FCB), @0(R6)                                 ; 0422
                                24          50  E8 0016B 29$:     BLBS    STATUS, 32$                                     ; 0424
                          000182B2  8F          50  D1 0016E          CMPL    STATUS, #98994                                  ; 0432
                                            1F  13 00175          BEQL    33$
                          000182AA  8F          50  D1 00177          CMPL    STATUS, #98986                                  ; 0434
                                            06  12 0017E          BNEQ    30$
                                7E      00G 8F  9A 00180          MOVZBL  #PAS$K_FAIGETLOC, -(SP)                         ; 0436
                                            04  11 00184          BRB     31$
                                7E      00G 8F  9A 00186 30$:     MOVZBL  #PAS$K_ERRDURFIN, -(SP)                         ; 0438
                          00000000G     00      01  FB 0018A 31$:     CALLS   #1, PAS$$SIGNAL
                                            04  00191          RET
                                02  AA          02  88 00192 32$:     BISB2   #2, 2(R10)                                      ; 0441
                                01  AB          08  88 00196 33$:     BISB2   #8, 1(R11)                                      ; 0449
                                01  AB          10  8A 0019A          BICB2   #16, 1(R11)                                     ; 0450
                                        D4  A7  D4  0019E          CLRL    -44(FCB)                                        ; 0451
                                03  AA      80  8F  8A 001A1          BICB2   #128, 3(R10)                                    ; 0452
                                            04  001A6          RET                                                     ; 0456
```

```
                                                0000 001A7 34$:     .WORD   Save nothing                             ; 0211
                                  50    08  AC  D0 001A9           MOVL    8(AP), R0                               ;
                                  50    04  A0  D0 001AD           MOVL    4(R0), R0                               ;
                                        F4  A0  9F 001B1           PUSHAB  ERROR_ADDR                              ;
                                        F8  A0  9F 001B4           PUSHAB  UNWIND_ACT                              ;
                                        FC  A0  9F 001B7           PUSHAB  PFV_ADDR                                ;
                                            03  DD 001BA           PUSHL   #3                                      ;
                                            5E  DD 001BC           PUSHL   SP                                      ;
                                  7E    04  AC  7D 001BE           MOVQ    4(AP), -(SP)                            ;
                         00000000G  00        03  FB 001C2           CALLS   #3, PAS$$IO_HANDLER                     ;
                                                04 001C9           RET                                             ;

; Routine Size:  458 bytes,    Routine Base:  _PAS$CODE + 0000

;  396          0457  1
;  397          0458  1 !<BLF/PAGE>
```

```
;  399          0459  1 END                                   ! End of module PAS$FINDK
;  400          0460  1
;  401          0461  0 ELUDOM
```

```
;                                  PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  _PAS$CODE                       458  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         25         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$:PASFINDK/OBJ=OBJ$:PASFINDK MSRC$:PASFINDK/UPDATE=(ENH$:PASFINDK)

; S ze:          458 code + 0 data bytes
; Run Time:        00:12.4
; Elapsed Time:    00:44.5
; Lines/CPU Min:    2232
; Lexemes/CPU-Min: 21283
; Memory Used:  194 pages
; Compilation Complete
```

PASFVOUTP
LIS

PASEOLN2
LIS

PASHEAP
LIS

PASHANDLE
LIS

PASFAB
LIS

PASGET
LIS

PASCVTRT
LIS

PASDATE
LIS

PASEOF2
LIS

PASFINDK
LIS

PASFVINPU
LIS

PASEXPO
LIS

PASGOTO
LIS

PASFILEUT
LIS

PASHALT
LIS

PASDELETE
LIS

PASFIND2
LIS