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
```

```
PPPPPPPP      AAAAAA      SSSSSSSS    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT
PPPPPPPP      AAAAAA      SSSSSSSS    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT
PP      PP  AA      AA  SS          GG          EE              TT
PP      PP  AA      AA  SS          GG          EE              TT
PP      PP  AA      AA  SS          GG          EE              TT
PP      PP  AA      AA  SS          GG          EE              TT
PPPPPPPP    AA      AA    SSSSSS    GG          EEEEEEEE        TT
PPPPPPPP    AA      AA    SSSSSS    GG          EEEEEEEE        TT
PP          AAAAAAAAAA          SS  GG  GGGGGG  EE              TT
PP          AAAAAAAAAA          SS  GG  GGGGGG  EE              TT
PP          AA      AA          SS  GG      GG  EE              TT          ....
PP          AA      AA          SS  GG      GG  EE              TT          ....
PP          AA      AA  SSSSSSSS      GGGGGG    EEEEEEEEEE      TT          ....
PP          AA      AA  SSSSSSSS      GGGGGG    EEEEEEEEEE      TT          ....

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$GET ( %TITLE 'GET procedure'
0                 IDENT = '1-002'          ! File: PASGET.B32 Edit: SBL1002
0                 ) =
1 BEGIN
1 !
1 !***************************************************************************
1 !*                                                                         *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
1 !*  ALL RIGHTS RESERVED.                                                   *
1 !*                                                                         *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
1 !*  TRANSFERRED.                                                           *
1 !*                                                                         *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
1 !*  CORPORATION.                                                           *
1 !*                                                                         *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
1 !*                                                                         *
1 !*                                                                         *
1 !***************************************************************************
1 !
1
1 !++
1 ! FACILITY:     Pascal Language Support
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains PAS$GET, which implements the
1 !       VAX-11 Pascal GET procedure.
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 - Turn off NLK bit after $GET.  SPR 11-55690  SBL 8-Apr-1983
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$GET: NOVALUE,                             ! Do a GET
;    62         0124  1     PAS$GET_UNLOCK: NOVALUE,                      ! Do a GET and UNLOCK
;    63         0125  1     PAS$$GET: JSB_GET NOVALUE;                    ! Internally callable
;    64         0126  1
;    65         0127  1 !
;    66         0128  1 ! MACROS:
;    67         0129  1 !
;    68         0130  1 !       NONE
;    69         0131  1 !
;    70         0132  1 ! EQUATED SYMBOLS:
;    71         0133  1 !
;    72         0134  1 !       NONE
;    73         0135  1 !
;    74         0136  1 ! FIELDS:
;    75         0137  1 !
;    76         0138  1 !       NONE
;    77         0139  1 !
;    78         0140  1 ! OWN STORAGE:
;    79         0141  1 !
;    80         0142  1 !       NONE
```

```
:   82          0143  1 %SBTTL 'PAS$GET - GET procedure'
:   83          0144  1 GLOBAL ROUTINE PAS$GET (                              ! Do a GET
:   84          0145  1         PFV: REF $PAS$PFV_FILE_VARIABLE,              ! File variable
:   85          0146  1         ERROR                                         ! Unwind address if error
:   86          0147  1     ): NOVALUE =
:   87          0148  1
:   88          0149  1 !++
:   89          0150  1 ! FUNCTIONAL DESCRIPTION:
:   90          0151  1 !
:   91          0152  1 !       PAS$GET implements the VAX-11 Pascal GET procedure.  It
:   92          0153  1 !       reads a record from the file and places its contents in
:   93          0154  1 !       the file buffer.
:   94          0155  1 !
:   95          0156  1 ! CALLING SEQUENCE:
:   96          0157  1 !
:   97          0158  1 !       CALL PAS$GET (PFV.mr.r [, ERROR.j.r])
:   98          0159  1 !
:   99          0160  1 ! FORMAL PARAMETERS:
:  100          0161  1 !
:  101          0162  1 !       PFV             - The Pascal File Variable (PFV) passed by reference.
:  102          0163  1 !                         The structure of the PFV is defined in PASPFV.REQ.
:  103          0164  1 !
:  104          0165  1 !       ERROR           - Optional.  If specified, the address to unwind to
:  105          0166  1 !                         if an error occurs.
:  106          0167  1 !
:  107          0168  1 ! IMPLICIT INPUTS:
:  108          0169  1 !
:  109          0170  1 !       NONE
:  110          0171  1 !
:  111          0172  1 ! IMPLICIT OUTPUTS:
:  112          0173  1 !
:  113          0174  1 !       NONE
:  114          0175  1 !
:  115          0176  1 ! ROUTINE VALUE:
:  116          0177  1 !
:  117          0178  1 !       NONE
:  118          0179  1 !
:  119          0180  1 ! SIDE EFFECTS:
:  120          0181  1 !
:  121          0182  1 !       Opens standard files INPUT or OUTPUT if not already open.
:  122          0183  1 !       Turns off RAB$V_NLK which might have been set by PAS$GET_UNLOCK
:  123          0184  1 !
:  124          0185  1 ! SIGNALLED ERRORS:
:  125          0186  1 !
:  126          0187  1 !       FILNOTOPE - file not open
:  127          0188  1 !       FILNOTINS - file not in Inspection mode
:  128          0189  1 !
:  129          0190  1 !--
:  130          0191  1
:  131          0192  2     BEGIN
:  132          0193  2
:  133          0194  2     LOCAL
:  134          0195  2         FCB: REF $PAS$FCB_CONTROL_BLOCK,              ! File control block
:  135          0196  2         PFV_ADDR: VOLATILE,                           ! Enable argument
:  136          0197  2         UNWIND_ACT: VOLATILE,                         ! Enable argument
:  137          0198  2         ERROR_ADDR: VOLATILE;                         ! Enable argument
:  138          0199  2
:  139          0200  2       BIND
:  140          0201  2           RAB = FCB: REF BLOCK [, BYTE];                       ! RAB is FCB
:  141          0202  2
:  142          0203  2       BUILTIN
:  143          0204  2           ACTUALCOUNT,
:  144          0205  2           TESTBITSS;
:  145          0206  2
:  146          0207  2       ENABLE
:  147          0208  2           PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);
:  148          0209  2
:  149          0210  2       IF ACTUALCOUNT () GEQU 2
.  150          0211  2       THEN
:  151          0212  2           ERROR_ADDR = .ERROR;                                 ! Set unwind address
:  152          0213  2
:  153          0214  2       !+
:  154          0215  2       ! Set PFV address enable argument.
:  155          0216  2       !-
:  156          0217  2
:  157          0218  2       PFV_ADDR = PFV [PFV$R_PFV];
:  158          0219  2
:  159          0220  2       !+
:  160          0221  2       ! Validate and lock PFV
:  161          0222  2       !-
:  162          0223  2
:  163          0224  2       PAS$$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
:  164          0225  2
:  165          0226  2       !+
:  166          0227  2       ! Set unwind action to unlock file.
:  167          0228  2       !-
:  168          0229  2
:  169          0230  2       UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
:  170          0231  2
:  171          0232  2       !+
:  172          0233  2       ! Resolve lazy lookahead, if any.  Also may open INPUT or OUTPUT.
:  173          0234  2       !-
:  174          0235  2
:  175          0236  2       IF NOT .PFV [PFV$V_VALID]
:  176          0237  2       THEN
:  177          0238  2           PAS$$LOOK_AHEAD (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
:  178          0239  2
:  179          0240  2       !+
:  180          0241  2       ! Verify that the file is open.
:  181          0242  2       !-
:  182          0243  2
:  183          0244  2       IF NOT .PFV [PFV$V_OPEN]
:  184          0245  2       THEN
:  185          0246  2           $PAS$IO_ERROR (PAS$_FILNOTOPE,0);
:  186          0247  2
:  187          0248  2       !+
:  188          0249  2       ! Verify that we are in Inspection mode
:  189          0250  2       !-
:  190          0251  2
:  191          0252  2       IF NOT .FCB [FCB$V_INSPECTION]
:  192          0253  2       THEN
:  193          0254  2           $PAS$IO_ERROR (PAS$_FILNOTINS,0);
:  194          0255  2
:  195          0256  2       !+
;  196          0257  2     ! Turn off RAB$V_NLK.
;  197          0258  2     !-
;  198          0259  2
;  199          0260  2     RAB [RAB$V_NLK] = 0;
;  200          0261  2
;  201          0262  2     !+
;  202          0263  2     ! Do the GET
;  203          0264  2     !-
;  204          0265  2
;  205          0266  2     PAS$$GET (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
;  206          0267  2
;  207          0268  2     !+
;  208          0269  2     ! Indicate successful completion
;  209          0270  2     ! Unlock the file
;  210          0271  2     !-
;  211          0272  2
;  212          0273  2     FCB [FCB$L_STATUS] = 0;
;  213          0274  2     PFV [PFV$V_LOCK] = 0;
;  214          0275  2
;  215          0276  2     RETURN;
;  216          0277  2
;  217          0278  1     END;                                         ! End of routine PAS$GET


                                                                           .TITLE  PAS$GET GET procedure
                                                                           .IDENT  \1-002\

                                                                           .EXTRN  PAS$GET, PAS$GET_UNLOCK
                                                                           .EXTRN  PAS$$GET, PAS$$IO_HANDLER
                                                                           .EXTRN  PAS$$VALIDATE_PFV
                                                                           .EXTRN  PAS$$LOOK_AHEAD
                                                                           .EXTRN  PAS$$SIGNAL, PAS$K_FILNOTOPE
                                                                           .EXTRN  PAS$K_FILNOTINS

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

                                                        00CC 00000         .ENTRY  PAS$GET, Save R2,R3,R6,R7                    ; 0144
                                        5E            08  C2 00002         SUBL2   #8, SP                                       ;
                                                      7E  D4 00005         CLRL    ERROR_ADDR                                   ; 0192
                                                  04  AE  7C 00007         CLRQ    UNWIND_ACT                                   ;
                                        6D      0056  CF  DE 0000A         MOVAL   6$, (FP)                                     ;
                                        02            6C  91 0000F         CMPB    (AP), #2                                     ; 0210
                                                      04  1F 00012         BLSSU   1$                                           ;
                                        6E        08  AC  D0 00014         MOVL    ERROR, ERROR_ADDR                            ; 0212
                                        56        04  AC  D0 00018 1$:     MOVL    PFV, R6                                      ; 0218
                                  08    AE            56  D0 0001C         MOVL    R6, PFV_ADDR                                 ;
                                           00000000G  00  16 00020         JSB     PAS$$VALIDATE_PFV                            ; 0224
                                  04    AE            01  D0 00026         MOVL    #1, UNWIND_ACT                               ; 0230
                                        06        06  A6  E8 0002A         BLBS    6(R6), 2$                                    ; 0236
                                           00000000G  00  16 0002E         JSB     PAS$$LOOK_AHEAD                              ; 0238
                         08       07    A6            05  E0 00034 2$:     BBS     #5, 7(R6), 3$                                ; 0244
                                                      7E  D4 00039         CLRL    -(SP)                                        ; 0246
                                        7E       00G  8F  9A 0003B         MOVZBL  #PAS$K_FILNOTOPE, -(SP)                      ;
                                                      0B  11 0003F         BRB     4$                                           ;
                         0E       FD    A7            03  E0 00041 3$:     BBS     #3, -3(FCB), 5$                              ; 0252
                                                      7E  D4 00046         CLRL    -(SP)                                        ; 0254
                                        7E       00G  8F  9A 00048         MOVZBL  #PAS$K_FILNOTINS, -(SP)                      ;
                         00000000G  00              02  FB 0004C 4$:      CALLS   #2, PAS$$SIGNAL                              ;
                                                    04 00053              RET                                                  ;
                                06  A7              10  8A 00054 5$:      BICB2   #16, 6(FCB)                                  ; 0260
                                                 0000V  30 00058          BSBW    PAS$$GET                                     ; 0266
                                             D4     A7  D4 0005B          CLRL    -44(FCB)                                     ; 0273
                                07  A6       80     8F  8A 0005E          BICB2   #128, 7(R6)                                  ; 0274
                                                    04 00063              RET                                                  ; 0278
                                                  0000 00064 6$:          .WORD   Save nothing                                 ; 0192
                                    50       08     AC  D0 00066          MOVL    8(AP), R0                                    ;
                                    50       04     A0  D0 0006A          MOVL    4(R0), R0                                    ;
                                             F4     A0  9F 0006E          PUSHAB  ERROR_ADDR                                   ;
                                             F8     A0  9F 00071          PUSHAB  UNWIND_ACT                                   ;
                                             FC     A0  9F 00074          PUSHAB  PFV_ADDR                                     ;
                                                    03  DD 00077          PUSHL   #3                                           ;
                                                    5E  DD 00079          PUSHL   SP                                           ;
                                    7E       04     AC  7D 0007B          MOVQ    4(AP), -(SP)                                 ;
                         00000000G  00              03  FB 0007F          CALLS   #3, PAS$$IO_HANDLER                          ;
                                                    04 00086              RET                                                  ;

; Routine Size:  135 bytes,    Routine Base:  _PAS$CODE + 0000


;  218          0279  1
;  219          0280  1 !<BLF/PAGE>
```

```
1 %SBTTL 'PAS$GET_UNLOCK - GET procedure with UNLOCK'
1 GLOBAL ROUTINE PAS$GET_UNLOCK (
1           PFV: REF $PAS$PFV_FILE_VARIABLE,                ! File variable
1           ERROR                                           ! Unwind address if error
1     ): NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         PAS$GET_UNLOCK is called by the compiled code when a READ is
1 !         done for a non-textfile.  It does the equivalent of a GET followed
1 !         by an UNLOCK.  What it actually does is set RAB$V_NLK which will
1 !         cause the next $GET to not lock the record.  This bit is reset by
1 !         PAS$FIND2, PAS$FINDK, PAS$GET, PAS$RESET and PAS$RESETK and
1 !         PAS$REWRITE2, which are the only procedures
1 !         which do $GETs on non-textfiles.  It doesn't matter for textfiles
1 !         since they may only be sequential organization.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !         CALL PAS$GET_UNLOCK (PFV.mr.r [, ERROR.j.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 !         ERROR           - Optional.  If specified, the address to unwind to
1 !                           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 !         Opens standard files INPUT or OUTPUT if not already open.
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !         FILNOTOPE - file not open
1 !         FILNOTINS - file not in Inspection mode
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2           FCB: REF $PAS$FCB_CONTROL_BLOCK,               ! File control block
```

```
2          PFV_ADDR: VOLATILE,                             ! Enable argument
2          UNWIND_ACT: VOLATI.E,                           ! Enable argument
2          ERROR_ADDR: VOLATILE;                           ! Enable argument
2
2      BIND
2          RAB = FCB: REF BLOCK [, BYTE];                  ! RAB is FCB address
2
2      BUILTIN
2          ACTUALCOUNT,
2          TESTBITSS;
2
2      ENABLE
2          PAS$$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);
2
2      IF ACTUALCOUNT () GEQU 2
2      THEN
2          ERROR_ADDR = .ERROR;                            ! Set unwind address
2
2      !+
2      ! Set PFV address enable argument.
2      !-
2
2      PFV_ADDR = PFV [PFV$R_PFV];
2
2      !+
2      ! Validate and lock PFV
2      !-
2
2      PAS$$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      ! Resolve lazy lookahead, if any.  Also may open INPUT or OUTPUT.
2      !-
2
2      IF NOT .PFV [PFV$V_VALID]
2      THEN
2          PAS$$LOOK_AHEAD (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
2
2      !+
2      ! Verify that the file is open.
2      !-
2
2      IF NOT .PFV [PFV$V_OPEN]
2      THEN
2          $PAS$IO_ERROR (PAS$_FILNOTOPE,0);
2
2      !+
2      ! Verify that we are in Inspection mode
2      !-
2
2      IF NOT .FCB [FCB$V_INSPECTION]
```

```
;   335         0395  2      THEN
;   336         0396  2          $PAS$IO_ERROR (PAS$_FILNOTINS,0);
;   337         0397  2
;   338         0398  2      !+
;   339         0399  2      ! If this is not a textfile, set RAB$V_NLK.
;   340         0400  2      !-
;   341         0401  2
;   342         0402  2      IF NOT .FCB [FCB$V_TEXT]
;   343         0403  2      THEN
;   344         0404  2          RAB [RAB$V_NLK] = 1;
;   345         0405  2
;   346         0406  2      !+
;   347         0407  2      ! Do the GET
;   348         0408  2      !-
;   349         0409  2
;   350         0410  2      PAS$$GET (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
;   351         0411  2
;   352         0412  2      !+
;   353         0413  2      ! Indicate successful completion
;   354         0414  2      ! Unlock the file
;   355         0415  2      !-
;   356         0416  2
;   357         0417  2      FCB [FCB$L_STATUS] = 0;
;   358         0418  2      PFV [PFV$V_LOCK] = 0;
;   359         0419  2
;   360         0420  2      RETURN;
;   361         0421  2
;   362         0422  1      END;                                          ! End of routine PAS$GET_UNLOCK



                                     00CC 00000          .ENTRY   PAS$GET_UNLOCK, Save R2,R3,R6,R7                    ; 0282
                  5E              08   C2 00002          SUBL2    #8, SP                                              ;
                                  7E   D4 00005          CLRL     ERROR_ADDR                                          ; 0334
                         04       AE   7C 00007          CLRQ     UNWIND_ACT                                          ;
                  6D   005A       CF   DE 0000A          MOVAL    7$, (FP)                                            ;
                  02              6C   91 0000F          CMPB     (AP), #2                                            ; 0352
                                  04   1F 00012          BLSSU    1$                                                  ;
                  6E     08       AC   D0 00014          MOVL     ERROR, ERROR_ADDR                                   ; 0354
                  56     04       AC   D0 00018 1$:      MOVL     PFV, R6                                             ; 0360
             08   AE              56   D0 0001C          MOVL     R6, PFV_ADDR                                        ;
                     00000000G    00   16 00020          JSB      PAS$$VALIDATE_PFV                                   ; 0366
             04   AE              01   D0 00026          MOVL     #1, UNWIND_ACT                                      ; 0372
                  06     06       A6   E8 0002A          BLBS     6(R6), 2$                                           ; 0378
                     00000000G    00   16 0002E          JSB      PAS$$LOOK_AHEAD                                     ; 0380
     08      07   A6              05   E0 00034 2$:      BBS      #5, 7(R6), 3$                                       ; 0386
                                  7E   D4 00039          CLRL     -(SP)                                               ; 0388
                  7E    00G       8F   9A 0003B          MOVZBL   #PAS$K_FILNOTOPE, -(SP)                             ;
                                  0B   11 0003F          BRB      4$                                                  ;
     0E      FD   A7              03   E0 00041 3$:      BBS      #3, -3(FCB), 5$                                     ; 0394
                                  7E   D4 00046          CLRL     -(SP)                                               ; 0396
                  7E    00G       8F   9A 00048          MOVZBL   #PAS$K_FILNOTINS, -(SP)                             ;
        00000000G 00              02   FB 0004C 4$:      CALLS    #2, PAS$$SIGNAL                                     ;
                                       04 00053          RET                                                          ;
                  04     F8       A7   E8 00054 5$:      BLBS     -8(FCB), 6$                                         ; 0402
```

```
                          06    A7             10  88 00058         BISB2   #16, 6(FCB)                                  ; 0404
                                            0000V  30 0005C 6$:     BSBW    PAS$$GET                                     ; 0410
                                      D4     A7  D4 0005F           CLRL    -44(FCB)                                     ; 0417
                          07    A6    80     8F  8A 00062           BICB2   #128, 7(R6)                                  ; 0418
                                                 04 00067           RET                                                  ; 0422
                                              0000 00068 7$:        .WORD   Save nothing                                 ; 0334
                                50    08     AC  D0 0006A           MOVL    8(AP), R0                                    ;
                                50    04     A0  D0 0006E           MOVL    4(R0), R0                                    ;
                                      F4     A0  9F 00072           PUSHAB  ERROR_ADDR                                   ;
                                      F8     A0  9F 00075           PUSHAB  UNWIND_ACT                                   ;
                                      FC     A0  9F 00078           PUSHAB  PFV_ADDR                                     ;
                                             03  DD 0007B           PUSHL   #3                                           ;
                                             5E  DD 0007D           PUSHL   SP                                           ;
                                7E    04     AC  7D 0007F           MOVQ    4(AP), -(SP)                                 ;
                     00000000G  00           03  FB 00083           CALLS   #3, PAS$$IO_HANDLER                          ;
                                                 04 0008A           RET                                                  ;

; Routine Size:  139 bytes,    Routine Base:  _PAS$CODE + 0087


;  363          0423  1
;  364          0424  1 !<BLF/PAGE>
```

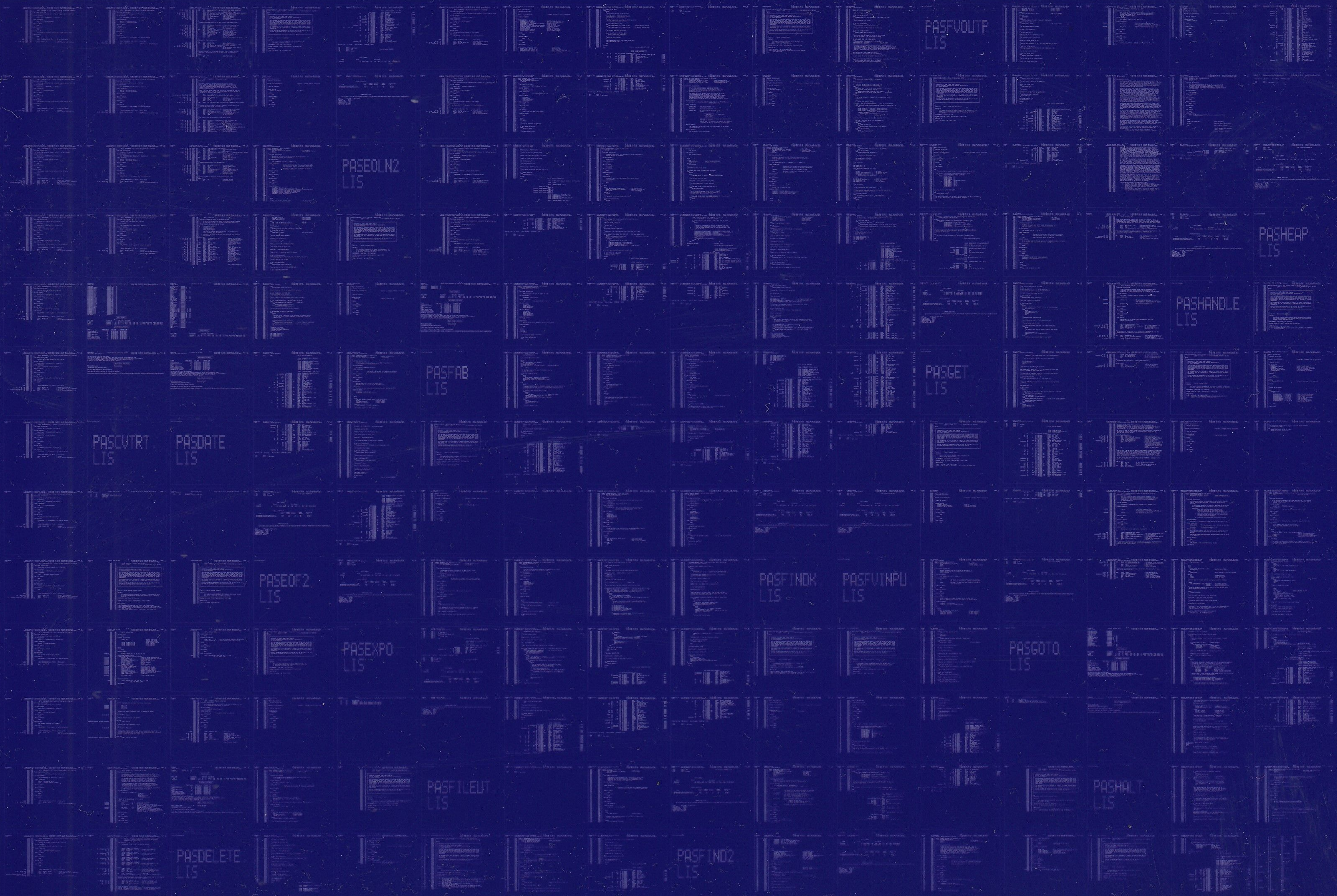
```
1 %SBTTL 'PAS$$GET - Internal'y callable GET'
1 GLOBAL ROUTINE PAS$$GET (                                    ! Do a GET
1         PFV: REF $PAS$PFV_FILE_VARIABLE,                      ! File variable
1         FCB: REF $PAS$FCB_CONTROL_BLOCK                       ! Control Block
1     ): JSB_GET NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       PAS$$GET implements the VAX-11 Pascal GET procedure.  It
1 !       reads a record from the file and places its contents in
1 !       the file buffer.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       JSB PAS$$GET (PFV.mr.r, FCB.mr.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 !       FCB             - The File Control Block for the file.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       It is assumed that the file is open, is in inspection mode and is locked.
1 !       It is assumed that lazy lookahead is not in progress.
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 !       FAIGETLOC - failed to GET locked component
1 !       GETAFTEOF - GET attempted after end-of-file
1 !       ERRDURGET - error during GET
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         STATUS;
2
2     BIND
2         RAB = FCB: REF BLOCK [, BYTE];
2
```

```
;   423        0482  2        !+
;   424        0483  2        ! Set initial conditions of PFV flags
;   425        0484  2        !-
;   426        0485  2
;   427        0486  2        PFV [PFV$V_DFB] = 0;          ! Buffer undefined
;   428        0487  2        PFV [PFV$V_EOF_DEFINED] = 1;! EOF defined after GET
;   429        0488  2
;   430        0489  2        !+
;   431        0490  2        ! If EOF is true, then signal an error.
;   432        0491  2        !-
;   433        0492  2
;   434        0493  2        IF .FCB [FCB$V_EOF]
;   435        0494  2        THEN
;   436        0495  2            $PAS$IO_ERROR (PAS$_GETAFTEOF,0);
;   437        0496  2
;   438        0497  2        !+
;   439        0498  2        ! GET from a textfile is different...
;   440        0499  2        !-
;   441        0500  2
;   442        0501  2        IF .FCB [FCB$V_TEXT]
;   443        0502  2        THEN
;   444        0503  3            BEGIN
;   445        0504  3            !+
;   446        0505  3            ! Are we already at end-of-line?  If so, set lazy-lookahead
;   447        0506  3            !-
;   448        0507  3            IF .PFV [PFV$V_EOLN]
;   449        0508  3            THEN
;   450        0509  4                BEGIN
;   451        0510  4                FCB [FCB$V_LAZY] = 1;
;   452        0511  4                PFV [PFV$V_VALID] = 0;
;   453        0512  4                RETURN;
;   454        0513  3                END;
;   455        0514  3
;   456        0515  3            !+
;   457        0516  3            ! Advance current record pointer
;   458        0517  3            !-
;   459        0518  3
;   460        0519  3            FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_CUR] + 1;
;   461        0520  3
;   462        0521  3            !+
;   463        0522  3            ! Are we now at end-of-line?
;   464        0523  3            !-
;   465        0524  3
;   466        0525  3            IF .FCB [FCB$A_RECORD_CUR] GEQA .FCB [FCB$A_RECORD_END]
;   467        0526  3            THEN
;   468        0527  4                BEGIN
;   469        0528  4                !+
;   470        0529  4                ! At end-of-line.  Set EOL, move a blank to user buffer.
;   471        0530  4                !-
;   472        0531  4
;   473        0532  4                PFV [PFV$V_EOLN] = 1;
;   474        0533  4                CH$WCHAR (%C' ', .PFV [PFV$A_BUFFER]);
;   475        0534  4                END
;   476        0535  3            ELSE
;   477        0536  3                !+
;   478        0537  3                ! Otherwise move new character to user buffer
;   479        0538  3                !-
;  480          0539  3
;  481          0540  3               CH$WCHAR (..FCB [FCB$A_RECORD_CUR], .PFV [PFV$A_BUFFER]);
;  482          0541  3
;  483          0542  3           !+
;  484          0543  3           ! Mark buffer and PFV as valid and return immediate'y.
;  485          0544  3           !-
;  486          0545  3
;  487          0546  3           PFV [PFV$V_VALID] = 1;
;  488          0547  3           PFV [PFV$V_DFB] = 1;
;  489          0548  3           RETURN;
;  490          0549  2           END;
;  491          0550  2
;  492          0551  2       !+
;  493          0552  2       ! Not textfile, call RMS to get a record.
;  494          0553  2       !-
;  495          0554  2
;  496          0555  2       RAB [RAB$B_RAC] = RAB$C_SEQ;          ! Set sequential access mode
;  497          0556  2       STATUS = $PAS$RMS_OP ($GET (RAB=.RAB));
;  498          0557  2
;  499          0558  2       !+
;  500          0559  2       ! If varying, put component length in first word of user buffer.
;  501          0560  2       ! Do this before we check for errors!
;  502          0561  2       !-
;  503          0562  2
;  504          0563  2       IF .FCB [FCB$V_VARYING]
;  505          0564  2       THEN
;  506          0565  2           (.PFV [PFV$A_BUFFER])<0,16,0> = .RAB [RAB$W_RSZ];
;  507          0566  2
;  508          0567  2       !+
;  509          0568  2       ! Turn off RAB$V_NLK now just to make sure we don't have problems
;  510          0569  2       ! later on.
;  511          0570  2       !-
;  512          0571  2
;  513          0572  2       RAB [RAR$V_NLK] = 0;
;  514          0573  2
;  515          0574  2       IF NOT .STATUS
;  516          0575  2       THEN
;  517          0576  3           BEGIN
;  518          0577  3           !+
;  519          0578  3           ! $GET failed.  If RMS$_EOF, simply set EOF condition.  Otherwise,
;  520          0579  3           ! signal an error.
;  521          0580  3           !-
;  522          0581  3           IF .RAB [RAB$L_STS] EQLU RMS$_EOF
;  523          0582  3           THEN
;  524          0583  3               FCB [FCB$V_EOF] = 1
;  525          0584  3           ELSE IF .RAB [RAB$L_STS] EQLU RMS$_RLK
;  526          0585  3           THEN
;  527          0586  4               $PAS$IO_ERROR (PAS$_FAIGETLOC)       ! Failed tc GET locked component
;  528          0587  3           ELSE
;  529          0588  3               $PAS$IO_ERROR (PAS$_ERRDURGET);
;  530          0589  3           PFV [PFV$V_DFB] = 0;
;  531          0590  3           END
;  532          0591  2       ELSE
;  533          0592  3           BEGIN
;  534          0593  3           FCB [FCB$V_EOF] = 0;
;  535          0594  3           PFV [PFV$V_DFB] = 1;
;  536          0595  2           END;
;  537        0596  2
;  538        0597  2        RETURN;
;  539        0598  2
;  540        0599  1        END;                                     . End of routine PAS$$GET


                                                                       .EXTRN   PAS$K_GETAFTEOF
                                                                       .EXTRN   SYS$GET, PAS$K_FAIGETLOC
                                                                       .EXTRN   PAS$K_ERRDURGET

                               52      04   A6   9E 00000 PAS$$GET::
                                                                       MOVAB    4(PFV), R2                          ; 0486
                        02     A2           02   8A 00004              BICB2    #2, 2(R2)                           ;
                        02     A2           04   88 00008              BISB2    #4, 2(R2)                           ; 0487
                               53      FC   A7   9E 0000C              MOVAB    -4(FCB), R3                         ; 0493
                 0E            63           0D   E1 00010              BBC      #13, (R3), 1$                       ;
                                            7E   D4 00014              CLRL     -(SP)                               ; 0495
                               7E      00G  8F   9A 00016              MOVZBL   #PAS$K_GETAFTEOF, -(SP)             ;
                    00000000G  00           02   FB 0001A              CALLS    #2, PAS$$SIGNAL                     ;
                                                 05 00021              RSB                                          ;
                               2C      F8   A7   E9 00022 1$:          BLBC     -8(FCB), 5$                         ; 0501
                 09            62           13   E1 00026              BBC      #19, (R2), 2$                       ; 0507
                        01     A3           04   88 0002A              BISB2    #4, 1(R3)                           ; 0510
                        02     A2           01   8A 0002E              BICB2    #1, 2(R2)                           ; 0511
                                                 05 00032              RSB                                          ; 0509
                                       EC   A7   D6 00033 2$:          INCL     -20(FCB)                            ; 0519
                        F0     A7      EC   A7   D1 00036              CMPL     -20(FCB), -16(FCB)                  ; 0525
                                            0A   1F 0003B              BLSSU    3$                                  ;
                        02     A2           08   88 0003D              BISB2    #8, 2(R2)                           ; 0532
                        00     B6           20   90 00041              MOVB     #32, @0(PFV)                        ; 0533
                                            05   11 00045              BRB      4$                                  ; 0540
                        00     B6      EC   B7   90 00047 3$:          MOVB     @-20(FCB), @0(PFV)                  ;
                        02     A2           01   88 0004C 4$:          BISB2    #1, 2(R2)                           ; 0546
                                            62   11 00050              BRB      14$                                 ; 0547
                                       1E   A7   94 00052 5$:          CLRB     30(FCB)                             ; 0555
                                            57   DD 00055 6$:          PUSHL    FCB                                 ; 0556
                    00000000G  00           01   FB 00057              CALLS    #1, SYS$GET                         ;
                               0D           50   E8 0005E              BLBS     $$STATUS, 7$                        ;
                    0001825A   8F           50   D1 00061              CMPL     $$STATUS, #98906                    ;
                                            04   12 00068              BNEQ     7$                                  ;
                               E7      03   A3   E8 0006A              BLBS     3(R3), 6$                           ;
                 05     F8     A7           02   E1 0006E 7$:          BBC      #2, -8(FCB), 8$                     ; 0563
                        00     B6      22   A7   B0 00073              MOVW     34(FCB), @0(PFV)                    ; 0565
                        06     A7           10   8A 00078 8$:          BICB2    #16, 6(FCB)                         ; 0572
                               31           50   E8 0007C              BLBS     STATUS, 13$                         ; 0574
                    0001827A   8F      08   A7   D1 0007F              CMPL     8(FCB), #98938                      ; 0581
                                            06   12 00087              BNEQ     9$                                  ;
                        01     A3           20   88 00089              BISB2    #32, 1(R3)                          ; 0583
                                            1C   11 00C8D              BRB      12$                                 ;
                    000182AA   8F      08   A7   D1 0008F 9$:          CMPL     8(FCB), #98986                      ; 0584
                                            06   12 00097              BNEQ     10$                                 ;
                               7E      00G  8F   9A 00099              MOVZBL   #PAS$K_FAIGETLOC, -(SP)             ; 0586
                                            04   11 0009D              BRB      11$                                 ;
                               7E      00G  8F   9A 0009F 10$:         MOVZBL   #PAS$K_ERRDURGET, -(SP)             ; 0588
                    00000000G  00           01   FB 000A3 11$:         CALLS    #1, PAS$$SIGNAL                     ;
                                                 05 000AA              RSB                                          ;
```

```
                     02   A2        02  8A 000AB 12$:     BICB2   #2, 2(R2)                          ; 0589
                                        05 000AF          RSB                                        ; 0574
                     01   A3        20  8A 000B0 13$:     BICB2   #32, 1(R3)                         ; 0593
                     02   A2        02  88 000B4 14$:     BISB2   #2, 2(R2)                          ; 0594
                                        05 000B8          RSB                                        ; 0599

; Routine Size:  185 bytes,    Routine Base:  _PAS$CODE + 0112


;  541          0600  1
;  542          0601  1 !<BLF/PAGE>
;  544          0602  1 END                                         ! End of module PAS$GET
;  545          0603  1
;  546          0604  0 ELUDOM



;                                 PSECT SUMMARY
;
;       Name                     Bytes                       Attributes
;
;    _PAS$CODE                        459  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        12         0         581        00:01.0
;    _$255$DUA28:[PASRTL.OBJ]PASLIB.L32;1           427        98        22          33        00:00.4



;                                 COMMAND QUALIFIERS

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

; Size:          459 code + 0 data bytes
; Run Time:         00:11.4
; Elapsed Time:     00:39.4
; Lines/CPU Min:      3181
; Lexemes/CPU-Min: 19359
; Memory Used:  109 pages
; Compilation Complete
```

0294 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



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