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

```
_$2

Sym
---

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

PAS

PAS

PAS

PAS
PAS
PAS
PAS
PAS

PAS
PAS
PAS
PAS
PAS
PAS
PAS
PAS
```

```
**FILE**ID**PASFILEUT                                  M  6

PPPPPPPP      AAAAAA      SSSSSSSS  FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE  UU      UU  TTTTTTTTTT
PPPPPPPP      AAAAAA      SSSSSSSS  FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE  UU      UU  TTTTTTTTTT
PP      PP  AA      AA  SS          FF              II      LL          EE          UU      UU      TT
PP      PP  AA      AA  SS          FF              II      LL          EE          UU      UU      TT
PP      PP  AA      AA  SS          FF              II      LL          EE          UU      UU      TT
PP      PP  AA      AA  SS          FF              II      LL          EE          UU      UU      TT
PPPPPPPP    AA      AA    SSSSSS    FFFFFFFF        II      LL          EEEEEEEE    UU      UU      TT
PPPPPPPP    AA      AA    SSSSSS    FFFFFFFF        II      LL          EEEEEEEE    UU      UU      TT
PP          AAAAAAAAAA          SS  FF              II      LL          EE          UU      UU      TT
PP          AAAAAAAAAA          SS  FF              II      LL          EE          UU      UU      TT
PP          AA      AA          SS  FF              II      LL          EE          UU      UU      TT          ....
PP          AA      AA          SS  FF              II      LL          EE          UU      UU      TT          ....
PP          AA      AA  SSSSSSSS    FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  UUUUUUUUUU      TT          ....
PP          AA      AA  SSSSSSSS    FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  UUUUUUUUUU      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$$FILE_UTIL ( %TITLE 'File manipulation utility procedures'
0               IDENT = '1-005'          ! File: PASFILEUT.B32 Edit: SBL1005
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 !       Utility procedures to manipulate the global list of files.
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 - Don't assume that PFV contains valid information in PAS$CLOSE_LOCAL_R3.
1 !         Use DO_CLOSE_HANDLER to display error messages from DO_CLOSE.
1 !         SBL 28-Jun-1982
1 ! 1-003 - Set all PFV fields that are needed to close the file in PAS$$CLOSE_LOCAL.
1 !         SBL 29-Jun-1982
1 ! 1-004 - Move FCB$L_STATUS to PFV$L_STATUS in PAS$$REMOVE_FILE.
1 !         QAR FT3-2  SBL 30-Aug-1982
1 ! 1-005 - Allow PAS$$REMOVE_FILE to be called without the queue having been
1 !         initialized.  This can occur if the first file opened in the program
1 !         fails to open and the OPEN is unwound.  SBL 10-Jan-1983
1 !--
1
```

```
0056  1 %SBTTL 'Declarations'
0057  1 !
0058  1 ! PROLOGUE DEFINITIONS:
0059  1 !
0060  1
0061  1 REQUIRE 'RTLIN:PASPROLOG';                         . Linkages, externals, PSECTs, structures
0125  1
0126  1 !
0127  1 ! TABLE OF CONTENTS:
0128  1 !
0129  1
0130  1 FORWARD ROUTINE
0131  1     PAS$$ADD_FILE: NOVALUE,                         ! Add file to global list
0132  1     PAS$$REMOVE_FILE: NOVALUE,                      ! Remove file from global list
0133  1     PAS$$PROMPT_ALL: NOVALUE,                       ! Prompt on all enabled files
0134  1     PAS$$PROMPT_FILE: JSB_PROMPT_FILE NOVALUE,      ! Prompt on a file
0135  1     PAS$$CLOSE_ALL: NOVALUE,                        ! Close all files
0136  1     PAS$CLOSE_LOCAL_R3: JSB_CLOSE_LOCAL NOVALUE,    ! Close all local files
0137  1     PAS$$CLOSE_LOCAL: JSB_CLOSE_LOCAL NOVALUE,      ! Internally callable
0138  1     DO_CLOSE: NOVALUE,                              ! Close a file
0139  1     DO_CLOSE_HANDLER,                               ! Handler for DO_CLOSE
0140  1     INITIALIZE_QUEUE: NOVALUE,                      ! Initialize FILE_QUEUE
0141  1     SERVICE_REQUEST: NOVALUE;                       ! Service remove request
0142  1
0143  1 !
0144  1 ! MACROS:
0145  1 !
0146  1 !         NONE
0147  1 !
0148  1 ! EQUATED SYMBOLS:
0149  1 !
0150  1 !         NONE
0151  1 !
0152  1 ! FIELDS:
0153  1 !
0154  1 !         NONE
0155  1 !
0156  1 ! OWN STORAGE:
0157  1 !
0158  1
0159  1 OWN
0160  1     FILE_QUEUE: VECTOR [2, LONG],                           ! Queue of FCBs
0161  1     REQUEST_LEVEL: INITIAL (-1),                            ! Reentrancy level
0162  1     QUEUE_INITIALIZED: INITIAL (0),                         ! True if queue initialized
0163  1     REMOVE_REQUESTED: INITIAL (0);                          ! Remove requested from AST level
```

```
;  103          0164  1 %SBTTL 'PAS$$ADD_FILE - Add file to queue'
;  104          0165  1 GLOBAL ROUTINE PAS$$ADD_FILE (
;  105          0166  1     FCB: REF $PAS$FCB_CONTROL_BLOCK
;  106          0167  1     ): NOVALUE =
;  107          0168  1
;  108          0169  1 !++
;  109          0170  1 ! FUNCTIONAL DESCRIPTION:
;  110          0171  1 !
;  111          0172  1 !       Adds a file's FCB to the queue of files.
;  112          0173  1 !
;  113          0174  1 ! CALLING SEQUENCE:
;  114          0175  1 !
;  115          0176  1 !       PAS$$ADD_FILE (FCB.r.r)
;  116          0177  1 !
;  117          0178  1 ! FORMAL PARAMETERS:
;  118          0179  1 !
;  119          0180  1 !       FCB             File Control Block for file
;  120          0181  1 !
;  121          0182  1 ! IMPLICIT INPUTS:
;  122          0183  1 !
;  123          0184  1 !       FILE_QUEUE
;  124          0185  1 !       REQUEST_LEVEL
;  125          0186  1 !       QUEUE_INITIALIZED
;  126          0187  1 !       REMOVE_REQUESTED
;  127          0188  1 !
;  128          0189  1 ! IMPLICIT OUTPUTS:
;  129          0190  1 !
;  130          0191  1 !       NONE
;  131          0192  1 !
;  132          0193  1 ! COMPLETION STATUS:
;  133          0194  1 !
;  134          0195  1 !       NONE
;  135          0196  1 !
;  136          0197  1 ! SIDE EFFECTS:
;  137          0198  1 !
;  138          0199  1 !       Inserts FCB onto head of FILE_QUEUE.
;  139          0200  1 !
;  140          0201  1 ! SIGNALLED ERRORS:
;  141          0202  1 !
;  142          0203  1 !       NONE
;  143          0204  1 !--
;  144          0205  1
;  145          0206  2     BEGIN
;  146          0207  2
;  147          0208  2     BUILTIN
;  148          0209  2         INSQUE;
;  149          0210  2
;  150          0211  2     !+
;  151          0212  2     ! Initialize the queue if necessary.
;  152          0213  2     !-
;  153          0214  2
;  154          0215  2     IF NOT .QUEUE_INITIALIZED
;  155          0216  2     THEN
;  156          0217  2         INITIALIZE_QUEUE ();
;  157          0218  2
;  158          0219  2     !+
;  159          0220  2     ! Increment REQUEST_LEVEL.
;  2      !-
;  2
;  2      REQUEST_LEVEL = .REQUEST_LEVEL + 1;
;  2
;  2      !+
;  2      ! Insert FCB onto FILE_QUEUE at head.
;  2      !-
;  2
;  2      INSQUE (FCB [FCB$L_QUEUE_FLINK], FILE_QUEUE);
;  2
;  2      !+
;  2      ! Mark the FCB as being on the queue.
;  2      !-
;  2
;  2      FCB [FCB$V_ON_QUEUE] = 1;
;  2
;  2      !+
;  2      ! Decrement REQUEST_LEVEL.
;  2      !-
;  2
;  2      REQUEST_LEVEL = .REQUEST_LEVEL - 1;
;  2
;  2      !+
;  2      ! If a remove request has been made, service it.
;  2      !-
;  2
;  2      IF .REMOVE_REQUESTED
;  2      THEN
;  2          SERVICE_REQUEST ();
;  2
;  2      RETURN;
;  2
;  1      END;                                      ! End of routine PAS$$ADD_FILE

                                                  .TITLE  PAS$$FILE_UTIL File manipulation utility proced
;                                                                        ures
                                                  .IDENT  \1-005\

                                                  .PSECT  _PAS$DATA,NOEXE,  PIC,2

                                  00000 FILE_QUEUE:
                                                  .BLKB   8
                        FFFFFFFF  00008 REQUEST_LEVEL:
                                                  .LONG   -1                                      ;
                        00000000  0000C QUEUE_INITIALIZED:
                                                  .LONG   0                                       ;
                        00000000  00010 REMOVE_REQUESTED:
                                                  .LONG   0                                       ;

                                                  .EXTRN  PAS$$ADD_FILE, PAS$$REMOVE_FILE
                                                  .EXTRN  PAS$$PROMPT_ALL
                                                  .EXTRN  PAS$$PROMPT_FILE
                                                  .EXTRN  PAS$$CLOSE_ALL, PAS$CLOSE_LOCAL_R3
                                                  .EXTRN  PAS$$CLOSE_LOCAL

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

```
                                     0004 00000          .ENTRY  PAS$$ADD_FILE, Save R2                   ; 0165
                  52 00000000'  EF   9E 00002            MOVAB   REQUEST_LEVEL, R2                        ;
                  05        04  A2   E8 00009            BLBS    QUEUE_INITIALIZED, 1$                    ; 0215
            0000V CF            00   FB 0000D            CALLS   #0, INITIALIZE_QUEUE                     ; 0217
                                62   D6 00012 1$:        INCL    REQUEST_LEVEL                            ; 0223
                  50        04  AC   D0 00014            MOVL    FCB, R0                                  ; 0229
             F8   A2        BC  A0   0E 00018            INSQUE  -68(R0), FILE_QUEUE                      ;
                  50        04  AC   D0 0001D            MOVL    FCB, R0                                  ; 0235
             FE   A0            20   88 00021            BISB2   #32, -2(R0)                              ;
                                62   D7 00025            DECL    REQUEST_LEVEL                            ; 0241
                  05        08  A2   E9 00027            BLBC    REMOVE_REQUESTED, 2$                     ; 0247
            0000V CF            00   FB 0002B            CALLS   #0, SERVICE_REQUEST                      ; 0249
                                     04 00030 2$:        RET                                              ; 0253

; Routine Size:  49 bytes,    Routine Base:  _PAS$CODE + 0000


;  193          0254  1
;  194          0255  1 !<BLF/PAGE>
```

```
1 %SBTTL 'PAS$$REMOVE_FILE - Remove file from queue'
1 GLOBAL ROUTINE PAS$$REMOVE_FILE (
1     FCB: REF $PAS$FCB_CONTROL_BLOCK
1     ): NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Remove a file's FCB from the queue of files.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       PAS$$REMOVE_FILE (FCB.r.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       FCB             File Control Block for file
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       FILE_QUEUE
1 !       REQUEST_LEVEL
1 !       QUEUE_INITIALIZED
1 !       REMOVE_REQUESTED
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       FILE_QUEUE
1 !       REQUEST_LEVEL
1 !       QUEUE_INITIALIZED
1 !       REMOVE_REQUESTED
1 !       FCB [FCB$V_DEALLOC]
1 !
1 ! COMPLETION STATUS:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Removes FCB from FILE_QUEUE or requests deallocation.
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       NONE
1 !--
1
2     BEGIN
2
2     BUILTIN
2         REMQUE;
2
2     !+
2     ! Initialize the queue if necessary.
2     !-
2
2     IF NOT .QUEUE_INITIALIZED
2     THEN
```

```
2           INITIALIZE_QUEUE ();
2
2       !+
2       ! Invalidate FCB pointer in PFV.
2       !-
2
3       BEGIN
3       LOCAL
3           PFV: REF $PAS$PFV_FILE_VARIABLE;
3       PFV = .FCB [FCB$A_PFV];
3       PFV [PFV$V_FCB_VALID] = 0;
3       PFV [PFV$L_STATUS] = .FCB [FCB$L_STATUS];   ! Overlays PFV$A_FCB
2       END;
2
2
2       !+
2       ! If the FCB is not on the queue then simply free the
2       ! storage and return.
2       !-
2
2       IF NOT .FCB [FCB$V_ON_QUEUE]
2       THEN
3           BEGIN
3           LOCAL
3               BLOCK_ADDR; ! Address of allocated block
3           BLOCK_ADDR = FCB [FCB$L_QUEUE_FLINK];
3           PAS$$FREE_VM (PAS$K_FILE_DYN_BLN, BLOCK_ADDR);
3           END
3
2       ELSE
2
3           BEGIN
3           !+
3           ! Increment REQUEST_LEVEL.  If we are at level zero, then we can do the
3           ! REMQUE directly, so do it and free the storage.
3           ! Otherwise set the DEALLOC bit in the FCB and set REMOVE_REQUESTED.
3           !-
3
3           IF (REQUEST_LEVEL=.REQUEST_LEVEL+1) EQL 0
3           THEN
4               BEGIN
4               LOCAL
4                   ITEM_ADDR;                      ! Output from REMQUE
4               REMQUE (FCB [FCB$L_QUEUE_FLINK], ITEM_ADDR);
4               FCB [FCB$V_ON_QUEUE] = 0;
4               PAS$$FREE_VM (PAS$K_FILE_DYN_BLN, ITEM_ADDR);
4               END
3           ELSE
4               BEGIN
4               FCB [FCB$V_DEALLOC] = 1;
4               REMOVE_REQUESTED = 1;
3               END;
3
3           !+
3           ! Decrement REQUEST_LEVEL.
3           !-
3
```

```
;  310          0370  3           REQUEST_LEVEL = .REQUEST_LEVEL - 1;
;  311          0371  3
;  312          0372  2           END;
;  313          0373  2
;  314          0374  2       !+
;  315          0375  2       ! If a remove request has been made, service it.
;  316          0376  2       !-
;  317          0377  2
;  318          0378  2       IF .REMOVE_REQUESTED
;  319          0379  2       THEN
;  320          0380  2           SERVICE_REQUEST ();
;  321          0381  2
;  322          0382  2       RETURN;
;  323          0383  2
;  324          0384  1       END;                                     ! End of routine PAS$$REMOVE_FILE


                                                                      .EXTRN  PAS$$FREE_VM

                                             001C 00000               .ENTRY  PAS$$REMOVE_FILE, Save R2,R3,R4      ; 0257
                        54 00000000G   00   9E 00002                  MOVAB   PAS$$FREE_VM, R4                     ;
                        53 00000000'   EF   9E 00009                  MOVAB   REQUEST_LEVEL, R3                    ;
                        5E             08   C2 00010                  SUBL2   #8, SP                               ;
                        05       04    A3   E8 00013                  BLBS    QUEUE_INITIALIZED, 1$                ; 0311
                 0000V  CF             00   FB 00017                  CALLS   #0, INITIALIZE_QUEUE                 ; 0313
                        52       04    AC   D0 0001C 1$:              MOVL    FCB, R2                              ; 0322
                        50       DC    A2   D0 00020                  MOVL    -36(R2), PFV                         ;
                   07   A0       40    8F   8A 00024                  BICB2   #64, 7(PFV)                          ; 0323
                   0C   A0       D4    A2   D0 00029                  MOVL    -44(R2), 12(PFV)                     ; 0324
          10       FE   A2             05   E0 0002E                  BBS     #5, -2(R2), 2$                       ; 0333
                        6E       BC    A2   9E 00033                  MOVAB   -68(R2), BLOCK_ADDR                  ; 0338
                                       5E   DD 00037                  PUSHL   SP                                   ; 0339
                        7E     0138    8F   3C 00039                  MOVZWL  #312, -(SP)                          ;
                        64             02   FB 0003E                  CALLS   #2, PAS$$FREE_VM                     ;
                                       2B   11 00041                  BRB     5$                                   ; 0333
                                       63   D6 00043 2$:              INCL    REQUEST_LEVEL                        ; 0351
                                       1D   12 00045                  BNEQ    3$                                   ;
                        52       BD    A2   9E 00047                  MOVAB   -67(R2), R2                          ; 0356
                   04   AE             72   0F 0004B                  REMQUE  -(R2), ITEM_ADDR                     ;
                        50       04    AC   D0 0004F                  MOVL    FCB, R0                              ; 0357
                   FE   A0             20   8A 00053                  BICB2   #32, -2(R0)                          ;
                                 04    AE   9F 00057                  PUSHAB  ITEM_ADDR                            ; 0358
                        7E     0138    8F   3C 0005A                  MOVZWL  #312, -(SP)                          ;
                        64             02   FB 0005F                  CALLS   #2, PAS$$FREE_VM                     ;
                                       08   11 00062                  BRB     4$                                   ; 0351
                   FE   A2             02   88 00064 3$:              BISB2   #2, -2(R2)                           ; 0362
                   08   A3             01   D0 00068                  MOVL    #1, REMOVE_REQUESTED                 ; 0363
                                       63   D7 0006C 4$:              DECL    REQUEST_LEVEL                        ; 0370
                        05       08    A3   E9 0006E 5$:              BLBC    REMOVE_REQUESTED, 6$                 ; 0378
                 0000V  CF             00   FB 00072                  CALLS   #0, SERVICE_REQUEST                  ; 0380
                                            04 00077 6$:              RET                                          ; 0384

; Routine Size:  120 bytes,    Routine Base:  _PAS$CODE + 0031


;  325          0385  1
;  326          0386  1 !<BLF/PAGE>
```

```
1 %SBTTL 'PAS$$PROMPT_ALL - Prompt on all prompt-enabled files'
1 GLOBAL ROUTINE PAS$$PROMPT_ALL
1     : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Finds all files for which prompting is enabled and which have
1 !       partial lines and writes the partial lines.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       PAS$$PROMPT_ALL ()
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       FILE_QUEUE
1 !       REQUEST_LEVEL
1 !       QUEUE_INITIALIZED
1 !       REMOVE_REQUESTED
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         FCB: REF $PAS$FCB_CONTROL_BLOCK;         ! File control block
2
2     BIND
2         RAB = FCB: REF BLOCK [, BYTE];           ! RAB is FCB
2
2     BUILTIN
2         TESTBITCS;
2
2     !+
2     ! If queue is not initialize, bugcheck.
2     !-
2
2     IF NOT .QUEUE_INITIALIZED
2     THEN
```

```
2           $PAS$BUGCHECK (BUG_FQNOTINIT);
2
2       !+
2       ! Increment REQUEST_LEVEL.
2       !-
2
2       REQUEST_LEVEL = .REQUEST_LEVEL + 1;
2
2       !+
2       ! Get the first file from the queue.
2       !-
2
2       FCB = .FILE_QUEUE [0];
2
2       !+
2       ! While there are files left, look for files to prompt on.
2       !-
2
2       WHILE (FCB [FCB$R_FCB] NEQA FILE_QUEUE) DO  ! Stop when we get back to header
3           BEGIN
3           FCB = FCB [FCB$R_FCB] + FCB$K_BLN;      ! Set correct FCB origin
3           IF .FCB [FCB$V_PROMPT_ENABLE] AND
3               .FCB [FCB$V_GENERATION] AND
3               NOT .FCB [FCB$V_DEALLOC]
3           THEN
4               BEGIN
4               LOCAL
4                   PFV: REF $PAS$PFV_FILE_VARIABLE;        ! Pascal File Variable
4               PFV = .FCB [FCB$A_PFV];      ! Get file variable
4               IF PFV [PFV$R_PFV] NEQA 0
4               THEN
4                   IF TESTBITCS (PFV [PFV$V_LOCK]) ! Test and set file lock
4                   THEN
5                       BEGIN
5                       !+
5                       ! File is locked.  See if it is in Generation mode
5                       ! and has a partial line in the buffer.  If so, call
5                       ! PAS$$PROMPT_FILE to output the prompt.
5                       !-
5
5                       IF .FCB [FCB$V_GENERATION] AND
6                           (.FCB [FCB$A_RECORD_CUR] NEQA .FCB [FCB$A_RECORD_BEG])
5                       THEN
5                           PAS$$PROMPT_FILE (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
5
5                       !+
5                       ! Unlock file.
5                       !-
5
5                       PFV [PFV$V_LOCK] = 0;
4                       END;
4
3               END;
3
3           !+
3           ! Get next file from queue.
3           !-
```

```
;    442          0501  3
;    443          0502  3              FCB = .FCB [FCB$L_QUEUE_FLINK];
;    444          0503  3              END;
;    445          0504  2
;    446          0505  2          !+
;    447          0506  2          ! Decrement REQUEST_LEVEL.
;    448          0507  2          !-
;    449          0508  2
;    450          0509  2          REQUEST_LEVEL = .REQUEST_LEVEL - 1;
;    451          0510  2
;    452          0511  2          !+
;    453          0512  2          ! If a remove request has been made, service it.
;    454          0513  2          !-
;    455          0514  2
;    456          0515  2          IF .REMOVE_REQUESTED
;    457          0516  2          THEN
;    458          0517  2              SERVICE_REQUEST ();
;    459          0518  2
;    460          0519  2          RETURN;
;    461          0520  2
;    462          0521  1          END;                                    ! End of routine PAS$$PROMPT_ALL


                                                                 .EXTRN  PAS$$BUGCHECK

                                             00C4 00000          .ENTRY  PAS$$PROMPT_ALL, Save R2,R6,R7                 ; 0388
                    52 00000000'   EF   9E 00002          MOVAB   REQUEST_LEVEL, R2                                      ;
                    0A        04   A2   E8 00009          BLBS    QUEUE_INITIALIZED, 1$                                  ; 0442
                                   01   DD 0000D          PUSHL   #1                                                     ; 0444
              00000000G  00        01   FB 0000F          CALLS   #1, PAS$$BUGCHECK                                      ;
                                        04 00016          RET                                                            ;
                                   62   D6 00017  1$:     INCL    REQUEST_LEVEL                                          ; 0450
                    57        F8   A2   D0 00019          MOVL    FILE_QUEUE, FCB                                        ; 0456
                    50        F8   A2   9E 0001D  2$:     MOVAB   FILE_QUEUE, R0                                         ; 0462
                    50             57   D1 00021          CMPL    FCB, R0                                                ;
                                   38   13 00024          BEQL    5$                                                     ;
                    57        44   A7   9E 00026          MOVAB   68(R7), FCB                                            ; 0464
        29       FD   A7               06   E1 0002A          BBC     #6, -3(FCB), 4$                                        ; 0465
        24       FD   A7               04   E1 0002F          BBC     #4, -3(FCB), 4$                                        ; 0466
        1F       FE   A7               01   E0 00034          BBS     #1, -2(FCB), 4$                                        ; 0467
                    56        DC   A7   D0 00039          MOVL    -36(FCB), PFV                                          ; 0472
                                   19   13 0003D          BEQL    4$                                                     ; 0473
        14       04   A6               1F   E2 0003F          BBSS    #31, 4(PFV), 4$                                        ; 0475
        0A       FD   A7               04   E1 00044          BBC     #4, -3(FCB), 3$                                        ; 0484
                 E8   A7          EC   A7   D1 00049          CMPL    -20(FCB), -24(FCB)                                     ; 0485
                                   03   13 0004E          BEQL    3$                                                     ;
                                0000V   30 00050          BSBW    PAS$$PROMPT_FILE                                       ; 0487
                 07   A6          80   8F   8A 00053  3$:     BICB2   #128, 7(PFV)                                           ; 0493
                    57        BC   A7   D0 00058  4$:     MOVL    -68(FCB), FCB                                          ; 0502
                                   BF   11 0005C          BRB     2$                                                     ; 0462
                                   62   D7 0005E  5$:     DECL    REQUEST_LEVEL                                          ; 0509
                    05        08   A2   E9 00060          BLBC    REMOVE_REQUESTED, 6$                                   ; 0515
               0000V  CF           00   FB 00064          CALLS   #0, SERVICE_REQUEST                                    ; 0517
                                        04 00069  6$:     RET                                                            ; 0521

; Routine Size:  106 bytes,    Routine Base:  _PAS$CODE + 00A9
;  463          0522  1
;  464          0523  1 !<BLF/PAGE>
```

```
1 %SBTTL 'PAS$$PROMPT_FILE - Prompt on a prompt-enabled files'
1 GLOBAL ROUTINE PAS$$PROMPT_FILE (
1     PFV: REF $PAS$PFV_FILE_VARIABLE,          ! Pascal File Variable
1     FCB: REF $PAS$FCB_CONTROL_BLOCK           ! File Control Block
1     ) : JSB_PROMPT_FILE NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Performs a partial-line write on a prompt-enabled file.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       PAS$$PROMPT_FILE (PFV.mr.r, FCB.mr.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       PFV             - The Pascal File Variable for the file.
1 !
1 !       FCB             - The File Control Block for the file.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       It is assumed that the file is a prompt-enabled textfile which
1 !       is locked and in Generation mode.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       A partial line is written to the file, with the cursor left at
1 !       the end of the text written.
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       ERRDURPRO - error during prompting
1 !--
1
2     BEGIN
2
2     LOCAL
2         CHARS_IN_LINE;                        ! Number of characters in the line
2
2     BIND
2         RAB = FCB: REF BLOCK [, BYTE];        ! RAB is FCB
2
2     !+
2     ! If the record has any characters in it, write the partial line.
2     !-
2
2     CHARS_IN_LINE = .FCB [FCB$A_RECORD_CUR] - .FCB [FCB$A_RECORD_BEG];
```

```
;   523        0581  2        IF .CHARS_IN_LINE NEQ 0
;   524        0582  2        THEN
;   525        0583  3            BEGIN
;   526        0584  3            !+
;   527        0585  3            ! Set up record pointer in RAB for $PUT.
;   528        0586  3            !-
;   529        0587  3
;   530        0588  3            RAB [RAB$L_RBF] = .FCB [FCB$A_RECORD_BEG];
;   531        0589  3            RAB [RAB$W_RSZ] = .CHARS_IN_LINE;
;   532        0590  3
;   533        0591  3            !+
;   534        0592  3            ! Set carriagecontrol depending on whether a partial
;   535        0593  3            ! line has been previously written.
;   536        0594  3            !-
;   537        0595  3
;   538        0596  3            IF .FCB [FCB$V_PARTIAL_LINE]
;   539        0597  3            THEN
;   540        0598  3                FCB [FCB$W_PROMPT_CC] = FCB$K_CC_NULL       ! Nothing before, nothing
;   541        0599  3            ELSE
;   542        0600  3                FCB [FCB$W_PROMPT_CC] = FCB$K_CC_LFNL;      ! LF before, nothing after
;   543        0601  3
;   544        0602  3            !+
;   545        0603  3            ! Do the $PUT and check for errors.
;   546        0604  3            !-
;   547        0605  3
;   548        0606  4            IF NOT $PAS$RMS_OP ($PUT (RAB=.RAB))
;   549        0607  3            THEN
;   550        0608  3                $PAS$IO_ERROR (PAS$_ERRDURPRO);
;   551        0609  3
;   552        0610  3            !+
;   553        0611  3            ! Reset the record buffer.
;   554        0612  3            !-
;   555        0613  3
;   556        0614  3            FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_BEG];
;   557        0615  3            FCB [FCB$V_PARTIAL_LINE] = 1;
;   558        0616  3
;   559        0617  2            END;
;   560        0618  2
;   561        0619  2        RETURN;
;   562        0620  2
;   563        0621  1        END;                                     ! End of routine PAS$$PROMPT_FILE


                                                        .EXTRN  SYS$PUT, PAS$$SIGNAL
                                                        .EXTRN  PAS$K_ERRDURPRO

                50      EC   A7      E8   A7   C3 00000 PAS$$PROMPT_FILE::
                                                        SUBL3   -24(FCB), -20(FCB), CHARS_IN_LINE          ; 0580
                                          49   13 00006         BEQL    5$                                         ; 0581
                        28   A7      E8   A7   D0 00008         MOVL    -24(FCB), 40(FCB)                          ; 0588
                        22   A7           50   B0 0000D         MOVW    CHARS_IN_LINE, 34(FCB)                     ; 0589
                                     FD   A7   95 00011         TSTB    -3(FCB)                                    ; 0596
                                          05   18 00014         BGEQ    1$                                         ;
                                     FA   A7   B4 00016         CLRW    -6(FCB)                                    ; 0598
                                          04   11 00019         BRB     2$                                         ;
                        FA   A7           01   B0 0001B 1$:     MOVW    #1, -6(FCB)                                ; 0600
```

```
                                            57  DD 0001F 2$:        PUSHL   FCB                                       ; 0606
                        00000000G  00           01  FB 00021            CALLS   #1, SYS$PUT                               ;
                                   1C           50  E8 00028            BLBS    $$STATUS, 4$                              ;
                        0001825A   8F           50  D1 0002B            CMPL    $$STATUS, #98906                          ;
                                                04  12 00032            BNEQ    3$                                        ;
                                   E7     FF    A7  E8 00034            BLBS    -1(FCB), 2$                               ;
                                   0C           50  E8 00038 3$:        BLBS    $$STATUS, 4$                              ;
                                   7E    00G    8F  9A 0003B            MOVZBL  #PAS$K_ERRDURPRO, -(SP)                   ; 0608
                        00000000G  00           01  FB 0003F            CALLS   #1, PAS$$SIGNAL                           ;
                                                05  00046               RSB                                               ;
                              EC   A7     E8    A7  D0 00047 4$:        MOVL    -24(FCB), -20(FCB)                        ; 0614
                              FD   A7     80    8F  88 0004C            BISB2   #128, -3(FCB)                             ; 0615
                                                05  00051 5$:           RSB                                               ; 0621

; Routine Size:  82 bytes,    Routine Base:  _PAS$CODE + 0113


;  564          0622  1
;  565          0623  1 !<BLF/PAGE>
```

```
1 %SBTTL 'PAS$$CLOSE_ALL - Close all open files'
1 GLOBAL ROUTINE PAS$$CLOSE_ALL
1     : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Closes all open files.  This procedure is called from the exit
1 !       handler declared by PAS$$OPEN.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       PAS$$CLOSE_ALL ()
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       FILE_QUEUE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Closes all open files, and removes their control blocks
1 !       from the queue.
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       NONE
1 !--
1
2     BEGIN
2
2     LOCAL
2         FCB: REF $PAS$FCB_CONTROL_BLOCK,        ! File control block
2         DUMMY_PFV: $PAS$PFV_FILE_VARIABLE,      ! Dummy PFV for local use
2         AST_STATUS;                             ! Status from $SETAST
2
2     BUILTIN
2         REMQUE;
2
2     !+
2     ! If queue not initialized, nothing to close.
2     !-
2
2     IF NOT .QUEUE_INITIALIZED
2     THEN
2         RETURN;
```

```
;  2
;  2      !+
;  2      ! Set up dummy PFV.  We will use this to close files since
;  2      ! the true PFV may be invalid.
;  2      !-
;  2
;  2      DUMMY_PFV [PFV$W_FLAGS] = 0;
;  2
;  2      !+
;  2      ! Disable ASTs
;  2      !-
;  2
;  2      AST_STATUS = $SETAST (ENBFLG=0);
;  2
;  2      !+
;  2      ! Remove all files from the queue, and close those still open.
;  2      !-
;  2
;  2      UNTIL (REMQUE (.FILE_QUEUE [0], FCB)) DO    ! True when REMQUE fails
;  3          BEGIN
;  3          FCB = FCB [FCB$R_FCB] + FCB$K_BLN;                ! Get correct FCB origin
;  3          IF NOT .FCB [FCB$V_DEALLOC]
;  3          THEN
;  4              BEGIN
;  4              !+
;  4              ! Use dummy PFV to do the close.
;  4              !-
;  4              DUMMY_PFV [PFV$A_FCB] = FCB [FCB$R_FCB];
;  4              DUMMY_PFV [PFV$A_PFD] = .FCB [FCB$A_PFD];
;  4              DUMMY_PFV [PFV$V_FCB_VALID] = 1;
;  4              DO_CLOSE (DUMMY_PFV [PFV$R_PFV]);   ! Close the file
;  3              END;
;  2          END;
;  2
;  2      !+
;  2      ! If ASTs were previously enabled, reenable them.
;  2      !-
;  2
;  2      IF .AST_STATUS EQL SS$_WASSET
;  2      THEN
;  2          $SETAST (ENBFLG = 1);
;  2
;  2      RETURN;
;  2
;  1      END;                                        ! End of routine PAS$$CLOSE_ALL


                                               .EXTRN  SYS$SETAST

                       001C 00000              .ENTRY  PAS$$CLOSE_ALL, Save R2,R3,R4          ; 0625
      54 00000000G  00   9E 00002              MOVAB   SYS$SETAST, R4                          ;
      5E            10   C2 00009              SUBL2   #16, SP                                 ;
      3E 00000000'  EF   E9 0000C              BLBC    QUEUE_INITIALIZED, 3$                   ; 0678
              06    AE   B4 00013              CLRW    DUMMY_PFV+6                             ; 0687
                    7E   D4 00016              CLRL    -(SP)                                   ; 0693
      64            01   FB 00018              CALLS   #1, SYS$SETAST                          ;
```

```
                                   53             50  D0 0001B              MOVL     R0, AST_STATUS                            ;
                                   52 00000000'   FF  0F 0001E 1$:          REMQUE   @FILE_QUEUE, FCB                          ; 0699
                                   20             1D 00025                  BVS      2$                                        ;
                                   52        44   A2  9E 00027              MOVAB    68(R2), FCB                               ; 0701
              EE         FE        A2             01  E0 0002B              BBS      #1, -2(FCB), 1$                           ; 0702
                         0C        AE             52  D0 00030              MOVL     FCB, DUMMY_PFV+12                         ; 0708
                         08        AE        E4   A2  D0 00034              MOVL     -28(FCB), DUMMY_PFV+8                     ; 0709
                         07        AE        40   8F  88 00039              BISB2    #64, DUMMY_PFV+7                          ; 0710
                                                  5E  DD 0003E              PUSHL    SP                                        ; 0711
                      0000V  CF                   01  FB 00040              CALLS    #1, DO_CLOSE                              ;
                                                  D7  11 00045              BRB      1$                                        ; 0699
                                   09             53  D1 00047 2$:          CMPL     AST_STATUS, #9                            ; 0719
                                                  05  12 0004A              BNEQ     3$                                        ;
                                                  01  DD 0004C              PUSHL    #1                                        ; 0721
                                   64             01  FB 0004E              CALLS    #1, SYS$SETAST                            ;
                                                      04 00051 3$:          RET                                                ; 0725

; Routine Size:  82 bytes,    Routine Base:  _PAS$CODE + 0165


;  669          0726  1
;  670          0727  1 !<BLF/PAGE>
```

```
1 %SBTTL 'PAS$CLOSE_LOCAL_R3 - Close local files'
1 GLOBAL ROUTINE PAS$CLOSE_LOCAL_R3
1      : JSB_CLOSE_LOCAL NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Closes all open files which were declared local by our caller.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       JSB PAS$CLOSE_LOCAL_R3
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       Current FP (Caller's)
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Preserves registers 0 and 1.
1 !       See PAS$$CLOSE_LOCAL
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       NONE
1 !--
1
2    BEGIN
2
2    BUILTIN
2        FP;
2
2    !+
2    ! Call PAS$$CLOSE_LOCAL with one argument, the current FP.  This
2    ! will get pushed on the stack.
2    !-
2
2    PAS$$CLOSE_LOCAL (.FP);
2
2    RETURN;
2
1    END;                                        ! End of routine PAS$$CLOSE_LOCAL_R3
```

```
                                        5D  DD 00000 PAS$CLOSE_LOCAL_R3::
                                                              PUSHL    FP                                   ; 0777
                                     0000V  30 00002          BSBW     PAS$$CLOSE_LOCAL                     ;
                         5E             04  C0 00005          ADDL2    #4, SP                               ;
                                            05 00008          RSB                                           ; 0781

; Routine Size:  9 bytes,    Routine Base:  _PAS$CODE + 01B7


;  726          0782  1
;  727          0783  1 !<BLF/PAGE>
```

```
1 %SBTTL 'PAS$$CLOSE_LOCAL - Close local files'
1 GLOBAL ROUTINE PAS$$CLOSE_LOCAL (PROCEDURE_FP)
1       : JSB_CLOSE_LOCAL NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Closes all open files which were declared local by our caller.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       JSB PAS$$CLOSE_LOCAL (PROCEDURE_FP.rlu.v)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       PROCEDURE_FP - This is the frame pointer of the procedure for
1 !                      which we are closing its local files.  This
1 !                      value is passed on the stack.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       Our SP
1 !       FILE_QUEUE
1 !       REQUEST_LEVEL
1 !       QUEUE_INITIALIZED
1 !       REMOVE_REQUESTED
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Closes all open files whose PFVs are between PROCEDURE_FP and SP
1 !       (i.e. declared locally in our caller's procedure).
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       NONE
1 !--
1
2    BEGIN
2
2    LOCAL
2       FCB: REF $PAS$FCB_CONTROL_BLOCK,        ! File control block
2       NEXT_FCB,                               ! Next FCB in QUEUE
2       REMQUE_OK;                              ! TRUE if ok to do REMQUEs
2
2    BUILTIN
2       REMQUE,
2       SP;
2
2    !+
2       ! If queue not initialized, nothing to close.
2       !-
2
2       IF NOT .QUEUE_INITIALIZED
2       THEN
2           RETURN;
2
2       !+
2       ! Increment REQUEST_LEVEL and set REMQUE_OK appropriately.
2       !-
2
2       IF (REQUEST_LEVEL = .REQUEST_LEVEL + 1) NEQ 0
2       THEN
2           REMQUE_OK = 0
2       ELSE
2           REMQUE_OK = 1;
2
2       !+
2       ! Get the first FCB from the queue.
2       !-
2
2       FCB = .FILE_QUEUE [0];                  ! Forward link
2
2       !+
2       ! While there are files left, look for local files to close.
2       !-
2
2       WHILE (FCB [FCB$R_FCB] NEQA FILE_QUEUE) DO  ! Stop when we get back to header
3           BEGIN
3           FCB = FCB [FCB$R_FCB] + FCB$K_BLN;      ! Get correct FCB origin
3           NEXT_FCB = .FCB [FCB$L_QUEUE_FLINK];    ! Next file in queue
3           IF NOT .FCB [FCB$V_DEALLOC] AND NOT .FCB [FCB$V_STATIC]
3           THEN
4               BEGIN
4               LOCAL
4                   PFV: REF $PAS$PFV_FILE_VARIABLE;
4               PFV = .FCB [FCB$A_PFV];     ! Get PFV
4               IF PFV [PFV$R_PFV] LSSA .PROCEDURE_FP AND PFV [PFV$R_PFV] GTRA .SP
4               THEN
5                   BEGIN
5                   !+
5                   ! We have a local file.  We can't be guaranteed that the
5                   ! contents of the PFV are valid, so set the necessary items
5                   ! here.  Close the file.
5                   !-
5
5                   PFV [PFV$W_FLAGS] = 0;
5                   PFV [PFV$V_LOCK] = 1;
5                   PFV [PFV$V_FCB_VALID] = 1;
5                   PFV [PFV$A_FCB] = FCB [FCB$R_FCB];
5                   PFV [PFV$A_PFD] = .FCB [FCB$A_PFD];
5                   DO_CLOSE (PFV [PFV$R_PFV]);
5
5                   !+
5                   ! Remove the file from the queue.  This will either be
5                   ! a REMQUE or a request to remove.
5                   !-
```

```
;  843         0898  5
;  844         0899  5                  IF .REMQUE_OK
;  845         0900  5                  THEN
;  846         0901  6                      BEGIN
;  847         0902  6                      LOCAL
;  848         0903  6                          ITEM_ADDR;                  ! Output from REMQUE
;  849         0904  6                      REMQUE (FCB [FCB$L_QUEUE_FLINK], ITEM_ADDR);
;  850         0905  6                      PAS$$FREE_VM (PAS$K_FILE_DYN_BLN, ITEM_ADDR);
;  851         0906  6                      END
;  852         0907  5                  ELSE
;  853         0908  6                      BEGIN
;  854         0909  6                      FCB [FCB$V_DEALLOC] = 1;
;  855         0910  6                      REMOVE_REQUESTED = 1;
;  856         0911  5                      END;
;  857         0912  4                  END;
;  858         0913  3              END;
;  859         0914  3          !+
;  860         0915  3          ! Get next FCB from queue.
;  861         0916  3          !-
;  862         0917  3          FCB = .NEXT_FCB;
;  863         0918  2          END;
;  864         0919  2
;  865         0920  2      !+
;  866         0921  2      ! Decrement REQUEST_LEVEL.
;  867         0922  2      !-
;  868         0923  2
;  869         0924  2      REQUEST_LEVEL = .REQUEST_LEVEL - 1;
;  870         0925  2
;  871         0926  2      !+
;  872         0927  2      ! If a remove request has been made, service it.
;  873         0928  2      !-
;  874         0929  2
;  875         0930  2      IF .REMOVE_REQUESTED
;  876         0931  2      THEN
;  877         0932  2          SERVICE_REQUEST ();
;  878         0933  2
;  879         0934  2      RETURN;
;  880         0935  2
;  881         0936  1      END;                                    ! End of routine PAS$$CLOSE_LOCAL



                                   03  BB 00000 PAS$$CLOSE_LOCAL::
                                                        PUSHR   #^M<R0,R1>                                   ; 0785
                5E                 08  C2 00002         SUBL2   #8, SP
                03 00000000'       EF  E8 00005         BLBS    QUEUE_INITIALIZED, 1$                        ; 0844
                             0099  31 0000C             BRW     8$                                           ;
                   00000000'       EF  D6 0000F 1$:     INCL    REQUEST_LEVEL                                ; 0852
                                   04  13 00015         BEQL    2$                                           ;
                                   6E  D4 00017         CLRL    REMQUE_OK                                    ; 0854
                                   03  11 00019         BRB     3$                                           ;
                6E                 01  D0 0001B 2$:     MOVL    #1, REMQUE_OK                                ; 0856
                52 00000000'       EF  D0 0001E 3$:     MOVL    FILE_QUEUE, FCB                              ; 0862
                50 00000000'       EF  9E 00025 4$:     MOVAB   FILE_QUEUE, R0                               ; 0868
                50                 52  D1 0002C         CMPL    FCB, R0                                      ;
```

```
                                         65  13 0002F          BEQL     7$                                        ;
                       52         44     A2  9E 00031          MOVAB    68(R2), FCB                               ; 0870
                       53         BC     A2  D0 00035          MOVL     -68(FCB), NEXT_FCB                        ; 0871
        53       FE    A2                01  E0 00039          BBS      #1, -2(FCB), 6$                           ; 0872
        4E       F8    A2                06  E0 0003E          BBS      #6, -8(FCB), 6$                           ;
                       50         DC     A2  D0 00043          MOVL     -36(FCB), PFV                             ; 0877
                 14    AE                50  D1 00047          CMPL     PFV, PROCEDURE_FP                         ; 0878
                                         44  1E 0004B          BGEQU    6$                                        ;
                       5E                50  D1 0004D          CMPL     PFV, SP                                   ;
                                         3F  1B 00050          BLEQU    6$                                        ;
                                  06     A0  B4 00052          CLRW     6(PFV)                                    ; 0887
                 07    A0         C0     8F  88 00055          BISB2    #192, 7(PFV)                              ; 0889
                 0C    A0                52  D0 0005A          MOVL     FCB, 12(PFV)                              ; 0890
                 08    A0         E4     A2  D0 0005E          MOVL     -28(FCB), 8(PFV)                          ; 0891
                                         50  DD 00063          PUSHL    PFV                                       ; 0892
              0000V    CF                01  FB 00065          CALLS    #1, DO_CLOSE                              ;
                       19                6E  E9 0006A          BLBC     REMQUE_OK, 5$                             ; 0899
                       50         BC     A2  9E 0006D          MOVAB    -68(FCB), R0                              ; 0904
                 04    AE                60  0F 00071          REMQUE   (R0), ITEM_ADDR                           ;
                                  04     AE  9F 00075          PUSHAB   ITEM_ADDR                                 ; 0905
                       7E       0138     8F  3C 00078          MOVZWL   #312, -(SP)                               ;
          00000000G    00                02  FB 0007D          CALLS    #2, PAS$$FREE_VM                          ;
                                         0B  11 00084          BRB      6$                                        ; 0899
                 FE    A2                02  88 00086 5$:      BISB2    #2, -2(FCB)                               ; 0909
          00000000'    EF                01  D0 0008A          MOVL     #1, REMOVE_REQUESTED                      ; 0910
                       52                53  D0 00091 6$:      MOVL     NEXT_FCB  FCB                             ; 0917
                                         8F  11 00094          BRB      4$                                        ; 0868
                           00000000'     EF  D7 00096 7$:      DECL     REQUEST_LEVEL                             ; 0924
                        05 00000000'     EF  E9 0009C          BLBC     REMOVE_REQUESTED, 8$                      ; 0930
              0000V    CF                00  FB 000A3          CALLS    #0, SERVICE_REQUEST                       ; 0932
                       5E                08  C0 000A8 8$:      ADDL2    #8, SP                                    ; 0936
                                         03  BA 000AB          POPR     #^M<R0,R1>                                ;
                                             05 000AD          RSB                                                ;

; Routine Size:  174 bytes,    Routine Base:  _PAS$CODE + 01C0


;  882          0937  1
;  883          0938  1 !<BLF/PAGE>
```

```
1 %SBTTL 'DO_CLOSE - Close a file'
1 ROUTINE DO_CLOSE (                                  ! Close a file
1       PFV: REF $PAS$PFV_FILE_VARIABLE               ! File variable
1    ): NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine closes a Pascal file.  This entry is called from
1 !       PAS$$CLOSE_ALL and PAS$CLOSE_LOCAL.  It is different from
1 !       PAS$CLOSE2 only in that it does not call PAS$$REMOVE_FILE to
1 !       remove the FCB from the list of open files.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       CALL DO_CLOSE (PFV.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 ! 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 !       See PAS$$CLOSE
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       ERRDURCLO - error during CLOSE
1 !
1 !--
1
2    BEGIN
2
2    LOCAL
2       PFV_ADDR: VOLATILE;                           ! Enable argument
2
2    !+
2    ! Enable a local condition handler to intercept any signals from
2    ! trying to close the file.
2    !-
2
2    ENABLE
2       DO_CLOSE_HANDLER (PFV_ADDR);
2
```

```
;  942         0996  2       !+
;  943         0997  2       ! Lock PFV  We don't care if it is already locked.
;  944         0998  2       !-
;  945         0999  2
;  946         1000  2       PFV [PFV$V_LOCK] = 1;
;  947         1001  2
;  948         1002  2       !+
;  949         1003  2       ! Set PFV_ADDR enable argument.
;  950         1004  2       !-
;  951         1005  2
;  952         1006  2       PFV_ADDR = PFV [PFV$R_PFV];
;  953         1007  2
;  954         1008  2       !+
;  955         1009  2       ! Call PAS$$CLOSE to do the work.
;  956         1010  2       !-
;  957         1011  2
;  958         1012  2       PAS$$CLOSE (PFV [PFV$R_PFV]);
;  959         1013  2
;  960         1014  2       !+
;  961         1015  2       ! Invalidate information in PFV
;  962         1016  2       !-
;  963         1017  2
;  964         1018  2       PFV [PFV$V_FCB_VALID] = 0;
;  965         1019  2       PFV [PFV$A_FCB] = 0;
;  966         1020  2
;  967         1021  2       RETURN;
;  968         1022  2
;  969         1023  1       END;                                       ! End of routine DO_CLOSE


                                                                .EXTRN  PAS$$CLOSE

                                           0004 00000 DO_CLOSE:
                                                                .WORD   Save R2                              ; 0940
                                      7E   D4 00002             CLRL    PFV_ADDR                             ; 0983
                    6D      001F      CF   DE 00004             MOVAL   1$, (FP)                             ;
                    52        04      AC   D0 00009             MOVL    PFV, R2                              ; 1000
               07   A2        80      8F   88 0000D             BISB2   #128, 7(R2)                          ;
                    6E                52   D0 00012             MOVL    R2, PFV_ADDR                         ; 1006
                                      52   DD 00015             PUSHL   R2                                   ; 1012
        00000000G   00                01   FB 00017             CALLS   #1, PAS$$CLOSE                       ;
               07   A2        40      8F   8A 0001E             BICB2   #64, 7(R2)                           ; 1018
                              0C      A2   D4 00023             CLRL    12(R2)                               ; 1019
                                           04 00026             RET                                          ; 1023
                                         0000 00027 1$:         .WORD   Save nothing                         ; 0983
                    50        08      AC   D0 00029             MOVL    8(AP), R0                            ;
                    50        04      A0   D0 0002D             MOVL    4(R0), R0                            ;
                              FC      A0   9F 00031             PUSHAB  PFV_ADDR                             ;
                                      01   DD 00034             PUSHL   #1                                   ;
                                      5E   DD 00036             PUSHL   SP                                   ;
                    7E        04      AC   7D 00038             MOVQ    4(AP), -(SP)                         ;
            0000V   CF                03   FB 0003C             CALLS   #3, DO_CLOSE_HANDLER                 ;
                                           04 00041             RET                                          ;

; Routine Size:  66 bytes,    Routine Base:  _PAS$CODE + 026E
;  970          1024  1
;  971          1025  1 !<BLF/PAGE>
```

```
1 %SBTTL 'DO_CLOSE_HANDLER - Error handler for DO_CLOSE'
1 ROUTINE DO_CLOSE_HANDLER (
1         SIGNAL_ARGS: REF BLOCK [, BYTE],        ! Signal arguments array
1         MECH_ARGS: REF BLOCK [, BYTE],          ! Mechanism arguments array
1         ENABLE_ARGS: REF VECTOR [, LONG]        ! Enable arguments array
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This is the condition handler enabled by DO_CLOSE.
1 !       If the current exception is a PAS$ message for the file
1 !       our establisher was processing, intercept the signal, use
1 !       $PUTMSG to display the message text, and unwind to our
1 !       establisher's caller.
1 !
1 !       The reason for using $PUTMSG is that DO_CLOSE may be called
1 !       from PAS$HANDLER during an unwind.  The current VAX
1 !       condition handling architecture does not specify what happens
1 !       when an exception occurs during an unwind, and the current
1 !       implementation performs the search for handlers incorrectly.
1 !       We are safe as long as we don't let the signal outside of the RTL.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       status.wlc.v = DO_CLOSE_HANDLER (SIGNAL_ARGS.rl.ra, MECH_ARGS.rl.ra
1 !                                       , ENABLE_ARGS.rl.ra)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SIGNAL_ARGS     - The signal argument list.
1 !
1 !       MECH_ARGS       - The mechanism argument list.
1 !
1 !       ENABLE_ARGS     - An array with the following
1 !                         format:
1 !
1 !                                       +-----------------+
1 !                                       :   ENB_COUNT     :  <-- ENABLE_ARGS
1 !                                       +-----------------+
1 !                                       :  ENB_PFV_ADDR   :
1 !                                       +-----------------+
1 !
1 !                         ENB_COUNT is the count of following enable arguments.
1 !                         The count is always 1.
1 !
1 !                         ENB_PFV_ADDR - If non-zero, the address of a longword
1 !                         containing the PFV our establisher is operating on.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       The signaller's PFV placed as the first FAO argument in the primary
1 !       signalled message.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       May use $PUTMSG to write a message
```

```
1 !
1 ! ROUTINE VALUE:
1 !
1 !       SS$_RESIGNAL
1 !
1 ! SIDE EFFECTS:
1 !
1 !       May cause an unwind.
1 !
1 !--
1
2     BEGIN
2
2     LITERAL
2         ENB_COUNT = 0,          ! Count of enable arguments
2         ENB_PFV_ADDR = 1;       ! Address of address of PFV
2
2     BUILTIN
2         ACTUALCOUNT;
2
2     !+
2     ! Determine if this is an unwind.
2     !-
2
2     IF .SIGNAL_ARGS [CHF$L_SIG_NAME] NEQU SS$_UNWIND
2     THEN
3         BEGIN
3
3         LOCAL
3             COND_NAME: BLOCK [4, BYTE]; ! Primary condition name
3
3         !+
3         ! Get primary condition name.
3         !-
3
3         COND_NAME = .SIGNAL_ARGS [CHF$L_SIG_NAME];
3
3         !+
3         ! Is this a PAS$ error?  If not, resignal.
3         !-
3
3         IF .COND_NAME [STS$V_FAC_NO] NEQU PAS$_FACILITY
3         THEN
3             RETURN SS$_RESIGNAL;
3
3         !+
3         ! See if the error message is one which is "trapped"
3         ! by ERROR:=CONTINUE.  This is done by comparing the
3         ! message number against a select range.
3         !-
3
3         IF .COND_NAME [STS$V_CODE] GEQU PAS$$K_MSGCONTLO AND ! Lowest number
3             .COND_NAME [STS$V_CODE] LEQU PAS$$K_MSGCONTHI
3         THEN
4             BEGIN
4
4             !+
; 1087          1140 4                  ! See if the PFVs match.  The signaller's PFV is the
; 1088          1141 4                  ! first FAO parameter in the primary message.
; 1089          1142 4                  !-
; 1090          1143 4
; 1091          1144 4                  IF .SIGNAL_ARGS [12,0,32,0] EQLA ..ENABLE_ARGS [ENB_PFV_ADDR]
; 1092          1145 4                  THEN
; 1093          1146 5                      BEGIN
; 1094          1147 5
; 1095          1148 5                      !+
; 1096          1149 5                      ! We want to use $PUTMSG to display the message, and then
; 1097          1150 5                      ! unwind to our establisher's caller.  First, subtract two
; 1098          1151 5                      ! from the signal argument count so that $PUTMSG doesn't see
; 1099          1152 5                      ! the PC and PSL.
; 1100          1153 5                      !-
; 1101          1154 5
; 1102          1155 5                      SIGNAL_ARGS [CHF$L_SIG_ARGS] = .SIGNAL_ARGS [CHF$L_SIG_ARGS] - 2;
; 1103          1156 5                      COND_NAME [STS$V_SEVERITY] = STS$K_ERROR;         ! Make E severity
; 1104          1157 5                      SIGNAL_ARGS [CHF$L_SIG_NAME] = .COND_NAME;
; 1105          1158 5                      $PUTMSG (MSGVEC = SIGNAL_ARGS [CHF$L_SIG_ARGS]);
; 1106          1159 5                      SIGNAL_ARGS [CHF$L_SIG_ARGS] = .SIGNAL_ARGS [CHF$L_SIG_ARGS] + 2;
; 1107          1160 5
; 1108          1161 6                      IF NOT $UNWIND ()
; 1109          1162 5                      THEN
; 1110          1163 5                          $PAS$BUGCHECK (BUG_UNWINDFAIL);
; 1111          1164 4                      END;
; 1112          1165 3                  END;
; 1113          1166 2              END;
; 1114          1167 2
; 1115          1168 2          RETURN SS$_RESIGNAL;                            ! Resignal error
; 1116          1169 2
; 1117          1170 1          END;                                            ! End of routine DO_CLOSE_HANDLER


                                                                .EXTRN  PAS$_FACILITY, PAS$$K_MSGCONTLO
                                                                .EXTRN  PAS$$K_MSGCONTHI
                                                                .EXTRN  SYS$PUTMSG, SYS$UNWIND

                                                 0004 00000 DO_CLOSE_HANDLER:
                                                                .WORD   Save R2                                         ; 1027
                               52        04    AC  D0 00002     MOVL    SIGNAL_ARGS, R2                                 ; 1107
                     00000920  8F        04    A2  D1 00006     CMPL    4(R2), #2336                                    ;
                                               5F  13 0000E     BEQL    1$                                              ;
                               51        04    A2  D0 00010     MOVL    4(R2), COND_NAME                                ; 1118
        00G            51      0C              10  ED 00014     CMPZV   #16, #12, COND_NAME, S^PAS$_FACILITY            ; 1124
                                               54  12 00019     BNEQ    1$                                              ;
00000000G  8F          51      0C              03  ED 0001B     CMPZV   #3, #12, COND_NAME, #PAS$$K_MSGCONTLO           ; 1134
                                               49  1F 00024     BLSSU   1$                                              ;
00000000G  8F          51      0C              03  ED 00026     CMPZV   #3, #12, COND_NAME, #PAS$$K_MSGCONTHI           ; 1135
                                               3E  1A 0002F     BGTRU   1$                                              ;
                               50        0C    AC  D0 00031     MOVL    ENABLE_ARGS, R0                                 ; 1144
                          04   B0        0C    A2  D1 00035     CMPL    12(R2), @4(R0)                                  ;
                                               33  12 0003A     BNEQ    1$                                              ;
                               62              02  C2 0003C     SUBL2   #2, (R2)                                        ; 1155
        51             03      00              02  F0 0003F     INSV    #2, #0, #3, COND_NAME                           ; 1156
                          04   A2              51  D0 00044     MOVL    COND_NAME, 4(R2)                                ; 1157
                                               7E  7C 00048     CLRQ    -(SP)                                           ; 1158
```

```
                                        7E   D4 0004A          CLRL    -(SP)                             ;
                                        52   DD 0004C          PUSHL   R2                                ;
                    00000000G  00       04   FB 0004E          CALLS   #4, SYS$PUTMSG                    ;
                               62       02   C0 00055          ADDL2   #2, (R2)                          ; 1159
                                        7E   7C 00058          CLRQ    -(SP)                             ; 1161
                    00000000G  00       02   FB 0005A          CALLS   #2, SYS$UNWIND                    ;
                               0B       50   E8 00061          BLBS    R0, 1$                            ;
                                        03   DD 00064          PUSHL   #3                                ; 1163
                    00000000G  00       01   FB 00066          CALLS   #1, PAS$$BUGCHECK                 ;
                                        06   11 0006D          BRB     2$                                ;
                               50  0918 8F   3C 0006F 1$:      MOVZWL  #2328, R0                         ; 1168
                                             04 00074          RET                                       ;
                                        50   D4 00075 2$:      CLRL    R0                                ; 1170
                                             04 00077          RET                                       ;

; Routine Size:  120 bytes,    Routine Base:  _PAS$CODE + 02B0


; 1118          1171  1
; 1119          1172  1 !<BLF/PAGE>
```

```
1 %SBTTL 'INITIALIZE_QUEUE - Initialize FILE_QUEUE'
1 ROUTINE INITIALIZE_QUEUE
1     : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Initializes FILE_QUEUE to be an empty queue.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       INITIALIZE_QUEUE ()
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       FILE_QUEUE
1 !       QUEUE_INITIALIZED
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       FILE_QUEUE
1 !       QUEUE_INITIALIZED
1 !
1 ! COMPLETION STATUS:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Makes FILE_QUEUE an empty queue.
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       NONE
1 !--
1
2     BEGIN
2
2     LOCAL
2         AST_STATUS;                         ! Previous AST enable status
2
2     BUILTIN
2         TESTBITCS;
2
2     !+
2     ! Disable ASTs.
2     !-
2
2     AST_STATUS = $SETAST (ENBFLG = 0);
2
2     !+
2     ! If QUEUE_INITIALIZED is still clear, initialize FILE_QUEUE to
2     ! be an empty queue.  Set QUEUE_INITIALIZED.
```

```
; 1178          1230  2      !-
; 1179          1231  2
; 1180          1232  2      IF TESTBITCS (QUEUE_INITIALIZED)
; 1181          1233  2      THEN
; 1182          1234  3          BEGIN
; 1183          1235  3          FILE_QUEUE [0] = FILE_QUEUE;            ! Set forward link
; 1184          1236  3          FILE_QUEUE [1] = .FILE_QUEUE [0];       ! Set backward link
; 1185          1237  2          END;
; 1186          1238  2
; 1187          1239  2      !+
; 1188          1240  2      ! Reenable ASTs if previously enabled.
; 1189          1241  2      !-
; 1190          1242  2
; 1191          1243  2      IF .AST_STATUS EQL SS$_WASSET
; 1192          1244  2      THEN
; 1193          1245  2          $SETAST (ENBFLG = 1);
; 1194          1246  2
; 1195          1247  2      RETURN;
; 1196          1248  2
; 1197          1249  1      END;                                        ! End of routine INITIALIZE_QUEUE



                                                  000C 00000 INITIALIZE_QUEUE:
                                                                        .WORD   Save R2,R3                              ; 1174
                                   53 00000000G  00  9E 00002           MOVAB   SYS$SETAST, R3                          ;
                                   52 00000000'  EF  9E 00009           MOVAB   FILE_QUEUE, R2                          ;
                                                 7E  D4 00010           CLRL    -(SP)                                   ; 1225
                                   63            01  FB 00012           CALLS   #1, SYS$SETAST                          ;
                   07        0C    A2            00  E2 00015           BBSS    #0, QUEUE_INITIALIZED, 1$               ; 1232
                                   62            62  9E 0001A           MOVAB   FILE_QUEUE, FILE_QUEUE                  ; 1235
                             04    A2            62  D0 0001D           MOVL    FILE_QUEUE, FILE_QUEUE+4                ; 1236
                                   09            50  D1 00021 1$:       CMPL    AST_STATUS, #9                          ; 1243
                                                 05  12 00024           BNEQ    2$                                      ;
                                                 01  DD 00026           PUSHL   #1                                      ; 1245
                                   63            01  FB 00028           CALLS   #1, SYS$SETAST                          ;
                                                     04 0002B 2$:       RET                                             ; 1249

; Routine Size:  44 bytes,    Routine Base:  _PAS$CODE + 0328


; 1198          1250  1
; 1199          1251  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SERVICE_REQUEST - Service remove request'
1 ROUTINE SERVICE_REQUEST
1     : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Removes all FCBs from FILE_QUEUE that have DEALLOC set.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       SERVICE_REQUEST ()
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       FILE_QUEUE
1 !       REQUEST_LEVEL
1 !       REMOVE_REQUESTED
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       FILE_QUEUE
1 !       REQUEST_LEVEL
1 !       REMOVE_REQUESTED
1 !
1 ! COMPLETION STATUS:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Removes FCBs from queue.
1 !
1 ! SIGNALLED ERRORS:
1 !
1 !       NONE
1 !--
1
2     BEGIN
2
2     LOCAL
2         FREE_LIST: REF VECTOR [, LONG];         ! List of FCBs we deallocated
2
2     BUILTIN
2         REMQUE;
2
2     !+
2     ! Initialize FREE_LIST.
2     !-
2
2     FREE_LIST = 0;
2
2     !+
2        ! Increment REQUEST_LEVEL.  If we are at level zero, then we can
2        ! scan the queue and do REMQUEs.
2        !-
2
2        IF (REQUEST_LEVEL=.REQUEST_LEVEL+1) EQL 0
2        THEN
3            BEGIN
3
3            LOCAL
3                AST_STATUS,                      ! Previous AST enable status
3                CURRENT_FCB: REF VECTOR [, LONG];   ! Current FCB to look at
3
3            !+
3            ! Disable ASTs and remember previous status.  This makes us
3            ! multi-stream AST reentrant.
3            !-
3
3            AST_STATUS = $SETAST (ENBFLG = 0);
3
3            !+
3            ! Get first FCB on FILE_QUEUE.
3            !-
3
3            CURRENT_FCB = .FILE_QUEUE [0];  ! Forward link
3
3            !+
3            ! Clear RENOVE_REQUESTED.
3            !-
3
3            REMOVE_REQUESTED = 0;
3
3            !+
3            ! While we haven't run out of FCBs, look for FCBs with the
3            ! DEALLOC bit set, remove them from the queue, and insert them
3            ! on the list of blocks to be freed.
3            !-
3
3            WHILE (.CURRENT_FCB NEQA FILE_QUEUE) DO ! Back at queue header?
4                BEGIN
4                !+
4                ! Allow offset to zero-origin of FCB.
4                !-
4                LOCAL
4                    FCB_ORIGIN: REF $PAS$FCB_CONTROL_BLOCK;
4                FCB_ORIGIN = .CURRENT_FCB + FCB$K_BLN;
4                IF .FCB_ORIGIN [FCB$V_DEALLOC]
4                THEN
5                    BEGIN
5                    LOCAL
5                        TEMP;                  ! Output from REMQUE
5                    REMQUE (CURRENT_FCB [0], TEMP);
5                    CURRENT_FCB [0] = .FREE_LIST;    ! Add FCB to free list
5                    FREE_LIST = .CURRENT_FCB;
4                    END;
4
4                !+
4                ! Get next FCB from the queue.
```

```
; 1315         1366  4               !-
; 1316         1367  4
; 1317         1368  4               CURRENT_FCB = .CURRENT_FCB [0];       ! Forward link
; 1318         1369  3               END;
; 1319         1370  3
; 1320         1371  3           !+
; 1321         1372  3           ! Reenable ASTs if they were previously enabled.
; 1322         1373  3           !-
; 1323         1374  3
; 1324         1375  3           IF .AST_STATUS EQL SS$_WASSET
; 1325         1376  3           THEN
; 1326         1377  3               $SETAST (ENBFLG = 1);
; 1327         1378  3
; 1328         1379  2           END;
; 1329         1380  2
; 1330         1381  2       !+
; 1331         1382  2       ! Decrement REQUEST_LEVEL.
; 1332         1383  2       !-
; 1333         1384  2
; 1334         1385  2       REQUEST_LEVEL = .REQUEST_LEVEL - 1;
; 1335         1386  2
; 1336         1387  2       !+
; 1337         1388  2       ! Free all blocks on FREE_LIST.
; 1338         1389  2       !-
; 1339         1390  2
; 1340         1391  2       WHILE (.FREE_LIST NEQA 0) DO
; 1341         1392  3           BEGIN
; 1342         1393  3           LOCAL
; 1343         1394  3               BLOCK_ADDR;
; 1344         1395  3           BLOCK_ADDR = .FREE_LIST;
; 1345         1396  3           FREE_LIST = .FREE_LIST [0];
; 1346         1397  3           PAS$$FREE_VM (PAS$K_FILE_DYN_BLN, BLOCK_ADDR);
; 1347         1398  2           END;
; 1348         1399  2
; 1349         1400  2       RETURN;
; 1350         1401  2
; 1351         1402  1       END;                                       ! End of routine SERVICE_REQUEST


                                    003C 00000 SERVICE_REQUEST:
                                                     .WORD   Save R2,R3,R4,R5                          ; 1253
                  55 00000000G  00   9E 00002        MOVAB   SYS$SETAST, R5                            ;
                  54 00000000'  EF   9E 00009        MOVAB   REQUEST_LEVEL, R4                         ;
                  5E            04   C2 00010        SUBL2   #4, SP                                    ;
                                53   D4 00013        CLRL    FREE_LIST                                 ; 1306
                                64   D6 00015        INCL    REQUEST_LEVEL                             ; 1313
                                36   12 00017        BNEQ    4$                                        ;
                                7E   D4 00019        CLRL    -(SP)                                     ; 1326
                  65            01   FB 0001B        CALLS   #1, SYS$SETAST                            ;
                  51        F8  A4   D0 0001E        MOVL    FILE_QUEUE, CURRENT_FCB                   ; 1332
                            08  A4   D4 00022        CLRL    REMOVE_REQUESTED                          ; 1338
                  52        F8  A4   9E 00025 1$:    MOVAB   FILE_QUEUE, R2                            ; 1346
                  52            51   D1 00029        CMPL    CURRENT_FCB, R2                           ;
                                17   13 0002C        BEQL    3$                                        ;
```

```
                        52      44  A1  9E 0002E        MOVAB   68(R1), FCB_ORIGIN                             ; 1353
            09      FE  A2          01  E1 00032        BBC     #1, -2(FCB_ORIGIN), 2$                         ; 1354
                        52          61  0F 00037        REMQUE  (CURRENT_FCB), TEMP                            ; 1359
                        61          53  D0 0003A        MOVL    FREE_LIST, (CURRENT_FCB)                       ; 1360
                        53          51  D0 0003D        MOVL    CURRENT_FCB, FREE_LIST                         ; 1361
                        51          61  D0 00040 2$:    MOVL    (CURRENT_FCB), CURRENT_FCB                     ; 1368
                                    E0  11 00043        BRB     1$                                             ; 1346
                        09          50  D1 00045 3$:    CMPL    AST_STATUS, #9                                 ; 1375
                                    05  12 00048        BNEQ    4$                                             ;
                                    01  DD 0004A        PUSHL   #1                                             ; 1377
                        65          01  FB 0004C        CALLS   #1, SYS$SETAST                                 ;
                                    64  D7 0004F 4$:    DECL    REQUEST_LEVEL                                  ; 1385
                                    53  D5 00051 5$:    TSTL    FREE_LIST                                      ; 1391
                                    16  13 00053        BEQL    6$                                             ;
                        6E          53  D0 00055        MOVL    FREE_LIST, BLOCK_ADDR                          ; 1395
                        53          63  D0 00058        MOVL    (FREE_LIST), FREE_LIST                         ; 1396
                                    5E  DD 0005B        PUSHL   SP                                             ; 1397
                        7E    0138  8F  3C 0005D        MOVZWL  #312, -(SP)                                    ;
               00000000G  00        02  FB 00062        CALLS   #2, PAS$$FREE_VM                               ;
                                    E6  11 00069        BRB     5$                                             ; 1391
                                        04 0006B 6$:    RET                                                    ; 1402

; Routine Size:  108 bytes,    Routine Base:  _PAS$CODE + 0354


; 1352          1403  1
; 1353          1404  1 !<BLF/PAGE>
; 1355          1405  1 END                                             ! End of module PAS$$FILE_UTIL
; 1356          1406  1
; 1357          1407  0 ELUDOM




;                                   PSECT SUMMARY
;
;       Name                    Bytes                         Attributes
;
;   _PAS$DATA                       20  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;   _PAS$CODE                      960  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        19         0        581         00:00.9
;   _$255$DUA28:[PASRTL.OBJ]PASLIB.L32;1         427       105        24         33         00:00.4




;                                   COMMAND QUALIFIERS

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

; Size:         960 code + 20 data bytes
; Run Time:        00:21.6
; Elapsed Time:    01:09.6
; Lines/CPU Min:    3915
; Lexemes/CPU-Min: 16758
; Memory Used:  102 pages
; Compilation Complete
```

0294 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 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