D 7

```
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLL              EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  EEEEEEEEEEEEEEE
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLL              EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  EEEEEEEEEEEEEEE
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLL              EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  EEEEEEEEEEEEEEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEEEEEEEEEEE     LLL              EEEEEEEEEEEE           TTT        EEEEEEEEEEEE
DDD         DDD  EEEEEEEEEEEE     LLL              EEEEEEEEEEEE           TTT        EEEEEEEEEEEE
DDD         DDD  EEEEEEEEEEEE     LLL              EEEEEEEEEEEE           TTT        EEEEEEEEEEEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDD         DDD  EEE              LLL              EEE                    TTT        EEE
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  EEEEEEEEEEEEEEE        TTT        EEEEEEEEEEEEEEE
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  EEEEEEEEEEEEEEE        TTT        EEEEEEEEEEEEEEE
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  EEEEEEEEEEEEEEE        TTT        EEEEEEEEEEEEEEE
```

```
**FILE**ID**PURGE                                    I 10
```

```
PPPPPPPP    UU      UU  RRRRRRRR      GGGGGGGG  EEEEEEEEEE
PPPPPPPP    UU      UU  RRRRRRRR      GGGGGGGG  EEEEEEEEEE
PP      PP  UU      UU  RR      RR  GG          EE
PP      PP  UU      UU  RR      RR  GG          EE
PP      PP  UU      UU  RR      RR  GG          EE
PP      PP  UU      UU  RR      RR  GG          EE
PPPPPPPP    UU      UU  RRRRRRRR    GG          EEEEEEEE
PPPPPPPP    UU      UU  RRRRRRRR    GG          EEEEEEEE
PP          UU      UU  RR  RR      GG  GGGGGG  EE
PP          UU      UU  RR  RR      GG  GGGGGG  EE
PP          UU      UU  RR    RR    GG      GG  EE              ....
PP          UU      UU  RR    RR    GG      GG  EE              ....
PP          UUUUUUUUUU  RR      RR    GGGGGG    EEEEEEEEEE      ....
PP          UUUUUUUUUU  RR      RR    GGGGGG    EEEEEEEEEE      ....

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 purge (                    ! Purge directory program
0                 IDENT = 'V04-000',
0                 ADDRESSING_MODE(EXTERNAL=GENERAL)
0                 ) =
1 BEGIN
1
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 ! FACILITY:  PURGE Command
1 !
1 ! ABSTRACT:
1 !
1 !       This utility purges a directory, basically removing
1 !       old versions of a specified group of files.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX/VMS operating system. unprivileged user mode,
1 !
1 ! AUTHOR:  Tim Halvorsen,                    CREATION DATE: Oct-1979
1 !
1 ! Modified by:
1 !
1 !       V03-007 SHZ0009         Stephen H. Zalewksi,    15-Mar-1984
1 !               Modify PURGE algorithm to make sticky searchlists work.
1 !
1 !       V03-006 SHZ0008         Stephen H. Zalewski,    21-Feb-1984
1 !               Add support for sticky searchlists.
1 !
1 !       V03-005 SHZ0007         Stephen H. Zalewski,    27-Dec-1983
1 !               Do defaulting of file name and type in module PURGE_FILES.
1 !               Add performance enhancement that cuts down on the number
1 !               of RMS $OPEN's and $CLOSE's that must be done to purge
1 !               a file.
```

```
!
!       V03-004 SHZ0006         Stephen H. Zalewski     25-Feb-1983
!               If PURGE command was issued on ODS-2 disk, do not cache the
!               filenames before attempting to delete the files.
!
!       V03-003 SHZ0005         Stephen H. Zalewski,    4-Nov-1982  15:42
!               Modify PURGE to use common command qualifier package.
!
!       V03-002 SHZ0004         Stephen H. Zalewski,    26-Aug-1982  22:19
!               Fix bug in SHZ0003 that caused PURGE to ACCVIO if you purged
!               an empty directory.  Also fixed bug that caused "No files
!               purged" message to be printed even when files were purged.
!
!               Fix bug introduced in SHZ0003 that prevented dangling directory
!               entries from being deleted if the PURGE comand was issued
!               with /LOG qualifier.
!
!               Finally, if a file is opened because of /SINCE, /BEFORE or
!               /LOG qualifiers being present, leave it open until we actually
!               delete it to optimize the number of FAL jobs necessary to do
!               the job in case we are doing this over the net.
!
!       V03-001 SHZ0003         Stephen H. Zalewski,    10-Aug-1982  21:24
!               Clean up error handling.  Modified routines to use new CLI.
!               Made PURGE/LOG also display size of file purged.
!
!--
```

```
1 LIBRARY 'SYS$LIBRARY:STARLET.L32';                                  ! VAX/VMS common definitions
1 REQUIRE 'SRC$:DELETE.REQ';                                          ! Common DELETE definitions
1
1
1 FORWARD ROUTINE
1     purge_files,                                                    ! Main directory routine
1     build_list :          NOVALUE,                                  ! Process each wildcard file for ods-1 disk
1     purge_ods1_directory :        NOVALUE,                          ! Purge versions for a directory
1     purge_ods2_files :  NOVALUE,                                    ! Purge files for ODS-2 disk.
1     purge_this_file :   NOVALUE;                                    ! Routine to delete a file.
1
1
1 EXTERNAL ROUTINE
1     del$search_error :  NOVALUE,                                    ! Report open/search error.
1     del$file_error :    NOVALUE,                                    ! Report RMS error messages
1     lib$file_scan,                                                  ! Search wildcard specifications
1     lib$set_erase,                                                  ! Mark file for erase-on-delete
1     lib$cvt_dtb,                                                    ! Convert decimal string to binary
1     lib$get_vm,                                                     ! Allocate dynamic storage
1     lib$free_vm,                                                    ! Deallocate dynamic storage
1     lib$qual_file_match;                                            ! Check to see if file should be purged.
1
1
1 EXTERNAL
1     scan_context,
1     del$cli_status:       $BBLOCK,                                  ! CLI qualifier bitmap
1     del$keepver_val,                                                ! Number of versions to save
1     del$files_deleted,                                              ! Number of files purged
1     del$file_size,                                                  ! Size of file being deleted
1     del$blocks_deleted,                                             ! Number of blocks released
1     del$context,                                                    ! Context longword for common qualifier package.
1     lib$_quipro,                                                    ! Return status code from common qualifier package.
1     lib$_filfaimat;                                                 ! Return status code from common qualifier package.
1
1
1 GLOBAL
1     version_list :        INITIAL(0);                               ! Address of linked list of versions
1
1 OWN
1     versions :            INITIAL(0),                               ! Number of versions seen so far
1     prev_name_len :       INITIAL(0),                               ! Length of previous name & type
1     prev_name :           VECTOR [nam$c_maxrss,BYTE],               ! Buffer to hold previous name/type
1     prev_dir_len :        INITIAL(0),                               ! Length of previous directory
1     prev_dir :            VECTOR [nam$c_maxrss,BYTE];               ! Buffer for previous directory
1
```

```
  1 GLOBAL ROUTINE purge_files (fab_block) =
  1
  1 !++
  1 ! Functional description
  1 !
  1 !       This routine performs all the main processing of the
  1 !       PURGE command.  The command line has already been parsed
  1 !       and the qualifier values saved.
  1 !
  1 ! Calling sequence
  1 !
  1 !       purge_files(fab) from the DELETE command mainline code
  1 !
  1 ! Input parameters
  1 !
  1 !       fab_block = Address of FAB with FNA, FNS filled in.
  1 !
  1 ! Output parameters
  1 !
  1 !       None
  1 !
  1 ! Routine value
  1 !
  1 !       Status.
  1 !----
  1
  2 BEGIN
  2
  2 MAP
  2     fab_block:  REF $BBLOCK;                            ! Address of FAB block
  2
  2 BIND
  2     nam_block = .fab_block [fab$l_nam]: $BBLOCK;        ! Address of NAM block
  2
  2 LOCAL
  2     status,
  2     devnam_desc : VECTOR[2],
  2     itmlst : VECTOR[4,LONG],
  2     buffer : INITIAL(0);
  2
  4 IF ((NOT .$BBLOCK[fab_block [fab$l_dev], dev$v_dir])    ! If not a directory device,
  3 AND (NOT .$BBLOCK[fab_block [fab$l_dev], dev$v_net]))   ! and not a network device
  2 THEN
  2     RETURN false;
  2
  3 IF (NOT .$BBLOCK[fab_block [fab$l_dev], dev$v_net])     ! If not a network device
  2 THEN                                                    !  then check to see if
  3     BEGIN                                               !  we are purging an ODS-2 disk.
  3     devnam_desc[0] = .nam_block[nam$b_dev];
  3     devnam_desc[1] = .nam_block[nam$l_dev];
  3     itmlst<0,16> = 4;
  3     itmlst<16,16> = dvi$_acptype;
  3     itmlst[1] = buffer;
  3     itmlst[2] = 0;
  3     itmlst[3] = 0;
P 3     status = $GETDVIW (DEVNAM = devnam_desc,
  3                        ITMLST = itmlst);
```

```
;  189          0292  3        IF NOT .status
;  190          0293  3        THEN
;  191          0294  3            SIGNAL_STOP (.status);
;  192          0295  2        END;
;  193          0296  2
;  194          0297  2  IF .buffer EQL dvi$c_acp_f11v2                          ! If this is an ODS-2 disk
;  195          0298  2      THEN purge_ods2_files (.fab_block)                  !  then use optimized purge routine
;  196          0299  2      ELSE build_list(.fab_block);                        !  else, use old one.
;  197          0300  2
;  198          0301  2  RETURN true;
;  199          0302  1  END;


                                                         .TITLE  PURGE
                                                         .IDENT  \V04-000\

                                                         .PSECT  $OWN$,NOEXE,2

                              00000000  00000 VERSIONS:
                                                         .LONG   0                                           ;
                              00000000  00004 PREV_NAME_LEN:
                                                         .LONG   0                                           ;
                                        00008 PREV_NAME:
                                                         .BLKB   255
                                        00107            .BLKB   1
                              00000000  00108 PREV_DIR_LEN:
                                                         .LONG   0                                           ;
                                        0010C PREV_DIR:
                                                         .BLKB   255

                                                         .PSECT  $GLOBAL$,NOEXE,2

                              00000000  00000 VERSION_LIST::
                                                         .LONG   0                                           ;

                                                         .EXTRN  DEL$SEARCH_ERROR
                                                         .EXTRN  DEL$FILE_ERROR, LIB$FILE_SCAN
                                                         .EXTRN  LIB$SET_ERASE, LIB$CVT_DTB
                                                         .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                                         .EXTRN  LIB$QUAL_FILE_MATCH
                                                         .EXTRN  SCAN_CONTEXT, DEL$CLI_STATUS
                                                         .EXTRN  DEL$KEEPVER_VAL
                                                         .EXTRN  DEL$FILES_DELETED
                                                         .EXTRN  DEL$FILE_SIZE, DEL$BLOCKS_DELETED
                                                         .EXTRN  DEL$CONTEXT, LIB$_QUIPRO
                                                         .EXTRN  LIB$_FILFAIMAT, SYS$GETDVIW

                                                         .PSECT  $CODE$,NOWRT,2

                                 0004  00000             .ENTRY  PURGE_FILES, Save R2                        ; 0235
                 5E                18  C2 00002          SUBL2   #24, SP                                     ;
                 52          04    AC  D0 00005          MOVL    FAB_BLOCK, R2                               ; 0267
                 50          28    A2  D0 00009          MOVL    40(R2), R0                                  ;
                                   7E  D4 0000D          CLRL    BUFFER                                      ;
     07          40    A2          03  E0 0000F          BBS     #3, 64(R2), 1$                              ; 0275
     3F          41    A2          05  E0 00014          BBS     #5, 65(R2), 2$                              ; 0276
                                   56  11 00019          BRB     5$                                          ; 0278
```

```
                38        41  A2          05  E0 0001B 1$:     BBS     #5, 65(R2), 2$                        ; 0280
                          14  AE      39  A0  9A 00020         MOVZBL  57(R0), DEVNAM_DESC                   ; 0283
                          18  AE      44  A0  D0 00025         MOVL    68(R0), DEVNAM_DESC+4                 ; 0284
                          04  AE 00420004 8F  D0 0002A         MOVL    #4325380, ITMLST                      ; 0285
                          08  AE          6E  9E 00032         MOVAB   BUFFER, ITMLST+4                      ; 0287
                                      0C  AE  7C 00036         CLRQ    ITMLST+8                              ; 0288
                                          7E  7C 00039         CLRQ    -(SP)                                 ; 0291
                                          7E  7C 0003B         CLRQ    -(SP)                                 ;
                                      14  AE  9F 0003D         PUSHAB  ITMLST                                ;
                                      28  AE  9F 00040         PUSHAB  DEVNAM_DESC                           ;
                                          7E  7C 00043         CLRQ    -(SP)                                 ;
                   00000000G  00          08  FB 00045         CALLS   #8, SYS$GETDVIW                       ;
                              09          50  E8 0004C         BLBS    STATUS, 2$                            ; 0292
                                          50  DD 0004F         PUSHL   STATUS                                ; 0294
                   00000000G  00          01  FB 00051         CALLS   #1, LIB$STOP                          ;
                              02          6E  D1 00058 2$:     CMPL    BUFFER, #2                            ; 0297
                                          09  12 0005B         BNEQ    3$                                    ;
                                          52  DD 0005D         PUSHL   R2                                    ; 0298
                       0000V  CF          01  FB 0005F         CALLS   #1, PURGE_ODS2_FILES                  ;
                                          07  11 00064         BRB     4$                                    ;
                                          52  DD 00066 3$:     PUSHL   R2                                    ; 0299
                       0000V  CF          01  FB 00068         CALLS   #1, BUILD_LIST                        ;
                              50          01  D0 0006D 4$:     MOVL    #1, R0                                ; 0301
                                              04 00070         RET                                           ;
                                          50  D4 00071 5$:     CLRL    R0                                    ; 0302
                                              04 00073         RET                                           ;

; Routine Size:  116 bytes,    Routine Base:  $CODE$ + 0000


;  200          0303  1
```

```
1 ROUTINE build_list (fab_block): NOVALUE =
1
1 !---
1 !
1 ! Functional description
1 !
1 !       This routine is called as an action routine from the directory wildcard
1 !       searching process.  It is given a FAB containing the full name
1 !       of the file to be processed.  It is used when purging an ods-1 disk to
1 !       build the filenames into an ods-2 list.
1 !
1 ! Input parameters
1 !
1 !       fab_block = Address of FAB
1 !
1 ! Output parameters
1 !
1 !       None
1 !
1 !----
1
2 BEGIN
2
2 MAP
2     fab_block:  REF $BBLOCK;                            ! Address of FAB
2
2 BIND
2     nam = .fab_block [fab$l_nam]: $BBLOCK;              ! Address of NAM
2
2 LOCAL
2     status,                                             ! Status code
2     dir_len,                                            ! Length of device/directory
2     version,                                            ! Version number in binary form
2     length,                                             ! Length of new version entry
2     entry:      REF VECTOR,                             ! Address of new version entry
2     prev:       REF VECTOR,                             ! Address of previous entry scanned
2     curr:       REF VECTOR;                             ! Address of current entry scanned
2
2 !
2 ! If we have reached a new directory, then peruse the linked
2 ! list of versions and delete the all but the # explicitly kept.
2 !
2
2 dir_len = .nam [nam$b_node] + .nam [nam$b_dev] + .nam [nam$b_dir];
2
2 IF CH$NEQ(                                              ! If new directory
2         .prev_dir_len, prev_dir,
2         .dir_len, .nam [nam$l_rsa], 0)
2 THEN
3     BEGIN
3     CH$MOVE(.dir_len, .nam [nam$l_rsa], prev_dir);
3     prev_dir_len = .dir_len;
3     purge_ods1_directory(version_list);                 ! Delete old versions
2     END;
2
2 !
2 ! Convert the version string to binary form
```

```
2 !
2
2 status = LIB$CVT_DTB(.nam [nam$b_ver]-1,                    ! Store version in binary
2                     .nam [nam$l_ver]+1,
2                     version);
2 IF NOT .status                                              ! If error converting value,
2 THEN
3     BEGIN
3     SIGNAL(.status);                                        ! signal the error
3     RETURN;
3     END;
2
2 !
2 ! Add this version to the ordered linked list of files for this directory.
2 ! This list is ordered first by file name & type in ascending order, and
2 ! then by version in descending order (this is the same as ODS-2).
2 !
2
2 length = 6*4 + .nam [nam$b_rsl];                            ! Length of overhead plus filespec
2 status = LIB$GET_VM(length,entry);                          ! Allocate storage for entry
2 IF NOT .status                                              ! If error allocating storage,
2 THEN
2     SIGNAL_STOP(.status);                                   ! then signal the error
2
2 entry [1] = .nam [nam$b_name] + .nam [nam$b_type];          ! Length of name/type
2 entry [2] = entry [6]                                       ! Address of name/type
2         + (.nam [nam$l_name] - .nam [nam$l_rsa]);
2 entry [3] = .version;                                       ! Store binary version number
2 entry [4] = .nam [nam$b_rsl];                               ! Store length of filespec
2 entry [5] = entry [6];                                      ! and address of filespec
2 CH$MOVE(.nam [nam$b_rsl], .nam [nam$l_rsa], entry [6]);     ! Store full filespec
2
2 prev = version_list;                                        ! Address of previous entry
2 curr = .prev [0];                                           ! Start at first entry
2
2 WHILE .curr NEQ 0                                           ! For each entry in list,
2 DO
3     BEGIN
3     LOCAL comparison;                                       ! -1 if less, 0 if equal, 1 if greater
3     comparison = CH$COMPARE(.entry [1], .entry [2],
3         .curr [1], .curr [2], 0);                           ! Compare new to old name/type
3     IF .comparison LSS 0                                    ! If found place to insert entry,
4         OR (.comparison EQL 0                               ! (ascending order by name/type,
4        AND .entry [3] GTRU .curr [3])                       ! (then descending order by version)
3     THEN
3         EXITLOOP;                                           ! then exit the loop
3     prev = .curr;                                           ! Save address of previous entry done
3     curr = .curr [0];                                       ! and link to next in list
3     END;
2
2 entry [0] = .prev [0];                                      ! Make new entry point to next in list
2 prev [0] = .entry;                                          ! Make previous entry point to new one
2
2 RETURN;
2
1 END;
```

```
                                             01FC 00000 BUILD_LIST:
                                                              .WORD   Save R2,R3,R4,R5,R6,R7,R8                         ; 0304
                              58    0000'  CF  9E 00002       MOVAB   PREV_DIR_LEN, R8                                  ;
                              5E           0C  C2 00007       SUBL2   #12, SP                                           ;
                              50       04  AC  D0 0000A       MOVL    FAB_BLOCK, R0                                     ; 0331
                              57       28  A0  D0 0000E       MOVL    40(R0), R7                                        ;
                              50       38  A7  9A 00012       MOVZBL  56(R7), R0                                        ; 0347
                              51       39  A7  9A 00016       MOVZBL  57(R7), R1                                        ;
                              50           51  C0 0001A       ADDL2   R1, R0                                            ;
                              56       3A  A7  9A 0001D       MOVZBL  58(R7), DIR_LEN                                   ;
                              56           50  C0 00021       ADDL2   R0, DIR_LEN                                       ;
     56           00     04   A8           68  2D 00024       CMPC5   PREV_DIR_LEN, PREV_DIR, #0, DIR_LEN, @4(R7)       ; 0349
                                       04  B7     0002A                                                                 ;
                                           12  13 0002C       BEQL    1$                                                ;
             04   A8     04   B7           56  28 0002E       MOVC3   DIR_LEN, @4(R7), PREV_DIR                         ; 0354
                              68           56  D0 00034       MOVL    DIR_LEN, PREV_DIR_LEN                             ; 0355
                                    0000'  CF  9F 00037       PUSHAB  VERSION_LIST                                      ; 0356
                      0000V   CF           01  FB 0003B       CALLS   #1, PURGE_ODS1_DIRECTORY                          ;
                                           5E  DD 00040 1$:   PUSHL   SP                                                ; 0363
                  7E     54   A7           01  C1 00042       ADDL3   #1, 84(R7), -(SP)                                 ; 0364
                              7E       3D  A7  9A 00047       MOVZBL  61(R7), -(SP)                                     ; 0363
                                           6E  D7 0004B       DECL    (SP)                                              ;
                  00000000G   00           03  FB 0004D       CALLS   #3, LIB$CVT_DTB                                   ;
                              52           50  D0 00054       MOVL    R0, STATUS                                        ;
                              0A           52  E8 00057       BLBS    STATUS, 2$                                        ; 0366
                                           52  DD 0005A       PUSHL   STATUS                                            ; 0369
                  00000000G   00           01  FB 0005C       CALLS   #1, LIB$SIGNAL                                    ;
                                               04 00063       RET                                                       ; 0368
                         08   AE       03  A7  9A 00064 2$:   MOVZBL  3(R7), LENGTH                                     ; 0379
                         08   AE           18  C0 00069       ADDL2   #24, LENGTH                                       ;
                                       04  AE  9F 0006D       PUSHAB  ENTRY                                             ; 0380
                                       0C  AE  9F 00070       PUSHAB  LENGTH                                            ;
                  00000000G   00           02  FB 00073       CALLS   #2, LIB$GET_VM                                    ;
                              52           50  D0 0007A       MOVL    R0, STATUS                                        ;
                              09           52  E8 0007D       BLBS    STATUS, 3$                                        ; 0381
                                           52  DD 00080       PUSHL   STATUS                                            ; 0383
                  00000000G   00           01  FB 00082       CALLS   #1, LIB$STOP                                      ;
                              56       04  AE  D0 00089 3$:   MOVL    ENTRY, R6                                         ; 0385
                              50       3B  A7  9A 0008D       MOVZBL  59(R7), R0                                        ;
                              51       3C  A7  9A 00091       MOVZBL  60(R7), R1                                        ;
             04   A6          50           51  C1 00095       ADDL3   R1, R0, 4(R6)                                     ;
                  50     4C   A7       04  A7  C3 0009A       SUBL3   4(R7), 76(R7), R0                                 ; 0387
                         08   A6      18 A046  9E 000A0       MOVAB   24(R0)[R6], 8(R6)                                 ;
                         0C   A6           6E  D0 000A6       MOVL    VERSION, 12(R6)                                   ; 0388
                         10   A6       03  A7  9A 000AA       MOVZBL  3(R7), 16(R6)                                     ; 0389
                         14   A6       18  A6  9E 000AF       MOVAB   24(R6), 20(R6)                                    ; 0390
                              50       03  A7  9A 000B4       MOVZBL  3(R7), R0                                         ; 0391
             18   A6     04   B7           50  28 000B8       MOVC3   R0, @4(R7), 24(R6)                                ;
                              57    0000'  CF  9E 000BE       MOVAB   VERSION_LIST, PREV                                ; 0393
                              54           67  D0 000C3       MOVL    (PREV), CURR                                      ; 0394
                                           28  13 000C6 4$:   BEQL    7$                                                ; 0396
                              55           01  D0 000C8       MOVL    #1, R5                                            ; 0400
04   A4           00     08   B6       04  A6  2D 000CB       CMPC5   4(R6), @8(R6), #0, 4(CURR), @8(CURR)              ;
```

```
                        08   B4    000D3                                          ;
                             03 1A 000D5         BGTRU   5$                       ;
            55               01 D9 000D7         SBWC    #1, R5                   ;
            50               55 D0 000DA 5$:     MOVL    R5, COMPARISON           ;
                             11 19 000DD         BLSS    7$                       ; 0402
                             07 12 000DF         BNEQ    6$                       ; 0403
      0C    A4          0C   A6 D1 000E1         CMPL    12(R6), 12(CURR)         ; 0404
                             08 1A 000E6         BGTRU   7$                       ;
            57               54 D0 000E8 6$:     MOVL    CURR, PREV               ; 0407
            54               64 D0 000EB         MOVL    (CURR), CURR             ; 0408
                             D6 11 000EE         BRB     4$                       ; 0396
            66               67 D0 000F0 7$:     MOVL    (PREV), (R6)             ; 0411
            67               56 D0 000F3         MOVL    R6, (PREV)               ; 0412
                                04 000F6         RET                              ; 0416

; Routine Size:  247 bytes,    Routine Base:  $CODE$ + 0074
```

```
  1 GLOBAL ROUTINE purge_ods1_directory (list) : NOVALUE =
  1
  1 !++
  1 ! Functional description:
  1 !
  1 !       This routine purges all versions beyond the number explicitly kept for
  1 !       an ods-1 directory.
  1 !
  1 ! Inputs:
  1 !
  1 !       list = Address of listhead of version list
  1 !                       0) link to next entry
  1 !                       1-2) descriptor of file name & type
  1 !                       3) version number in binary
  1 !                       4-5) descriptor of filespec
  1 !                       6) File specification follows
  1 !
  1 ! Outputs:
  1 !
  1 !       None, the old versions are deleted.
  1 !---
  1
  2 BEGIN
  2
  2 BIND context = .del$context : BITVECTOR[32];
  2
  2 OWN
  2     nam:        $NAM(),                                 ! NAM used for deleting open files
  2     xabpro:     $XABPRO(),                              ! XAB needed for common qualifiers package
  2     xabdat:     $XABDAT(NXT = xabpro),                  ! XAB needed for common qualifiers package
P 2     fab:        $FAB(NAM = nam,                         ! FAB used for deleting versions
  2                     XAB = xabdat);                      ! Chain XAB's to FAB.
  2
  2 LOCAL
  2     status,
  2     prev_name:  VECTOR [2],                             ! Descriptor of previous name & type
  2     prev_buffer: VECTOR [nam$c_maxrss,BYTE],            ! Buffer to hold previous name/type
  2     length,                                             ! Length of current entry
  2     entry:      REF VECTOR,                             ! Current entry in list
  2     next;                                               ! Next entry in list
  2
  2
  2 prev_name [0] = 0;                                      ! Initialize previous name/type = null
  2 entry = ..list;                                         ! Start at first entry in list
  2
  2 WHILE .entry NEQ 0                                      ! For each entry in list,
  2 DO
  3     BEGIN
  3     IF CH$NEQ(.prev_name [0], .prev_name [1],
  3         .entry [1], .entry [2], 0)                      ! If new file name & type,
  3     THEN
  4         BEGIN
  4         versions = 0;                                   ! Reset number of versions seen
  4         context[0] = 0;                                 ! Prevent confirm messgae.
  4         prev_name [0] = .entry [1];                     ! Save "current" name & type
  4         prev_name [1] = prev_buffer;
  4         CH$MOVE(.entry [1], .entry [2], .prev_name [1]);
```

```
3           END;
3
3       fab [fab$b_fns] = .entry [4];                          ! Copy length and address
3       fab [fab$l_fna] = .entry [5];                          ! of string into FAB and
3       nam [nam$b_rsl] = .entry [4];                          ! NAM blocks.
3       nam [nam$l_rsa] = .entry [5];
3
3       purge_this_file(fab);
3
3   !
3   ! Delete the storage used for this version entry
3   !
3
3       next = .entry [0];                                     ! Save address of next one
3       length = 6*4 + .entry [4];                             ! Length of entry
3       status = LIB$FREE_VM(length,entry);                    ! Delete storage for entry
3       IF NOT .status                                         ! If error deleting storage,
3       THEN
3           SIGNAL(.status);                                   ! then only signal the error
3       entry = .next;                                         ! Link to next in list
2       END;
2
2   .list = 0;                                                 ! Re-init listhead
2
1   END;


                                        .PSECT  $OWN$,NOEXE,2

                  0020B                 .BLKB   1
              02  0020C  NAM:           .BYTE   2
              60  0020D                 .BYTE   96
              00  0020E                 .BYTE   0
              00  0020F                 .BYTE   0
        00000000  00210                 .LONG   0
              00  00214                 .BYTE   0
              00  00215                 .BYTE   0
              00  00216                 .BYTE   0
              00  00217                 .BYTE   0
        00000000  00218                 .LONG   0
        00000000  0021C                 .LONG   0
           0000#  00220                 .WORD   0[8]
           0000#  00230                 .WORD   0[3]
           0000#  00236                 .WORD   0[3]
        00000000  0023C                 .LONG   0
        00000000  00240                 .LONG   0
              00  00244                 .BYTE   0
              00  00245                 .BYTE   0
              00  00246                 .BYTE   0
              00  00247                 .BYTE   0
              00  00248                 .BYTE   0
              00  00249                 .BYTE   0
             00#  0024A                 .BYTE   0[2]
        00000000  0024C                 .LONG   0
        00000000  00250                 .LONG   0
        00000000  00254                 .LONG   0
```

```
              00000000  00258            .LONG   0                                                  ;
              00000000  0025C            .LONG   0                                                  ;
              00000000  00260            .LONG   0                                                  ;
              00000000# 00264            .LONG   0[2]                                               ;
                    13  0026C  XABPRO:   .BYTE   19                                                 ;
                    58  0026D            .BYTE   88                                                 ;
                  0000  0026E            .WORD   0                                                  ;
              00000000  00270            .LONG   0                                                  ;
                  FFFF  00274            .WORD   -1                                                 ;
                    00  00276            .BYTE   0                                                  ;
                    00  00277            .BYTE   0                                                  ;
             0000  0000  00278            .WORD   0, 0                                              ;
                    00  0027C            .BYTE   0                                                  ;
                    00  0027D            .BYTE   0                                                  ;
                  0000  0027E            .WORD   0                                                  ;
              00000000  00280            .LONG   0                                                  ;
              00000000  00284            .LONG   0                                                  ;
                  0000  00288            .WORD   0                                                  ;
                  0000  0028A            .WORD   0                                                  ;
              00000000  0028C            .LONG   0                                                  ;
              00000000  00290            .LONG   0                                                  ;
                        00294            .BLKB   48                                                 ;
                    12  002C4  XABDAT:   .BYTE   18                                                 ;
                    2C  002C5            .BYTE   44                                                 ;
                  0000  002C6            .WORD   0                                                  ;
              00000000' 002C8            .ADDRESS XABPRO                                            ;
                  0000  002CC            .WORD   0                                                  ;
                  0000  002CE            .WORD   0                                                  ;
              00000000# 002D0            .LONG   0[2]                                               ;
              00000000# 002D8            .LONG   0[2]                                               ;
              00000000  002E0            .LONG   0                                                  ;
              00000000  002E4            .LONG   0                                                  ;
              00000000# 002E8            .LONG   0[2]                                               ;
                    03  002F0  FAB:      .BYTE   3                                                  ;
                    50  002F1            .BYTE   80                                                 ;
                  0000  002F2            .WORD   0                                                  ;
              00000000  002F4            .LONG   0                                                  ;
              00000000  002F8            .LONG   0                                                  ;
              00000000  002FC            .LONG   0                                                  ;
              00000000  00300            .LONG   0                                                  ;
                  0000  00304            .WORD   0                                                  ;
                    02  00306            .BYTE   2                                                  ;
                    00  00307            .BYTE   0                                                  ;
              00000000  00308            .LONG   0                                                  ;
                    00  0030C            .BYTE   0                                                  ;
                    00  0030D            .BYTE   0                                                  ;
                    00  0030E            .BYTE   0                                                  ;
                    02  0030F            .BYTE   2                                                  ;
              00000000  00310            .LONG   0                                                  ;
              00000000' 00314            .ADDRESS XABDAT                                            ;
              00000000' 00318            .ADDRESS NAM                                               ;
              00000000  0031C            .LONG   0                                                  ;
              00000000  00320            .LONG   0                                                  ;
                    00  00324            .BYTE   0                                                  ;
                    00  00325            .BYTE   0                                                  ;
                  0000  00326            .WORD   0                                                  ;
              00000000  00328            .LONG   0                                                  ;
```

```
                                         0000  0032C             .WORD   0                                                  ;
                                           00  0032E             .BYTE   0                                                  ;
                                           00  0032F             .BYTE   0                                                  ;
                                     00000000  00330             .LONG   0                                                  ;
                                     00000000  00334             .LONG   0                                                  ;
                                         0000  00338             .WORD   0                                                  ;
                                           00  0033A             .BYTE   0                                                  ;
                                           00  0033B             .BYTE   0                                                  ;
                                     00000000  0033C             .LONG   0                                                  ;



                                                                 .PSECT  $CODE$,NOWRT,2

                                         03FC  00000             .ENTRY  PURGE_ODS1_DIRECTORY, Save R2,R3,R4,R5,R6,-       ; 0417
                                                                         R7,R8,R9
                       5E        FEF4    CE  9E  00002           MOVAB   -268(SP), SP                                      ;
                       59   00000000G    00  D0  00007           MOVL    DEL$CONTEXT, R9                                   ; 0441
                                   F8    AD  D4  0000E           CLRL    PREV_NAME                                         ; 0459
                                   04    BC  DD  00011           PUSHL   @LIST                                             ; 0460
                       56              6E  D0  00014 1$:         MOVL    ENTRY, R6                                         ; 0462
                                       6E  13  00017             BEQL    4$                                                ;
  04    A6        00       FC      BD    F8    AD  2D  00019     CMPC5   PREV_NAME, @PREV_NAME+4, #0, 4(R6), @8(R6)        ; 0465
                                         08    B6      00021
                                         18    13  00023         BEQL    2$                                                ;
                            0000'    CF    D4  00025             CLRL    VERSIONS                                          ; 0469
                       69              01  8A  00029             BICB2   #1, (R9)                                          ; 0470
                       F8    AD    04    A6  D0  0002C           MOVL    4(R6), PREV_NAME                                  ; 0471
                       FC    AD    08    AE  9E  00031           MOVAB   PREV_BUFFER, PREV_NAME+4                          ; 0472
          FC    BD    08    B6    04    A6  28  00036            MOVC3   4(R6), @8(R6), @PREV_NAME+4                       ; 0473
                  0000'    CF    10    A6  90  0003D 2$:         MOVB    16(R6), FAB+52                                    ; 0476
                  0000'    CF    14    A6  D0  00043             MOVL    20(R6), FAB+44                                    ; 0477
                  0000'    CF    10    A6  90  00049             MOVB    16(R6), NAM+3                                     ; 0478
                  0000'    CF    14    A6  D0  0004F             MOVL    20(R6), NAM+4                                     ; 0479
                            0000'    CF    9F  00055             PUSHAB  FAB                                               ; 0481
                  0000V    CF          01  FB  00059             CALLS   #1, PURGE_THIS_FILE                               ;
                           57          66  D0  0005E             MOVL    (R6), NEXT                                        ; 0487
          04    AE    10    A6          18  C1  00061            ADDL3   #24, 16(R6), LENGTH                               ; 0488
                                        5E  DD  00067            PUSHL   SP                                                ; 0489
                                 08    AE  9F  00069             PUSHAB  LENGTH                                            ;
              00000000G    00          02  FB  0006C             CALLS   #2, LIB$FREE_VM                                   ;
                           58          50  D0  00073             MOVL    R0, STATUS                                        ;
                           09          58  E8  00076             BLBS    STATUS, 3$                                        ; 0490
                                        58  DD  00079            PUSHL   STATUS                                            ; 0492
              00000000G    00          01  FB  0007B             CALLS   #1, LIB$SIGNAL                                    ;
                           6E          57  D0  00082 3$:         MOVL    NEXT, ENTRY                                       ; 0493
                                       8D  11  00085             BRB     1$                                                ; 0462
                                 04    BC  D4  00087 4$:         CLRL    @LIST                                             ; 0496
                                           04  0008A             RET                                                       ; 0498

; Routine Size:  139 bytes,    Routine Base:  $CODE$ + 016B
;  399      0499  1 ROUTINE purge_ods2_files (fab_block) : NOVALUE =
;  400      0500  1
;  401      0501  1 !++
;  402      0502  1 ! Functional description:
;  403      0503  1 !
;  404      0504  1 !       This routine purges all versions beyond the number explicitly kept for
;  405      0505  1 !       an ods-2 disk.
;  406      0506  1 !
;  407      0507  1 ! Inputs:
;  408      0508  1 !
;  409      0509  1 !       fab_block = Address of FAB
;  410      0510  1 !
;  411      0511  1 ! Outputs:
;  412      0512  1 !
;  413      0513  1 !       None, the old versions are deleted.
;  414      0514  1 !---
;  415      0515  1
;  416      0516  2 BEGIN
;  417      0517  2
;  418      0518  2 MAP
;  419      0519  2     fab_block : REF $BBLOCK;                              ! Address of FAB
;  420      0520  2
;  421      0521  2 BIND
;  422      0522  2     context = .del$context : BITVECTOR[32],               ! Context bitmap for common qualifier package.
;  423      0523  2     nam = .fab_block [fab$l_nam] : $BBLOCK;               ! Address of NAM block.
;  424      0524  2
;  425      0525  2 LOCAL
;  426      0526  2     name_len;                                             ! Size of a filename.
;  427      0527  2
;  428      0528  2
;  429      0529  2 name_len = .nam [nam$b_node] + .nam [nam$b_dev] + .nam [nam$b_dir] +
;  430      0530  2           .nam [nam$b_name] + .nam[nam$b_type];
;  431      0531  2
;  432      0532  2 IF CH$NEQ(                                                ! If new device, directory, name or type
;  433      0533  2         .prev_name_len, prev_name,
;  434      0534  2         .name_len, .nam [nam$l_rsa], 0)
;  435      0535  2 THEN                                                      ! Then
;  436      0536  3     BEGIN
;  437      0537  3     CH$MOVE(.name_len, .nam [nam$l_rsa], prev_name);      ! Save new file spec,
;  438      0538  3     prev_name_len = .name_len;                            !  and its size.
;  439      0539  3     versions = 1;                                         ! Reset the version count.
;  440      0540  3     context[0] = 0;                                       ! Disable /CONFIRM.
;  441      0541  3     RETURN;
;  442      0542  3     END
;  443      0543  2 ELSE
;  444      0544  2     purge_this_file(.fab_block);                          ! Attempt to purge this file.
;  445      0545  2
;  446      0546  1 END;



                                        01FC 00000 PURGE_ODS2_FILES:
                                                         .WORD   Save R2,R3,R4,R5,R6,R7,R8                       ; 0499
                        58      0000'  CF   9E 00002     MOVAB   PREV_NAME_LEN, R8
                        57 00000000G   00   D0 00007     MOVL    DEL$CONTEXT, R7                                 ; 0522
```

```
                              55       04   AC  D0 0000E          MOVL    FAB_BLOCK, R5                              ; 0523
                              54       28   A5  D0 00012          MOVL    40(R5), R4                                 ;
                              50       38   A4  9A 00016          MOVZBL  56(R4), R0                                 ; 0529
                              51       39   A4  9A 0001A          MOVZBL  57(R4), R1                                 ;
                              50            51  C0 0001E          ADDL2   R1, R0                                     ;
                              51       3A   A4  9A 00021          MOVZBL  58(R4), R1                                 ;
                              50            51  C0 00025          ADDL2   R1, R0                                     ;
                              51       3B   A4  9A 00028          MOVZBL  59(R4), R1                                 ; 0530
                              50            51  C0 0002C          ADDL2   R1, R0                                     ; 0529
                              56       3C   A4  9A 0002F          MOVZBL  60(R4), NAME_LEN                           ; 0530
                              56            50  C0 00033          ADDL2   R0, NAME_LEN                               ;
     56        00       04    A8            68  2D 00036          CMPC5   PREV_NAME_LEN, PREV_NAME, #0, NAME_LEN, -  ; 0532
                                       04   B4     0003C                  @4(R4)                                     ;
                                            11  13 0003E          BEQL    1$                                         ;
          04   A8       04    B4            56  28 00040          MOVC3   NAME_LEN, @4(R4), PREV_NAME                ; 0537
                              68            56  D0 00046          MOVL    NAME_LEN, PREV_NAME_LEN                    ; 0538
                        FC    A8            01  D0 00049          MOVL    #1, VERSIONS                               ; 0539
                              67            01  8A 0004D          BICB2   #1, (R7)                                   ; 0540
                                                04 00050          RET                                                ; 0536
                                            55  DD 00051 1$:      PUSHL   R5                                         ; 0544
                     0000V    CF            01  FB 00053          CALLS   #1, PURGE_THIS_FILE                        ;
                                                04 00058          RET                                                ; 0546

; Routine Size:  89 bytes,    Routine Base:  $CODE$ + 01F6
```

```
1 ROUTINE purge_this_file (fab_block) : NOVALUE =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine accepts a filename, and if it exceeds the /KEEP qualifier,
1 !       attempts to delete the file.
1 !
1 ! Inputs:
1 !
1 !       fab_block  = Address of the FAB
1 !
1 ! Outputs:
1 !
1 !       None, the file is deleted if /KEEP qualifier is exceeded.
1 !---
1
2 BEGIN
2
2 MAP
2     fab_block :                     REF $BBLOCK;
2
2 BIND
2     context = .del$context :                BITVECTOR[32],
2     nam = .fab_block [fab$l_nam] :          $BBLOCK,
2     fab = .fab_block :                      $BBLOCK;
2
2 LOCAL
2     prompt_desc,                                             ! Address of prompt arguments
2     name_desc : VECTOR[2],                                   ! Descriptor for file name.
2     status;
2
2 name_desc[0] = .nam[nam$b_rsl];                              ! Save file name and size.
2 name_desc[1] = .nam[nam$l_rsa];
2 prompt_desc = name_desc;                                     ! Point prompt_desc to it.
2
2
2 IF .del$cli_status [del$v_log_msg] OR                        ! If /LOG requested,
2    .del$cli_status [del$v_open_file]                         !  or open_file bit set
2 THEN
3     BEGIN
3     fab [fab$l_alq] = 0;                                     ! Set block size of file to be zero because
3     $OPEN (FAB = fab);                                       ! Open the file.
3     del$file_size = .fab [fab$l_alq];                        ! Get block size of file.
3     END;
2
2 IF .del$cli_status[del$v_conf_prompt]                        ! If user said /CONFIRM, and
3   AND (.versions EQL .del$keepver_val)                       !  we have exceeded /KEEP limit
2     THEN context[0] = 1;                                     !  then enable /CONFIRM option.
2
2 status = lib$qual_file_match( del$context,fab,0,             ! Does this file meet purge criteria?
2                     $descriptor('!AS, delete? [N]:'),
2                     prompt_desc,0);
2
2 IF NOT .status                                               ! If failure status returned,
2 THEN
3     BEGIN
```

```
;  505       0604 3         IF .status EQL lib$_quipro                            ! If user said CNTRL/Z
;  506       0605 3         THEN                                                  !  then stop processing.
;  507       0606 3             del$cli_status [del$v_cntrl_z_stop] = TRUE;
;  508       0607 3         IF (.status NEQ lib$_quipro) AND                      ! If user said CNTRL/Z
;  509       0608 4            (.status NEQ lib$_filfaimat)                       !  or file did not meet criteria
;  510       0609 3         THEN                                                  !  then do not report an error.
;  511       0610 3             del$file_error(msg$_filnotacc,fab);
;  512       0611 3         $CLOSE ( FAB = fab);                                  ! Ask RMS to close the file.
;  513       0612 3         END
;  514       0613 2     ELSE
;  515       0614 2         versions = .versions + 1;                             ! Increment versions seen
;  516       0615 2
;  517       0616 2
;  518       0617 2     IF (.versions GTR .del$keepver_val) AND .status       ! If past specified limit,
;  519       0618 2     THEN                                                  !  then delete the file as we have exhausted
;  520       0619 3         BEGIN                                             !  the keep version count.
;  521       0620 3         IF .del$cli_status [del$v_erase]                  ! If /ERASE requested
;  522       0621 3         THEN
;  523       0622 4             BEGIN
;  524       0623 4             $CLOSE (FAB = fab);                           ! Close the file so that we cn do it.
;  525       0624 4             status = lib$set_erase (name_desc);           ! Set ERASE bit in header.
;  526       0625 4             fab[fab$l_sts] = .status;                     !  and save the status.
;  527       0626 4             fab[fab$l_stv] = 0;
;  528       0627 3             END;
;  529       0628 3
;  530       0629 3         IF .status                                        ! If successful so far,
;  531       0630 3         THEN
;  532       0631 4             BEGIN
;  533       0632 4             IF .fab [fab$w_ifi] NEQ 0                     ! If the file is open,
;  534       0633 4             THEN
;  535       0634 5                 BEGIN
;  536       0635 5                 fab [fab$v_dlt] = TRUE;                   !  then set the deletion bit,
;  537       0636 5                 status = $CLOSE ( FAB = fab);             !  and ask RMS to close and delete the file.
;  538       0637 5                 fab [fab$v_dlt] = FALSE;                  ! Turn off the delete bit to avoide side effects.
;  539       0638 5                 END
;  540       0639 4             ELSE
;  541       0640 4                 status = $ERASE ( FAB = fab);             ! Erase the file.
;  542       0641 3             END;
;  543       0642 3
;  544       0643 3         IF .status                                        ! If successful,
;  545       0644 3         THEN
;  546       0645 4             BEGIN
;  547       0646 4             IF .del$cli_status [del$v_log_msg] THEN       ! If logging requested,
;  548       0647 5                 BEGIN
;  549       0648 5                 del$files_deleted = .del$files_deleted + 1; ! Increment number of files purged
;  550       0649 5                 del$blocks_deleted = .del$blocks_deleted + .del$file_size; ! Increment total blocks deleted by t
;  551     P 0650 5                 put_message(msg$_filpurg,3,name_desc,     ! Output log message
;  552       0651 5                             .del$file_size);
;  553       0652 4                 END;
;  554       0653 3             END
;  555       0654 3         ELSE
;  556       0655 3             del$file_error(msg$_filnotpur,fab)            ! Delete failed, output message giving reason
;  557       0656 2         END;
;  558       0657 2
;  559       0658 2     $CLOSE (FAB=fab);                                     ! Make sure file is closed!
;  560       0659 2
;  561       0660 1     END;
```

```
                                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

4E 5B 20 3F 65 74 65 6C 65 64 20 2C 53 41 21 00000 P.AAB:   .ASCII  \!AS, delete? [N]:\                     ;
                                       3A 5D 0000F                                                             ;
                                             00011          .BLKB   3
                                  00000011   00014 P.AAA:   .LONG   17                                         ;
                                  00000000'  00018          .ADDRESS P.AAB                                     ;

                                                            .EXTRN  SYS$OPEN, SYS$CLOSE
                                                            .EXTRN  SYS$ERASE

                                                            .PSECT  $CODE$,NOWRT,2

                                       OFFC 00000 PURGE_THIS_FILE:
                                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      ; 0547
                   5B       0000'   CF   9E 00002           MOVAB   VERSIONS, R11                              ;
                   5A   00000000G   00   9E 00007           MOVAB   DEL$CONTEXT, R10                           ;
                   59   00000000G   00   9E 0000E           MOVAB   DEL$FILE_ERROR, R9                         ;
                   58   00000000G   00   9E 00015           MOVAB   LIB$_QUIPRO, R8                            ;
                   57   00000000G   00   9E 0001C           MOVAB   DEL$KEEPVER_VAL, R7                        ;
                   56   00000000G   00   9E 00023           MOVAB   DEL$FILE_SIZE, R6                          ;
                   55   00000000G   00   9E 0002A           MOVAB   SYS$CLOSE, R5                              ;
                   54   00000000G   00   9E 00031           MOVAB   DEL$CLI_STATUS, R4                         ;
                   5E               04   C2 00038           SUBL2   #4, SP                                     ;
                   53               6A   D0 0003B           MOVL    DEL$CONTEXT, R3                            ; 0570
                   52          04   AC   D0 0003E           MOVL    FAB_BLOCK, R2                              ; 0571
                   50          28   A2   D0 00042           MOVL    40(R2), R0                                 ;
                   7E          03   A0   9A 00046           MOVZBL  3(R0), NAME_DESC                           ; 0579
            04     AE          04   A0   D0 0004A           MOVL    4(R0), NAME_DESC+4                         ; 0580
                                    5E   DD 0004F           PUSHL   SP                                         ; 0581
      04           64               01   E0 00051           BBS     #1, DEL$CLI_STATUS, 1$                     ; 0584
                                    64   95 00055           TSTB    DEL$CLI_STATUS                             ; 0585
                                    10   18 00057           BGEQ    2$                                         ;
                               10   A2   D4 00059 1$:       CLRL    16(R2)                                     ; 0588
                                    52   DD 0005C           PUSHL   R2                                         ; 0589
         00000000G 00               01   FB 0005E           CALLS   #1, SYS$OPEN                               ;
                   66          10   A2   D0 00065           MOVL    16(R2), DEL$FILE_SIZE                      ; 0590
      08           64               06   E1 00069 2$:       BBC     #6, DEL$CLI_STATUS, 3$                     ; 0593
                   67               6B   D1 0006D           CMPL    VERSIONS, DEL$KEEPVER_VAL                  ; 0594
                                    03   12 00070           BNEQ    3$                                         ;
                   63               01   88 00072           BISB2   #1, (R3)                                   ; 0595
                                    7E   D4 00075 3$:       CLRL    -(SP)                                      ; 0597
                               04   AE   9F 00077           PUSHAB  PROMPT_DESC                                ;
                            0000'   CF   9F 0007A           PUSHAB  P.AAA                                      ; 0598
                                    7E   D4 0007E           CLRL    -(SP)                                      ; 0597
                                    52   DD 00080           PUSHL   R2                                         ;
                                    5A   DD 00082           PUSHL   R10                                        ;
         00000000G 00               06   FB 00084           CALLS   #6, LIB$QUAL_FILE_MATCH                    ;
                   53               50   D0 0008B           MOVL    R0, STATUS                                 ;
                   31               53   E8 0008E           BLBS    STATUS, 6$                                 ; 0601
                   50               68   9E 00091           MOVAB   LIB$_QUIPRO, R0                            ; 0604
                   50               53   D1 00094           CMPL    STATUS, R0                                 ;
                                    03   12 00097           BNEQ    4$                                         ;
                   64               20   88 00099           BISB2   #32, DEL$CLI_STATUS                        ; 0606
```

```
                                                          C 12
PURGE                                                     15-Sep-1984 23:39:44    VAX-11 Bliss-32 V4.0-742              Page  20
V04-000                                                   14-Sep-1984 12:18:48    DISK$VMSMASTER:[DELETE.SRC]PURGE.B32;1   (7)

                 50                68 9E 0009C 4$:  MOVAB   LIB$_QUIPRO, R0                  ; 0607
                 50                53 D1 0009F      CMPL    STATUS, R0                       ;
                                   17 13 000A2      BEQL    5$                               ;
                 50   00000000G    00 9E 000A4      MOVAB   LIB$_FILFAIMAT, R0               ; 0608
                 50                53 D1 000AB      CMPL    STATUS, R0                       ;
                                   0B 13 000AE      BEQL    5$                               ;
                                   52 DD 000B0      PUSHL   R2                               ; 0610
                      00931338     8F DD 000B2      PUSHL   #9638712                         ;
                 69                02 FB 000B8      CALLS   #2, DEL$FILE_ERROR               ;
                                   52 DD 000BB 5$:  PUSHL   R2                               ; 0611
                 65                01 FB 000BD      CALLS   #1, SYS$CLOSE                    ;
                                   02 11 000C0      BRB     7$                               ; 0601
                                   6B D6 000C2 6$:  INCL    VERSIONS                         ; 0614
                 67                6B D1 000C4 7$:  CMPL    VERSIONS, DEL$KEEPVER_VAL        ; 0617
                                   73 15 000C7      BLEQ    11$                              ;
                 7D                53 E9 000C9      BLBC    STATUS, 13$                      ;
  19             64                04 E1 000CC      BBC     #4, DEL$CLI_STATUS, 8$           ; 0620
                                   52 DD 000D0      PUSHL   R2                               ; 0623
                 65                01 FB 000D2      CALLS   #1, SYS$CLOSE                    ;
                            04     AE 9F 000D5      PUSHAB  NAME_DESC                        ; 0624
      00000000G  00                01 FB 000D8      CALLS   #1, LIB$SET_ERASE                ;
                 53                50 D0 000DF      MOVL    R0, STATUS                       ;
            08   A2                53 D0 000E2      MOVL    STATUS, 8(R2)                    ; 0625
                            0C     A2 D4 000E6      CLRL    12(R2)                           ; 0626
                 52                53 E9 000E9 8$:  BLBC    STATUS, 12$                      ; 0629
                            02     A2 B5 000EC      TSTW    2(R2)                            ; 0632
                                   14 13 000EF      BEQL    9$                               ;
            05   A2         80     8F 88 000F1      BISB2   #128, 5(R2)                      ; 0635
                                   52 DD 000F6      PUSHL   R2                               ; 0636
                 65                01 FB 000F8      CALLS   #1, SYS$CLOSE                    ;
                 53                50 D0 000FB      MOVL    R0, STATUS                       ;
            05   A2         80     8F 8A 000FE      BICB2   #128, 5(R2)                      ; 0637
                                   0C 11 00103      BRB     10$                              ; 0632
                                   52 DD 00105 9$:  PUSHL   R2                               ; 0640
      00000000G  00                01 FB 00107      CALLS   #1, SYS$ERASE                    ;
                 53                50 D0 0010E      MOVL    R0, STATUS                       ;
                 2A                53 E9 00111 10$: BLBC    STATUS, 12$                      ; 0643
  31             64                01 E1 00114      BBC     #1, DEL$CLI_STATUS, 13$          ; 0646
                      00000000G    00 D6 00118      INCL    DEL$FILES_DELETED                ; 0648
                 50                66 D0 0011E      MOVL    DEL$FILE_SIZE, R0                ; 0649
      00000000G  00                50 C0 00121      ADDL2   R0, DEL$BLOCKS_DELETED           ;
                                   50 DD 00128      PUSHL   R0                               ; 0651
                            08     AE 9F 0012A      PUSHAB  NAME_DESC                        ;
                                   03 DD 0012D      PUSHL   #3                               ;
                      0093131B     8F DD 0012F      PUSHL   #9638683                         ;
      00000000G  00                04 FB 00135      CALLS   #4, LIB$SIGNAL                   ;
                                   0B 11 0013C 11$: BRB     13$                              ; 0643
                                   52 DD 0013E 12$: PUSHL   R2                               ; 0655
                      00931230     8F DD 00140      PUSHL   #9638448                         ;
                 69                02 FB 00146      CALLS   #2, DEL$FILE_ERROR               ;
                                   52 DD 00149 13$: PUSHL   R2                               ; 0658
                 65                01 FB 0014B      CALLS   #1, SYS$CLOSE                    ;
                                      04 0014E      RET                                      ; 0660

; Routine Size:  335 bytes,    Routine Base:  $CODE$ + 024F
;  563          0661  1 END
;  564          0662  0 ELUDOM



                                                                     .EXTRN  LIB$SIGNAL, LIB$STOP

;                              PSECT SUMMARY
;
;       Name                        Bytes                    Attributes
;
;   $GLOBAL$                           4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                            832  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                           926  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                            28  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                         Library Statistics
;
;                                            -------- Symbols --------        Pages        Processing
;       File                                 Total    Loaded   Percent       Mapped       Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1         9776       85         0          581          00:02.3





;                              COMMAND QUALIFIERS

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

;  565          0663  0
; Size:         926 code + 864 data bytes
; Run Time:        00:44.7
; Elapsed Time:    00:55.1
; Lines/CPU Min:       890
; Lexemes/CPU-Min: 12378
; Memory Used:  147 pages
; Compilation Complete
```

0101 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SSIK
LIS

DELTA
LIS

DELETE

DELEMAIN
LIS

SSITAB
LIS

DELETE
MAP

DELTA

DELTA
MAP

SSIU
LIS

SSIUW
LIS

PURGE
LIS

STRUCDEF
LIS

SSIDISP
LIS

S0DELTA
MAP

DELETE
REQ