```
DDDDDDDDDDDD      EEEEEEEEEEEEEEE  LLL              EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  EEEEEEEEEEEEEEE
DDDDDDDDDDDDD     EEEEEEEEEEEEEEE  LLL              EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  EEEEEEEEEEEEEEE
DDDDDDDDDDDDD     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  EEEEEEEEEEE      LLL              EEEEEEEEEEE            TTT        EEEEEEEEEEE
DDD          DDD  EEEEEEEEEEE      LLL              EEEEEEEEEEE            TTT        EEEEEEEEEEE
DDD          DDD  EEEEEEEEEEE      LLL              EEEEEEEEEEE            TTT        EEEEEEEEEEE
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  EEE              LLL              EEE                    TTT        EEE
DDDDDDDDDDDD      EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  EEEEEEEEEEEEEEE        TTT        EEEEEEEEEEEEEEE
DDDDDDDDDDDD      EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  EEEEEEEEEEEEEEE        TTT        EEEEEEEEEEEEEEE
DDDDDDDDDDDD      EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  EEEEEEEEEEEEEEE        TTT        EEEEEEEEEEEEEEE
```

```
PPPPPPPP    UU      UU  RRRRRRR      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  RRRRRRR      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
```

```
;     1          0001  0 MODULE purge (                    ! Purge directory program
;     2          0002  0                 IDENT = 'V04-000',
;     3          0003  0                 ADDRESSING_MODE(EXTERNAL=GENERAL)
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**************************************************************************
;     9          0009  1 !*                                                                        *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                   *
;    13          0013  1 !*                                                                        *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    19          0019  1 !*  TRANSFERRED.                                                           *
;    20          0020  1 !*                                                                        *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    23          0023  1 !*  CORPORATION.                                                           *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !*                                                                        *
;    29          0029  1 !**************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:  PURGE Command
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       This utility purges a directory, basically removing
;    37          0037  1 !       old versions of a specified group of files.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT:
;    40          0040  1 !
;    41          0041  1 !       VAX/VMS operating system. unprivileged user mode,
;    42          0042  1 !
;    43          0043  1 ! AUTHOR:  Tim Halvorsen,                        CREATION DATE: Oct-1979
;    44          0044  1 !
;    45          0045  1 ! Modified by:
;    46          0046  1 !
;    47          0047  1 !       V03-007 SHZ0009         Stephen H. Zalewksi,    15-Mar-1984
;    48          0048  1 !               Modify PURGE algorithm to make sticky searchlists work.
;    49          0049  1 !
;    50          0050  1 !       V03-006 SHZ0008         Stephen H. Zalewski,    21-Feb-1984
;    51          0051  1 !               Add support for sticky searchlists.
;    52          0052  1 !
;    53          0053  1 !       V03-005 SHZ0007         Stephen H. Zalewski,    27-Dec-1983
;    54          0054  1 !               Do defaulting of file name and type in module PURGE_FILES.
;    55          0055  1 !               Add performance enhancement that cuts down on the number
;    56          0056  1 !               of RMS $OPEN's and $CLOSE's that must be done to purge
;    57          0057  1 !               a file.
```

```
;    58          0058  1 ;
;    59          0059  1 ;       V03-004 SHZ0006         Stephen H. Zalewski     25-Feb-1983
;    60          0060  1 ;               If PURGE command was issued on ODS-2 disk, do not cache the
;    61          0061  1 ;               filenames before attempting to delete the files.
;    62          0062  1 ;
;    63          0063  1 ;       V03-003 SHZ0005         Stephen H. Zalewski,    4-Nov-1982  15:42
;    64          0064  1 ;               Modify PURGE to use common command qualifier package.
;    65          0065  1 ;
;    66          0066  1 ;       V03-002 SHZ0004         Stephen H. Zalewski,    26-Aug-1982  22:19
;    67          0067  1 ;               Fix bug in SHZ0003 that caused PURGE to ACCVIO if you purged
;    68          0068  1 ;               an empty directory.  Also fixed bug that caused "No files
;    69          0069  1 ;               purged" message to be printed even when files were purged.
;    70          0070  1 ;
;    71          0071  1 ;               Fix bug introduced in SHZ0003 that prevented dangling directory
;    72          0072  1 ;               entries from being deleted if the PURGE comand was issued
;    73          0073  1 ;               with /LOG qualifier.
;    74          0074  1 ;
;    75          0075  1 ;               Finally, if a file is opened because of /SINCE, /BEFORE or
;    76          0076  1 ;               /LOG qualifiers being present, leave it open until we actually
;    77          0077  1 ;               delete it to optimize the number of FAL jobs necessary to do
;    78          0078  1 ;               the job in case we are doing this over the net.
;    79          0079  1 ;
;    80          0080  1 ;       V03-001 SHZ0003         Stephen H. Zalewski,    10-Aug-1982  21:24
;    81          0081  1 ;               Clean up error handling.  Modified routines to use new CLI.
;    82          0082  1 ;               Made PURGE/LOG also display size of file purged.
;    83          0083  1 ;
;    84          0084  1 ;--
```

```
;    86          0085  1 LIBRARY 'SYS$LIBRARY:STARLET.L32';                    ! VAX/VMS common definitions
;    87          0086  1 REQUIRE 'SRC$:DELETE.REQ';                           ! Common DELETE definitions
;    88          0192  1
;    89          0193  1
;    90          0194  1 FORWARD ROUTINE
;    91          0195  1     purge_files,                                     ! Main directory routine
;    92          0196  1     build_list :            NOVALUE,                 ! Process each wildcard file for ods-1 disk
;    93          0197  1     purge_ods1_directory :      NOVALUE,             ! Purge versions for a directory
;    94          0198  1     purge_ods2_files :  NOVALUE,                     ! Purge files for ODS-2 disk.
;    95          0199  1     purge_this_file :   NOVALUE;                     ! Routine to delete a file.
;    96          0200  1
;    97          0201  1
;    98          0202  1 EXTERNAL ROUTINE
;    99          0203  1     del$search_error :  NOVALUE,                     ! Report open/search error.
;   100          0204  1     del$file_error :    NOVALUE,                     ! Report RMS error messages
;   101          0205  1     lib$file_scan,                                   ! Search wildcard specifications
;   102          0206  1     lib$set_erase,                                   ! Mark file for erase-on-delete
;   103          0207  1     lib$cvt_dtb,                                     ! Convert decimal string to binary
;   104          0208  1     lib$get_vm,                                      ! Allocate dynamic storage
;   105          0209  1     lib$free_vm,                                     ! Deallocate dynamic storage
;   106          0210  1     lib$qual_file_match;                             ! Check to see if file should be purged.
;   107          0211  1
;   108          0212  1
;   109          0213  1 EXTERNAL
;   110          0214  1     scan_context,
;   111          0215  1     del$cli_status:     $BBLOCK,                     ! CLI qualifier bitmap
;   112          0216  1     del$keepver_val,                                 ! Number of versions to save
;   113          0217  1     del$files_deleted,                               ! Number of files purged
;   114          0218  1     del$file_size,                                   ! Size of file being deleted
;   115          0219  1     del$blocks_deleted,                              ! Number of blocks released
;   116          0220  1     del$context,                                     ! Context longword for common qualifier package.
;   117          0221  1     lib$_quipro,                                     ! Return status code from common qualifier package.
;   118          0222  1     lib$_filfaimat;                                  ! Return status code from common qualifier package.
;   119          0223  1
;   120          0224  1
;   121          0225  1 GLOBAL
;   122          0226  1     version_list :      INITIAL(0);                  ! Address of linked list of versions
;   123          0227  1
;   124          0228  1 OWN
;   125          0229  1     versions :          INITIAL(0),                  ! Number of versions seen so far
;   126          0230  1     prev_name_len :     INITIAL(0),                  ! Length of previous name & type
;   127          0231  1     prev_name :         VECTOR [nam$c_maxrss,BYTE],  ! Buffer to hold previous name/type
;   128          0232  1     prev_dir_len :      INITIAL(0),                  ! Length of previous directory
;   129          0233  1     prev_dir :          VECTOR [nam$c_maxrss,BYTE];  ! Buffer for previous directory
;   130          0234  1
```

```
;  132          0235  1 GLOBAL ROUTINE purge_files (fab_block) =
;  133          0236  1
;  134          0237  1 !++
;  135          0238  1 ! Functional description
;  136          0239  1 !
;  137          0240  1 !       This routine performs all the main processing of the
;  138          0241  1 !       PURGE command.  The command line has already been parsed
;  139          0242  1 !       and the qualifier values saved.
;  140          0243  1 !
;  141          0244  1 ! Calling sequence
;  142          0245  1 !
;  143          0246  1 !       purge_files(fab) from the DELETE command mainline code
;  144          0247  1 !
;  145          0248  1 ! Input parameters
;  146          0249  1 !
;  147          0250  1 !       fab_block = Address of FAB with FNA, FNS filled in.
;  148          0251  1 !
;  149          0252  1 ! Output parameters
;  150          0253  1 !
;  151          0254  1 !       None
;  152          0255  1 !
;  153          0256  1 ! Routine value
;  154          0257  1 !
;  155          0258  1 !       Status.
;  156          0259  1 !----
;  157          0260  1
;  158          0261  2 BEGIN
;  159          0262  2
;  160          0263  2 MAP
;  161          0264  2     fab_block:  REF $BBLOCK;                        ! Address of FAB block
;  162          0265  2
;  163          0266  2 BIND
;  164          0267  2     nam_block = .fab_block [fab$l_nam]: $BBLOCK;    ! Address of NAM block
;  165          0268  2
;  166          0269  2 LOCAL
;  167          0270  2     status,
;  168          0271  2     devnam_desc : VECTOR[2],
;  169          0272  2     itmlst : VECTOR[4,LONG],
;  170          0273  2     buffer : INITIAL(0);
;  171          0274  2
;  172          0275  4 IF ((NOT .$BBLOCK[fab_block [fab$l_dev], dev$v_dir])    ! If not a directory device,
;  173          0276  3 AND (NOT .$BBLOCK[fab_block [fab$l_dev], dev$v_net]))   ! and not a network device
;  174          0277  2 THEN
;  175          0278  2     RETURN false;
;  176          0279  2
;  177          0280  3 IF (NOT .$BBLOCK[fab_block [fab$l_dev], dev$v_net])     ! If not a network device
;  178          0281  2 THEN                                                    !  then check to see if
;  179          0282  3     BEGIN                                               !  we are purging an ODS-2 disk.
;  180          0283  3     devnam_desc[0] = .nam_block[nam$b_dev];
;  181          0284  3     devnam_desc[1] = .nam_block[nam$l_dev];
;  182          0285  3     itmlst<0,16> = 4;
;  183          0286  3     itmlst<16,16> = dvi$_acptype;
;  184          0287  3     itmlst[1] = buffer;
;  185          0288  3     itmlst[2] = 0;
;  186          0289  3     itmlst[3] = 0;
;  187        P 0290  3     status = $GETDVIW (DEVNAM = devnam_desc,
;  188          0291  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
```

```
;  202     0304  1 ROUTINE build_list (fab_block): NOVALUE =
;  203     0305  1
;  204     0306  1 !---
;  205     0307  1 !
;  206     0308  1 ! Functional description
;  207     0309  1 !
;  208     0310  1 !       This routine is called as an action routine from the directory wildcard
;  209     0311  1 !       searching process.  It is given a FAB containing the full name
;  210     0312  1 !       of the file to be processed.  It is used when purging an ods-1 disk to
;  211     0313  1 !       build the filenames into an ods-2 list.
;  212     0314  1 !
;  213     0315  1 ! Input parameters
;  214     0316  1 !
;  215     0317  1 !       fab_block = Address of FAB
;  216     0318  1 !
;  217     0319  1 ! Output parameters
;  218     0320  1 !
;  219     0321  1 !       None
;  220     0322  1 !
;  221     0323  1 !----
;  222     0324  1
;  223     0325  2 BEGIN
;  224     0326  2
;  225     0327  2 MAP
;  226     0328  2     fab_block:  REF $BBLOCK;                        ! Address of FAB
;  227     0329  2
;  228     0330  2 BIND
;  229     0331  2     nam = .fab_block [fab$l_nam]: $BBLOCK;          ! Address of NAM
;  230     0332  2
;  231     0333  2 LOCAL
;  232     0334  2     status,                                         ! Status code
;  233     0335  2     dir_len,                                        ! Length of device/directory
;  234     0336  2     version,                                        ! Version number in binary form
;  235     0337  2     length,                                         ! Length of new version entry
;  236     0338  2     entry:      REF VECTOR,                         ! Address of new version entry
;  237     0339  2     prev:       REF VECTOR,                         ! Address of previous entry scanned
;  238     0340  2     curr:       REF VECTOR;                         ! Address of current entry scanned
;  239     0341  2
;  240     0342  2 !
;  241     0343  2 ! If we have reached a new directory, then peruse the linked
;  242     0344  2 ! list of versions and delete the all but the # explicitly kept.
;  243     0345  2 !
;  244     0346  2
;  245     0347  2 dir_len = .nam [nam$b_node] + .nam [nam$b_dev] + .nam [nam$b_dir];
;  246     0348  2
;  247     0349  2 IF CH$NEQ(                                          ! If new directory
;  248     0350  2         .prev_dir_len, prev_dir,
;  249     0351  2         .dir_len, .nam [nam$l_rsa], 0)
;  250     0352  2 THEN
;  251     0353  3     BEGIN
;  252     0354  3     CH$MOVE(.dir_len, .nam [nam$l_rsa], prev_dir);
;  253     0355  3     prev_dir_len = .dir_len;
;  254     0356  3     purge_ods1_directory(version_list);              ! Delete old versions
;  255     0357  2     END;
;  256     0358  2
;  257     0359  2 !
;  258     0360  2 ! Convert the version string to binary form
```

```
;  259          0361  2 !
;  260          0362  2
;  261          0363  2 status = LIB$CVT_DTB(.nam [nam$b_ver]-1,                 ! Store version in binary
;  262          0364  2                     .nam [nam$l_ver]+1,
;  263          0365  2                     version);
;  264          0366  2 IF NOT .status                                          ! If error converting value,
;  265          0367  2 THEN
;  266          0368  3     BEGIN
;  267          0369  3     SIGNAL(.status);                                    ! signal the error
;  268          0370  3     RETURN;
;  269          0371  2     END;
;  270          0372  2
;  271          0373  2 !
;  272          0374  2 ! Add this version to the ordered linked list of files for this directory.
;  273          0375  2 ! This list is ordered first by file name & type in ascending order, and
;  274          0376  2 ! then by version in descending order (this is the same as ODS-2).
;  275          0377  2 !
;  276          0378  2
;  277          037?  2 length = 6+4 + .nam [nam$b_rsl];                        ! Length of overhead plus filespec
;  278          0380  2 status = LIB$GET_VM(length,entry);                       ! Allocate storage for entry
;  279          0381  2 IF NOT .status                                          ! If error allocatino storage,
;  280          0382  2 THEN
;  281          0383  2     SIGNAL_STOP(.status);                               ! then signal the error
;  282          0384  2
;  283          0385  2 entry [1] = .nam [nam$b_name] + .nam [nam$b_type];       ! Length of name/type
;  284          0386  2 entry [2] = entry [6]                                   ! Address of name/type
;  285          0387  2           + (.nam [nam$l_name] - .nam [nam$l_rsa]);
;  286          0388  2 entry [3] = .version;                                   ! Store binary version number
;  287          0389  2 entry [4] = .nam [nam$b_rsl];                           ! Store length of filespec
;  288          0390  2 entry [5] = entry [6];                                  ! and address of filespec
;  289          0391  2 CH$MOVE(.nam [nam$b_rsl], .nam [nam$l_rsa], entry [6]); ! Store full filespec
;  290          0392  2
;  291          0393  2 prev = version_list;                                    ! Address of previous entry
;  292          0394  2 curr = .prev [0];                                       ! Start at first entry
;  293          0395  2
;  294          0396  2 WHILE .curr NEQ 0                                       ! For each entry in list,
;  295          0397  2 DO
;  296          0398  3     BEGIN
;  297          0399  3     LOCAL comparison;                                   ! -1 if less, 0 if equal, 1 if greater
;  298          0400  3     comparison = CH$COMPARE(.entry [1], .entry [2],
;  299          0401  3         .curr [1], .curr [2], 0);                       ! Compare new to old name/type
;  300          0402  3     IF .comparison LSS 0                                ! If found place to insert entry,
;  301          0403  4         OR (.comparison EQL 0                           ! (ascending order by name/type,
;  302          0404  4        AND .entry [3] GTRU .curr [3])                   ! (then descending order by version)
;  303          0405  3     THEN
;  304          0406  3         EXITLOOP;                                       ! then exit the loop
;  305          0407  3     prev = .curr;                                       ! Save address of previous entry done
;  306          0408  3     curr = .curr [0];                                   ! and link to next in list
;  307          0409  2     END;
;  308          0410  2
;  309          0411  2 entry [0] = .prev [0];                                  ! Make new entry point to next in list
;  310          0412  2 prev [0] = .entry;                                      ! Make previous entry point to new one
;  311          0413  2
;  312          0414  2 RETURN;
;  313          0415  2
;  314          0416  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 00029
                                                                    12  13 0002C         BEQL    1$
                        04      A8        04   B7                  56  28 0002E         MOVC3   DIR_LEN, @4(R7), PREV_DIR                  ; 0354
                                                                    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$STOP
                                                                         04    AE  D0 00089 3$:     MOVL    ENTRY, R6                  ; 0385
                                                                         3B    A7  9A 0008D         MOVZBL  59(R7), R0
                                                                         3C    A7  9A 00091         MOVZBL  60(R7), R1
                                                 04      A6                    50  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
```

```
;  316          0417  1 GLOBAL ROUTINE purge_ods1_directory (list) : NOVALUE =
;  317          0418  1
;  318          0419  1 !++
;  319          0420  1 ! Functional description:
;  320          0421  1 !
;  321          0422  1 !       This routine purges all versions beyond the number explicitly kept for
;  322          0423  1 !       an ods-1 directory.
;  323          0424  1 !
;  324          0425  1 ! Inputs:
;  325          0426  1 !
;  326          0427  1 !       list = Address of listhead of version list
;  327          0428  1 !                      0) link to next entry
;  328          0429  1 !                      1-2) descriptor of file name & type
;  329          0430  1 !                      3) version number in binary
;  330          0431  1 !                      4-5) descriptor of filespec
;  331          0432  1 !                      6) File specification follows
;  332          0433  1 !
;  333          0434  1 ! Outputs:
;  334          0435  1 !
;  335          0436  1 !       None, the old versions are deleted.
;  336          0437  1 !--
;  337          0438  1
;  338          0439  2 BEGIN
;  339          0440  2
;  340            41  2 BIND context = .del$context : BITVECTOR[32];
;  341          0442  2
;  342          0443  2 OWN
;  343          0444  2     nam:        $NAM(),                                 ! NAM used for deleting open files
;  344          0445  2     xabpro:     $XABPRO(),                              ! XAB needed for common qualifiers package
;  345          0446  2     xabdat:     $XABDAT(NXT = xabpro),                  ! XAB needed for common qualifiers package
;  346        P 0447  2     fab:        $FAB(NAM = nam,                         ! FAB used for deleting versions
;  347          0448  2                      XAB = xabdat);                    ! Chain XAB's to FAB.
;  348          0449  2
;  349          0450  2 LOCAL
;  350          0451  2     status,
;  351          0452  2     prev_name:  VECTOR [2],                             ! Descriptor of previous name & type
;  352          0453  2     prev_buffer: VECTOR [nam$c_maxrss,BYTE],            ! Buffer to hold previous name/type
;  353          0454  2     length,                                             ! Length of current entry
;  354          0455  2     entry:      REF VECTOR,                             ! Current entry in list
;  355          0456  2     next;                                               ! Next entry in list
;  356          0457  2
;  357          0458  2
;  358          0459  2 prev_name [0] = 0;                                      ! Initialize previous name/type = null
;  359          0460  2 entry = ..list;                                         ! Start at first entry in list
;  360          0461  2
;  361          0462  2 WHILE .entry NEQ 0                                      ! For each entry in list,
;  362          0463  2 DO
;  363          0464  3     BEGIN
;  364          0465  3     IF CH$NEQ(.prev_name [0], .prev_name [1],
;  365          0466  3         .entry [1], .entry [2], 0)                      ! If new file name & type,
;  366          0467  3     THEN
;  367          0468  4         BEGIN
;  368          0469  4         versions = 0;                                   ! Reset number of versions seen
;  369          0470  4         context[0] = 0;                                 ! Prevent confirm messgae.
;  370          0471  4         prev_name [0] = .entry [1];                     ! Save "current" name & type
;  371          0472  4         prev_name [1] = prev_buffer;
;  372          0473  4         CH$MOVE(.entry [1], .entry [2], .prev_name [1]);
```

```
;  373          0474  3          END;
;  374          0475  3
;  375          0476  3      fab [fab$b_fns] = .entry [4];                  ! Copy length and address
;  376          0477  3      fab [fab$l_fna] = .entry [5];                  ! of string into FAB and
;  377          0478  3      nam [nam$b_rsl] = .entry [4];                  ! NAM blocks.
;  378          0479  3      nam [nam$l_rsa] = .entry [5];
;  379          0480  3
;  380          0481  3      purge_this_file(fab);
;  381          0482  3
;  382          0483  3  !
;  383          0484  3  ! Delete the storage used for this version entry
;  384          0485  3  !
;  385          0486  3
;  386          0487  3      next = .entry [0];                             ! Save address of next one
;  387          0488  3      length = 6*4 + .entry [4];                     ! Length of entry
;  388          0489  3      status = LIB$FREE_VM(length,entry);            ! Delete storage for entry
;  389          0490  3      IF NOT .status                                 ! If error deleting storage,
;  390          0491  3      THEN
;  391          0492  3          SIGNAL(.status);                           ! then only signal the error
;  392          0493  3      entry = .next;                                 ! Link to next in list
;  393          0494  3      END;
;  394          0495  2
;  395          0496  2  .list = 0;                                         ! Re-init listhead
;  396          0497  2
;  397          0498  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$:     MOVB    16(R6), FAB+52                 ; 0476
                                0000'  CF         14   A6    D0 00043         MOVL    20(R6), FAB+44                 ; 0477
                                0000'  CF         10   A6    90 00049         MOVB    16(R6), NAM+3                  ; 0478
                                0000'  CF         14   A6    D0 0004F         MOVL    20(R6), NAM+4                  ; 0479
                                       CF    0000'  9F    9F 00055         PUSHAB  FAB                            ; 0481
                                0000V  CF             01    FB 00059         CALLS   #1, PURGE_THIS_FILE            ;
                                       57             66    D0 0005E         MOVL    (R6), NEXT                     ; 0487
                        04    AE        10   A6               18    C1 00061         ADDL3   #24, 16(R6), LENGTH    ; 0488
                                                              5E    DD 00067         PUSHL   SP                     ; 0489
                                                        08    AE    9F 00069         PUSHAB  LENGTH                 ;
                                00000000G   00               02    FB 0006C         CALLS   #2, LIB$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
```

```
;  448          0547  1 ROUTINE purge_this_file (fab_block) : NOVALUE =
;  449          0548  1
;  450          0549  1 !++
;  451          0550  1 ! Functional description:
;  452          0551  1 !
;  453          0552  1 !       This routine accepts a filename, and if it exceeds the /KEEP qualifier,
;  454          0553  1 !       attempts to delete the file.
;  455          0554  1 !
;  456          0555  1 ! Inputs:
;  457          0556  1 !
;  458          0557  1 !       fab_block  = Address of the FAB
;  459          0558  1 !
;  460          0559  1 ! Outputs:
;  461          0560  1 !
;  462          0561  1 !       None, the file is deleted if /KEEP qualifier is exceeded.
;  463          0562  1 !---
;  464          0563  1
;  465          0564  2 BEGIN
;  466          0565  2
;  467          0566  2 MAP
;  468          0567  2     fab_block :                 REF $BBLOCK;
;  469          0568  2
;  470          0569  2 BIND
;  471          0570  2     context = .del$context :            BITVECTOR[32],
;  472          0571  2     nam = .fab_block [fab$l_nam] :      $BBLOCK,
;  473          0572  2     fab = .fab_block :                  $BBLOCK;
;  474          0573  2
;  475          0574  2 LOCAL
;  476          0575  2     prompt_desc,                                   ! Address of prompt arguments
;  477          0576  2     name_desc : VECTOR[2],                         ! Descriptor for file name.
;  478          0577  2     status;
;  479          0578  2
;  480          0579  2 name_desc[0] = .nam[nam$b_rsl];                     ! Save file name and size.
;  481          0580  2 name_desc[1] = .nam[nam$l_rsa];
;  482          0581  2 prompt_desc = name_desc;                           ! Point prompt_desc to it.
;  483          0582  2
;  484          0583  2
;  485          0584  2 IF .del$cli_status [del$v_log_msg] OR               ! If /LOG requested,
;  486          0585  2    .del$cli_status [del$v_open_file]                !  or open_file bit set
;  487          0586  2 THEN
;  488          0587  3     BEGIN
;  489          0588  3     fab [fab$l_alq] = 0;                           ! Set block size of file to be zero because
;  490          0589  3     $OPEN (FAB = fab);                             ! Open the file.
;  491          0590  3     del$file_size = .fab [fab$l_alq];              ! Get block size of file.
;  492          0591  2     END;
;  493          0592  2
;  494          0593  2 IF .del$cli_status[del$v_conf_prompt]               ! If user said /CONFIRM, and
;  495          0594  3   AND (.versions EQL .del$keepver_val)             !  we have exceeded /KEEP limit
;  496          0595  2     THEN context[0] = 1;                           !  then enable /CONFIRM option.
;  497          0596  2
;  498          0597  2 status = lib$qual_file_match( del$context,fab,0,    ! Does this file meet purge criteria?
;  499          0598  2                             $descriptor('!AS, delete? [N]:'),
;  500          0599  2                             prompt_desc,0);
;  501          0600  2
;  502          0601  2 IF NOT .status                                      ! If failure status returned,
;  503          0602  2 THEN
;  504          0603  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  4          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

                                                 0FFC 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$:    CLRL    16(R2)                    ; 0588
                                                            52  DD 0005C       PUSHL   R2                        ; 0589
                                           00000000G        00  FB 0005E       CALLS   #1, SYS$OPEN              ;
                                                            66  D0 00065       MOVL    16(R2), DEL$FILE_SIZE     ; 0590
                                        08                  64  E1 00069 2$:    BBC     #6, DEL$CLI_STATUS, 3$    ; 0593
                                                            67  D1 0006D       CMPL    VERSIONS, DEL$KEEPVER_VAL ; 0594
                                                            03  12 00070       BNEQ    3$                        ;
                                                            63  88 00072       BISB2   #1, (R3)                  ; 0595
                                                            7E  D4 00075 3$:    CLRL    -(SP)                     ; 0597
                                                       04   AE  9F 00077       PUSHAB  PROMPT_DESC               ;
                                                     0000'  CF  9F 0007A       PUSHAB  P.AAA                     ; 0598
                                                            7E  D4 0007E       CLRL    -(SP)                     ; 0597
                                                            52  DD 00080       PUSHL   R2                        ;
                                                            5A  DD 00082       PUSHL   R10                       ;
                                           00000000G        00  FB 00084       CALLS   #6, LIB$QUAL_FILE_MATCH   ;
                                                            53  D0 00088       MOVL    R0, STATUS                ;
                                                            31  E8 0008E       BLBS    STATUS, 6$                ; 0601
                                                            50  9E 00091       MOVAB   LIB$_QUIPRO, R0           ; 0604
                                                            50  D1 00094       CMPL    STATUS, R0                ;
                                                            03  12 00097       BNEQ    4$                        ;
                                                            64  88 00099       BISB2   #32, DEL$CLI_STATUS       ; 0606
```

```
                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
```

SSIK
LIS

DELETE

DELEMAIN
LIS

DELTA
LIS

SSITAB
LIS

DELETE
MAP

DELTA

DELTA
MAP

SSIU
LIS

SSIUW
LIS

PURGE
LIS

STRUCDEF
LIS

SSIDISP
LIS

SØDELTA
MAP

DELETE
REQ