```
IIIIIIIII      NNN          NNN     SSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA     LLL
IIIIIIIII      NNN          NNN     SSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA     LLL
IIIIIIIII      NNN          NNN     SSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA     LLL
   III         NNN          NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNN          NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNN          NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNNNNN       NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNNNNN       NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNNNNN       NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNN    NNN   NNN     SSSSSSSSS          TTT          AAA         AAA  LLL
   III         NNN    NNN   NNN     SSSSSSSSS          TTT          AAA         AAA  LLL
   III         NNN    NNN   NNN     SSSSSSSSS          TTT          AAA         AAA  LLL
   III         NNN       NNNNNN               SSS      TTT          AAAAAAAAAAAAAAA  LLL
   III         NNN       NNNNNN               SSS      TTT          AAAAAAAAAAAAAAA  LLL
   III         NNN       NNNNNN               SSS      TTT          AAAAAAAAAAAAAAA  LLL
   III         NNN          NNN               SSS      TTT          AAA         AAA  LLL
   III         NNN          NNN               SSS      TTT          AAA         AAA  LLL
   III         NNN          NNN               SSS      TTT          AAA         AAA  LLL
IIIIIIIII      NNN          NNN  SSSSSSSSSSS           TTT          AAA         AAA  LLLLLLLLLLLLLLL
IIIIIIIII      NNN          NNN  SSSSSSSSSSS           TTT          AAA         AAA  LLLLLLLLLLLLLLL
IIIIIIIII      NNN          NNN  SSSSSSSSSSS           TTT          AAA         AAA  LLLLLLLLLLLLLLL
```

```
IIIIII    NN      NN    SSSSSSSS  LL          IIIIII     SSSSSSSS  TTTTTTTTTT
IIIIII    NN      NN    SSSSSSSS  LL          IIIIII     SSSSSSSS  TTTTTTTTTT
  II      NN      NN  SS          LL            II     SS              TT
  II      NN      NN  SS          LL            II     SS              TT
  II      NNNN    NN  SS          LL            II     SS              TT
  II      NNNN    NN  SS          LL            II     SS              TT
  II      NN  NN  NN    SSSSSS    LL            II       SSSSSS        TT
  II      NN  NN  NN    SSSSSS    LL            II       SSSSSS        TT
  II      NN    NNNN          SS  LL            II             SS      TT
  II      NN    NNNN          SS  LL            II             SS      TT
  II      NN      NN          SS  LL            II             SS      TT          ....
  II      NN      NN          SS  LL            II             SS      TT          ....
IIIIII    NN      NN  SSSSSSSS    LLLLLLLLLL  IIIIII  SSSSSSSS        TT          ....
IIIIII    NN      NN  SSSSSSSS    LLLLLLLLLL  IIIIII  SSSSSSSS        TT          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE INSLIST (                      ! Process /LIST and /FULL qualifiers
;     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:  Install
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       Print the contents of a KFE entry or of all the entries.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:
;    39          0039  1 !
;    40          0040  1 !       VAX/VMS operating system.
;    41          0041  1 !
;    42          0042  1 ! AUTHOR:  Bob Grosso, April 1983
;    43          0043  1 !
;    44          0044  1 ! Modified by:
;    45          0045  1 !
;    46          0046  1 !       V03-013 MSH0061         Michael S. Harvey       5-Jul-1984
;    47          0047  1 !               List EXECUTE_ONLY attribute if set for known image.
;    48          0048  1 !
;    49          0049  1 !       V03-012 MSH0057         Michael S. Harvey       26-Jun-1984
;    50          0050  1 !               List WRITEABLE attribute along with all the others.
;    51          0051  1 !
;    52          0052  1 !       V03-011 MSH0049         Michael S. Harvey       17-May-1984
;    53          0053  1 !               Don't output meaningless and inaccurate data for
;    54          0054  1 !               non-native mode installed images.
;    55          0055  1 !
;    56          0056  1 !       V03-010 MSH0037         Michael S. Harvey       26-Apr-1984
;    57          0057  1 !               Fall back to hard device name if no volume name
```

```
;    58          0058  1 !                 is available.
;    59          0059  1 !
;    60          0060  1 !       V03-009 MSH0034         Michael S. Harvey       18-Apr-1984
;    61          0061  1 !                 Display raw device name string in KFD for /STRUCTURE
;    62          0062  1 !                 listing so we can see what's really stored there.
;    63          0063  1 !
;    64          0064  1 !       V03-008 MSH0028         Michael S. Harvey       9-Apr-1984
;    65          0065  1 !                 List maximum shared count correctly. Also, display
;    66          0066  1 !                 global section count to help interpret the other
;    67          0067  1 !                 counts being displayed on /FULL listings.
;    68          0068  1 !
;    69          0069  1 !       V03-007 MSH0026         Michael S. Harvey       4-Apr-1984
;    70          0070  1 !                 Recognize when known file database either doesn't
;    71          0071  1 !                 exist or is empty, and do the right thing when one
;    72          0072  1 !                 tries to access it.
;    73          0073  1 !
;    74          0074  1 !       V03-006 MSH0022         Michael S. Harvey       20-Mar-1984
;    75          0075  1 !                 Convert unconcealed device name, which may be an
;    76          0076  1 !                 allocation class type device name, into a form that
;    77          0077  1 !                 does not have the allocation class in it.
;    78          0078  1 !
;    79          0079  1 !       V03-005 BLS0256         Benn Schreiber          27-Dec-1983
;    80          0080  1 !                 Clean up buffer handling.  Reference all pool from EXEC
;    81          0081  1 !                 mode, since protected against user mode.
;    82          0082  1 !                                                                        ;
;    83          0083  1 !       V03-004 RPG0004         Bob Grosso              13-Sep-1983
;    84          0084  1 !                 List WCB info.
;    85          0085  1 !                 Trim blanks from end of line.                          ;
;    86          0086  1 !
;    87          0087  1 !       V03-003 RPG0003         Bob Grosso              July 20, 1983
;    88          0088  1 !                 Clean up listing format.
;    89          0089  1 !                 Add /structure listing.
;    90          0090  1 !                 Print listing from user mode.
;    91          0091  1 !
;    92          0092  1 !       V03-002 RPG0002         Bob Grosso              July 8, 1983
;    93          0093  1 !                 Bypass printing WCB info.
;    94          0094  1 !
;    95          0095  1 !       V03-001 RPG0001         Bob Grosso              July 7, 1983
;    96          0096  1 !                 Reduce signalling while in EXEC mode.
;    97          0097  1 !
;    98          0098  1 !--
;    99          0099  1
;   100          0100  1 !
;   101          0101  1 ! Include files
;   102          0102  1 !
;   103          0103  1
;   104          0104  1 LIBRARY 'SYS$LIBRARY:LIB.L32';          ! VAX/VMS system definitions
;   105          0105  1
;   106          0106  1 REQUIRE 'SRC$:INSPREFIX.REQ';
;   107          0248  1 REQUIRE 'LIB$:INSDEF.R32';
```

```
;  109          0307  1 %SBTTL  'Declarations';
;  110          0308  1 !
;  111          0309  1 ! Table of contents
;  112          0310  1 !
;  113          0311  1
;  114          0312  1 FORWARD ROUTINE
;  115          0313  1     INS_LIST,
;  116          0314  1     LIST_KFE_ENTRIES,                       ! Traverse structure to list all KFEs
;  117          0315  1     LIST_KFE_ENTRY,                         ! List one KFE
;  118          0316  1     FORMAT_KFD,                             ! Format and print KFD block
;  119          0317  1     FORMAT_KFE,                             ! Format and print KFE entry
;  120          0318  1     PRINT_PRIVS,                            ! Print the ASCII keywords for the bits set in a quadword pr
;  121          0319  1     FORMAT_LINE,                            ! Format ASCII output into line buffer.
;  122          0320  1     TERMINATE_LINE : NOVALUE,               ! Copy line buffer to temporary buffer
;  123          0321  1     FORMAT_TERMINATE_LINE : NOVALUE,        ! Format then terminate line
;  124          0322  1     PRINTOUT;                               ! Print the contents of the temporary buffer
;  125          0323  1
;  126          0324  1 !
;  127          0325  1 ! External routines
;  128          0326  1 !
;  129          0327  1
;  130          0328  1 EXTERNAL ROUTINE
;  131          0329  1     LIB$GET_VM,                             ! get virtual memory
;  132          0330  1     LIB$FREE_VM,                            ! return virtual memory
;  133          0331  1     LIB$PUT_OUTPUT,
;  134          0332  1     SYS$GETDVIW : ADDRESSING_MODE (GENERAL),    ! Get Device Information
;  135          0333  1     SYS$FAOL : ADDRESSING_MODE (GENERAL);       ! Format ASCII output
;  136          0334  1
;  137          0335  1 EXTERNAL ROUTINE
;  138          0336  1     INS$EXECUTE_IN_EXEC_WITH_R_LOCK;
;  139          0337  1
;  140          0338  1 EXTERNAL
;  141          0339  1     CTL$GL_KNOWNFIL,                        ! Process pointer to the Known file list pointer block
;  142          0340  1     EXE$GL_KNOWN_FILES,                     ! Exec pointer to the Known file list pointer block
;  143          0341  1     INS$GL_CTLMSK : BLOCK [1],              ! INSTALL control flags
;  144          0342  1     INS$G_OUTRAB : BBLOCK,                  ! Record output block for output buffer
;  145          0343  1     PRV$AB_NAMES;                           ! ASCII list of privileges
;  146          0344  1
;  147          0345  1 EXTERNAL LITERAL
;  148          0346  1     INS$_EMPTYLST,                          ! The Known File List is empty
;  149          0347  1     INS$_FAILGETVM,                         ! Failed to get virtual memory
;  150          0348  1     INS$_NOLIST,                            ! There is no Known File List
;  151          0349  1     INS$_NOVER;                             ! Error obtaining file version
;  152          0350  1
;  153          0351  1 GLOBAL
;  154          0352  1     INS$FAOOUTBUF,                          ! Output buffer
;  155          0353  1     INS$FAOBUFDESC : BBLOCK [DSC$C_S_BLN];  ! Descriptor of output buffer
;  156          0354  1
;  157          0355  1 GLOBAL LITERAL
;  158          0356  1     INS$C_FAOBUFLEN = 255;                  ! size of output buffer
;  159          0357  1
;  160          0358  1 !
;  161          0359  1 !       Set up user buffer for copying lists to while in kernel mode
;  162          0360  1 !
;  163          0361  1 OWN
;  164          0362  1     TMPBUF_LEN,                             ! Size of allocated buffer
;  165          0363  1     TMPBUF;                                 ! Address of allocated buffer
```

```
;  166     0364  1     TMPBUF_PTR : REF $BBLOCK;                    ! Point to free buffer space
;  167     0365  1
;  168     0366  1 BIND
;  169     0367  1     !
;  170     0368  1     !   Control strings for FAO
;  171     0369  1     !
;  172     0370  1     FAOCTL_DDT          = $DESCRIPTOR ('!AS!AS'),
;  173     0371  1     FAOCTL_VERSION      = $DESCRIPTOR (';!UW'),
;  174     0372  1     FAOCTL_KFDADR       = $DESCRIPTOR (' List head adr/siz/ref = !XL/!UW/!UW'),
;  175     0373  1     FAOCTL_FILNAM       = $DESCRIPTOR ('   !AC'),
;  176     0374  1     FAOCTL_FLAGS        = $DESCRIPTOR ('!AC'),
;  177     0375  1     FAOCTL_KFEADR       = $DESCRIPTOR ('        Entry address/size/index   = !XL/!UW/!XB'),
;  178     0376  1     FAOCTL_WINDOW       = $DESCRIPTOR ('        Window address/size        = !XL/!UW'),
;  179     0377  1     FAOCTL_HEADER       = $DESCRIPTOR ('        Header address/size        = !XL/!UW'),
;  180     0378  1     FAOCTL_USECNT       = $DESCRIPTOR ('        Entry access count         = !UL'),
;  181     0379  1     FAOCTL_SHRUSECNT    = $DESCRIPTOR ('        Current / Maximum shared   = !UW / !UW'),
;  182     0380  1     FAOCTL_CMODCURR     = $DESCRIPTOR ('        Current shared count       = !UW'),
;  183     0381  1     FAOCTL_GBLCNT       = $DESCRIPTOR ('        Global section count       = !UW'),
;  184     0382  1     FAOCTL_COMPAT_TYP   = $DESCRIPTOR ('        Compatability type         = !XW'),
;  185     0383  1     FAOCTL_PRIVHD       = $DESCRIPTOR ('        Privileges = '),
;  186     0384  1     FAOCTL_PRIVHD2      = $DESCRIPTOR ('                     '),
;  187     0385  1     FAOCTL_PRIV         = $DESCRIPTOR ('!AC ');
;  188     0386  1
```

```
;  190          0387  1 %SBTTL  'GET_NUMENTRIES';
;  191          0388  1 ROUTINE GET_NUMENTRIES (RETCOUNT) =
;  192          0389  2 BEGIN
;  193          0390  2 !+++
;  194          0391  2 ! FUNCTIONAL DESCRIPTION:
;  195          0392  2 !
;  196          0393  2 !       Return the number of entries to allocate for the listing.
;  197          0394  2 !
;  198          0395  2 !--
;  199          0396  2 MAP
;  200          0397  2     RETCOUNT : REF VECTOR[,LONG];
;  201          0398  2
;  202          0399  2 BIND
;  203          0400  2     KFPB = EXE$GL_KNOWN_FILES : REF $BBLOCK;
;  204          0401  2
;  205          0402  2 IF .KFPB EQL 0
;  206          0403  2     THEN RETURN INS$_NOLIST;
;  207          0404  2
;  208          0405  2 IF .KFPB[KFPB$L_KFDLST] EQL 0
;  209          0406  2     THEN RETURN INS$_EMPTYLST;
;  210          0407  2
;  211          0408  2 RETCOUNT[0] = .KFPB[KFPB$W_KFDLSTCNT];
;  212          0409  2 RETURN TRUE
;  213          0410  1 END;


                                                                .TITLE  INSLIST
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                53  41  21  53  41  21  00000 P.AAB:  .ASCII  \!AS!AS\                                    ;
                                                          00006         .BLKB   2                                           ;
                                                    00000006  00008 P.AAA:  .LONG   6                                       ;
                                                    00000000' 0000C         .ADDRESS P.AAB                                  ;
                                        57  55  21  3B  00010 P.AAD:  .ASCII  \;!UW\                                      ;
                                                    00000004  00014 P.AAC:  .LONG   4                                       ;
                                                    00000000' 00018         .ADDRESS P.AAD                                  ;
2F  72  64  61  20  64  61  65  68  20  74  73  69  4C  20  0001C P.AAF:  .ASCII  \ List head adr/siz/ref = !XL/!UW/!UW\     ;
21  2F  4C  58  21  20  3D  20  66  65  72  2F  7A  69  73  0002B                                                           ;
                                    57  55  21  2F  57  55  0003A                                                           ;
                                                    00000024  00040 P.AAE:  .LONG   36                                      ;
                                                    00000000' 00044         .ADDRESS P.AAF                                  ;
                                43  41  21  20  20  20  00048 P.AAH:  .ASCII  \   !AC\                                    ;
                                                          0004E         .BLKB   2                                           ;
                                                    00000006  00050 P.AAG:  .LONG   6                                       ;
                                                    00000000' 00054         .ADDRESS P.AAH                                  ;
                                            43  41  21  00058 P.AAJ:  .ASCII  \!AC\                                       ;
                                                          0005B         .BLKB   1                                           ;
                                                    00000003  0005C P.AAI:  .LONG   3                                       ;
                                                    00000000' 00060         .ADDRESS P.AAJ                                  ;
61  20  79  72  74  6E  45  20  20  20  20  20  20  20  20  00064 P.AAL:  .ASCII  \        Entry address/size/index    = !XL\ ;
64  6E  69  2F  65  7A  69  73  2F  73  73  65  72  64  64  00073                                                           ;
                        4C  58  21  20  3D  20  20  20  78  65  00082                                                       ;
                            42  58  21  2F  57  55  21  2F  0008C         .ASCII  \/!UW/!XB\                              ;
                                                    00000030  00094 P.AAK:  .LONG   48                                      ;
                                                    00000000' 00098         .ADDRESS P.AAL                                  ;
```

```
20 77 6F 64 6E 69 57 20 20 20 20 20 20 20 20 0009C P.AAN:  .ASCII  \        Window address/size        = !XL\
20 20 20 65 7A 69 73 2F 73 73 65 72 64 64 61 000AB
                  4C 58 21 20 3D 20 20 20 20 000BA
                                 57 55 21 2F 000C4         .ASCII  \/!UW\
                                    0000002C 000C8 P.AAM:  .LONG   44
                                    00000000' 000CC        .ADDRESS P.AAN
20 72 65 64 61 65 48 20 20 20 20 20 20 20 20 000D0 P.AAP:  .ASCII  \        Header address/size        = !XL\
20 20 20 65 7A 69 73 2F 73 73 65 72 64 64 61 000DF
                  4C 58 21 20 3D 20 20 20 20 000EE
                                 57 55 21 2F 000F8         .ASCII  \/!UW\
                                    0000002C 000FC P.AAO:  .LONG   44
                                    00000000' 00100        .ADDRESS P.AAP
61 20 79 72 74 6E 45 20 20 20 20 20 20 20 20 00104 P.AAR:  .ASCII  \        Entry access count         = !UL\
20 20 20 20 74 6E 75 6F 63 20 73 73 65 63 63 00113
                  4C 55 21 20 3D 20 20 20 20 00122
                                    00000028 0012C P.AAQ:  .LONG   40
                                    00000000' 00130        .ADDRESS P.AAR
74 6E 65 72 72 75 43 20 20 20 20 20 20 20 20 00134 P.AAT:  .ASCII  \        Current / Maximum shared   = !UW\
72 61 68 73 20 6D 75 6D 69 78 61 4D 20 2F 20 00143
                  57 55 21 20 3D 20 20 20 64 65 00152
                              57 55 21 20 2F 20 0015C         .ASCII  \ / !UW\
                                                00162         .BLKB   2
                                    0000002E 00164 P.AAS:  .LONG   46
                                    00000000' 00168        .ADDRESS P.AAT
74 6E 65 72 72 75 43 20 20 20 20 20 20 20 20 0016C P.AAV:  .ASCII  \        Current shared count\<9>\   = \
20 09 74 6E 75 6F 63 20 64 65 72 61 68 73 20 0017B
                                 20 3D 20 20 0018A
                                    57 55 21 0018E         .ASCII  \!UW\
                                                00191         .BLKB   3
                                    00000025 00194 P.AAU:  .LONG   37
                                    00000000' 00198        .ADDRESS P.AAV
20 6C 61 62 6F 6C 47 20 20 20 20 20 20 20 20 0019C P.AAX:  .ASCII  \        Global section count       = !UW\
20 20 74 6E 75 6F 63 20 6E 6F 69 74 63 65 73 001AB
                  57 55 21 20 3D 20 20 20 20 001BA
                                    00000028 001C4 P.AAW:  .LONG   40
                                    00000000' 001C8        .ADDRESS P.AAX
61 74 61 70 6D 6F 43 20 20 20 20 20 20 20 20 001CC P.AAZ:  .ASCII  \        Compatability type         = !XW\
20 20 20 20 65 70 79 74 20 79 74 69 6C 69 62 001DB
                  57 58 21 20 3D 20 20 20 20 001EA
                                    00000028 001F4 P.AAY:  .LONG   40
                                    00000000' 001F8        .ADDRESS P.AAZ
65 6C 69 76 69 72 50 20 20 20 20 20 20 20 20 001FC P.ABB:  .ASCII  \        Privileges = \
                              20 3D 20 73 65 67 0020B
                                                00211         .BLKB   3
                                    00000015 00214 P.ABA:  .LONG   21
                                    00000000' 00218        .ADDRESS P.ABB
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 0021C P.ABD:  .ASCII  \                    \
                              20 20 20 20 20 20 0022B
                                                00231         .BLKB   3
                                    00000015 00234 P.ABC.  .LONG   21
                                    00000000' 00238        .ADDRESS P.ABD
                                 20 43 41 21 0023C P.ABF:  .ASCII  \!AC \
                                    00000004 00240 P.ABE:  .LONG   4
                                    00000000' 00244        .ADDRESS P.ABF

                                                           .PSECT  $OWN$,NOEXE,2
```

```
00000 TMPBUF_LEN:
                .BLKB   4
00004 TMPBUF:   .BLKB   4
00008 TMPBUF_PTR:
                .BLKB   4

                .PSECT  $GLOBAL$,NOEXE,2

00000 INS$FAOOUTBUF::
                .BLKB   4
00004 INS$FAOBUFDESC::
                .BLKB   8

      INS$C_FAOBUFLEN==   255
      FAOCTL_DDT=         P.AAA
      FAOCTL_VERSION=     P.AAC
      FAOCTL_KFDADR=      P.AAE
      FAOCTL_FILNAM=      P.AAG
      FAOCTL_FLAGS=       P.AAI
      FAOCTL_KFEADR=      P.AAK
      FAOCTL_WINDOW=      P.AAM
      FAOCTL_HEADER=      P.AAO
      FAOCTL_USECNT=      P.AAQ
      FAOCTL_SHRUSECNT=   P.AAS
      FAOCTL_CMODCURR=    P.AAU
      FAOCTL_GBLCNT=      P.AAW
      FAOCTL_COMPAT_TYP=  P.AAY
      FAOCTL_PRIVHD=      P.ABA
      FAOCTL_PRIVHD2=     P.ABC
      FAOCTL_PRIV=        P.ABE
                .EXTRN  LIB$GET_VM, LIB$FREE_VM
                .EXTRN  LIB$PUT_OUTPUT, SYS$GETDVIW
                .EXTRN  SYS$FAOL, INS$EXECUTE_IN_EXEC_WITH_R_LOCK
                .EXTRN  CTL$GL_KNOWNFIL
                .EXTRN  EXE$GL_KNOWN_FILES
                .EXTRN  INS$GL_CTLMSR, INS$G_OUTRAB
                .EXTRN  PRV$AB_NAMES, INS$_EMPTYLST
                .EXTRN  INS$_FAILGETVM, INS$_NOLIST
                .EXTRN  INS$_NOVER

                .PSECT  $CODE$,NOWRT,2

                    0000 00000 GET_NUMENTRIES:
                               .WORD   Save nothing                          ; 0388
      50 00000000G   00  D0 00002      MOVL    KFPB, R0                      ; 0402
                     08  12 00009      BNEQ    1$                            ;
      50 00000000G   8F  D0 0000B      MOVL    #INS$_NOLIST, R0              ; 0403
                     04  00012        RET                                   ;
                     60  D5 00013 1$: TSTL    (R0)                          ; 0405
                     08  12 00015      BNEQ    2$                            ;
      50 00000000G   8F  D0 00017      MOVL    #INS$_EMPTYLST, R0            ; 0406
                     04  0001E        RET                                   ;
04    BC       0C    A0  3C 0001F 2$: MOVZWL  12(R0), @RETCOUNT             ; 0408
      50             01  D0 00024      MOVL    #1, R0                        ; 0409
                         04 00027      RET                                   ; 0410

; Routine Size:  40 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  215     0411  1 %SBTTL  'INS$LIST';
;  216     0412  1
;  217     0413  1 GLOBAL ROUTINE INS$LIST ( KFE ) =
;  218     0414  2 BEGIN
;  219     0415  2 !+++
;  220     0416  2 !
;  221     0417  2 !   FUNCTIONAL DESCRIPTION:
;  222     0418  2 !
;  223     0419  2 !       Print the contents of either a specific KFE or all the KFE's.
;  224     0420  2 !
;  225     0421  2 !   INPUT:
;  226     0422  2 !
;  227     0423  2 !       kfe     = 0 : list all the KFE entries in all the lists.
;  228     0424  2 !               = n : List the KFE entry at address "n".
;  229     0425  2 !
;  230     0426  2 !   IMPLICIT OUTPUT:
;  231     0427  2 !
;  232     0428  2 !       none
;  233     0429  2 !
;  234     0430  2 !   ROUTINE VALUE:
;  235     0431  2 !
;  236     0432  2 !---
;  237     0433  2 LITERAL
;  238     0434  2     MAXLINLEN = 80,
;  239     0435  2     NUM_FULL_LINES = 3,
;  240     0436  2     NUM_STRUC_LINES = 3;
;  241     0437  2
;  242     0438  2 LOCAL
;  243     0439  2     NUM_ENTRIES,
;  244     0440  2     NUM_LINES,
;  245     0441  2     CME_ARGLST : VECTOR[2,LONG],
;  246     0442  2     STATUS;
;  247     0443  2
;  248     0444  2 !
;  249     0445  2 ! Initialize output buffer and descriptor
;  250     0446  2 !
;  251     0447  2 CH$FILL (%C' ',INS$C_FAOBUFLEN, .INS$FAOOUTBUF);
;  252     0448  2 INS$FAOBUFDESC [DSC$W_LENGTH] = INS$C_FAOBUFLEN;
;  253     0449  2 INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOOUTBUF;
;  254     0450  2
;  255     0451  2 NUM_ENTRIES = 0;
;  256     0452  2 CME_ARGLST[0] = 1;
;  257     0453  2 CME_ARGLST[1] = NUM_ENTRIES;
;  258     0454  2 STATUS = $CMEXEC(ROUTIN=GET_NUMENTRIES,ARGLST=CME_ARGLST);
;  259     0455  2 IF .STATUS NEQ TRUE
;  260     0456  3 THEN BEGIN
;  261     0457  3     SIGNAL(.STATUS);
;  262     0458  3     RETURN TRUE
;  263     0459  2     END;
;  264     0460  2
;  265     0461  2 IF .KFE NEQ 0
;  266     0462  2 THEN
;  267     0463  2     NUM_ENTRIES = 2;                            ! KFE and KFD
;  268     0464  2
;  269     0465  2 NUM_LINES = 2;
;  270     0466  2 IF .INS$GL_CTLMSK [INS$V_FULL] THEN NUM_LINES = .NUM_LINES + NUM_FULL_LINES;
;  271     0467  2 IF .INS$GL_CTLMSK [INS$V_STRUCTURE] THEN NUM_LINES = .NUM_LINES + NUM_STRUC_LINES;
```

```
;  272          0468  2 TMPBUF_LEN = MAXLINLEN * .NUM_LINES * .NUM_ENTRIES;
;  273          0469  2 STATUS = LIB$GET_VM (TMPBUF_LEN, TMPBUF);
;  274          0470  2 IF NOT .STATUS
;  275          0471  2 THEN
;  276          0472  3     BEGIN
;  277          0473  3     SIGNAL (INS$_FAILGETVM, 1, .TMPBUF_LEN, .STATUS);
;  278          0474  3     RETURN TRUE;
;  279          0475  2     END;
;  280          0476  2
;  281          0477  2 CH$FILL (%C' ', .TMPBUF_LEN, .TMPBUF);
;  282          0478  2 TMPBUF_PTR = .TMPBUF;
;  283          0479  2
;  284          0480  2
;  285          0481  2 STATUS = INS$EXECUTE_IN_EXEC_WITH_R_LOCK (INS_LIST, .KFE);
;  286          0482  2 PRINTOUT ();                                    ! Print the contents of TMPBUF
;  287          0483  2
;  288          0484  2 EXECUTE ( LIB$FREE_VM (TMPBUF_LEN, TMPBUF) );          ! Return the buffer
;  289          0485  2
;  290          0486  2 RETURN .STATUS;
;  291          0487  1 END;                                    ! routine  INS$LIST


                                                        .EXTRN  SYS$CMEXEC

                                                01FC 00000      .ENTRY  INS$LIST, Save R2,R3,R4,R5,R6,R7,R8          ; 0413
                                58 00000000G    00   9E 00002  MOVAB   LIB$SIGNAL, R8                             ;
                                57      0000'   CF   9E 00009  MOVAB   TMPBUF_LEN, R7                           ;
                                5E              0C   C2 0000E  SUBL2   #12, SP                                  ;
   00FF    8F           20      6E              00   2C 00011  MOVC5   #0, (SP), #32, #255, @INS$FAOOUTBUF      ; 0447
                                        0000'   DF      00018                                                   ;
                      0000'     CF        FF    8F   9B 0001B  MOVZBW  #255, INS$FAOBUFDESC                     ; 0448
                      0000'     CF      0000'   CF   D0 00021  MOVL    INS$FAOOUTBUF, INS$FAOBUFDESC+4          ; 0449
                                        6E      D4   00028      CLRL    NUM_ENTRIES                             ; 0451
                        04      AE              01   D0 0002A  MOVL    #1, CME_ARGLST                          ; 0452
                        08      AE              6E   9E 0002E  MOVAB   NUM_ENTRIES, CME_ARGLST+4               ; 0453
                                        04      AE   9F 00032  PUSHAB  CME_ARGLST                              ; 0454
                                        A0      AF   9F 00035  PUSHAB  GET_NUMENTRIES                          ;
                00000000G       00              02   FB 00038  CALLS   #2, SYS$CMEXEC                         ;
                                56              50   D0 0003F  MOVL    R0, STATUS                            ;
                                01              56   D1 00042  CMPL    STATUS, #1                            ; 0455
                                                07   13 00045  BEQL    1$                                    ;
                                                56   DD 00047  PUSHL   STATUS                                ; 0457
                                68              01   FB 00049  CALLS   #1, LIB$SIGNAL                        ;
                                                4D   11 0004C  BRB     5$                                    ; 0458
                                        04      AC   D5 0004E  1$:     TSTL    KFE                           ; 0461
                                                03   13 00051  BEQL    2$                                    ;
                                6E              02   D0 00053  MOVL    #2, NUM_ENTRIES                       ; 0463
                                50              02   D0 00056  2$:     MOVL    #2, NUM_LINES                 ; 0465
                03 00000000G    00              02   E1 00059  BBC     #2, INS$GL_CTLMSK+1, 3$               ; 0466
                                50              03   C0 00061  ADDL2   #3, NUM_LINES                        ;
                03 00000000G    00              03   E1 00064  3$:     BBC     #3, INS$GL_CTLMSK+1, 4$       ; 0467
                                50              03   C0 0006C  ADDL2   #3, NUM_LINES                        ;
                                50              6E   C4 0006F  4$:     MULL2   NUM_ENTRIES, R0               ; 0468
   67                   50 00000050    8F   C5 00072  MULL3   #80, R0, TMPBUF_LEN                           ;
                                   04   A7   9F 0007A  PUSHAB  TMPBUF                                         ; 0469
                                        57   DD 0007D  PUSHL   R7                                           ;
```

```
                    00000000G  00          02  FB 0007F          CALLS   #2, LIB$GET_VM                                    ;
                               56          50  D0 00086          MOVL    R0, STATUS                                      ;
                               13          56  E8 00089          BLBS    STATUS, 6$                                      ; 0470
                                           56  DD 0008C          PUSHL   STATUS                                          ; 0473
                                           67  DD 0008E          PUSHL   TMPBUF_LEN                                      ;
                                           01  DD 00090          PUSHL   #1                                              ;
                                00000000G  8F  DD 00092          PUSHL   #INS$_FAILGETVM                                 ;
                               68          04  FB 00098          CALLS   #4, LIB$SIGNAL                                  ;
                               50          01  D0 0009B 5$:      MOVL    #1, R0                                          ; 0474
                                           04  0009E             RET                                                     ;
          67          20                   6E  00  2C 0009F 6$:  MOVC5   #0, (SP), #32, TMPBUF_LEN, @TMPBUF              ; 0477
                                        04     B7     000A4                                                              ;
                                  08    A7  04  A7  D0 000A6      MOVL    TMPBUF, TMPBUF_PTR                              ; 0478
                                        04     AC  DD 000AB      PUSHL   KFE                                             ; 0481
                                     0000V     CF  9F 000AE      PUSHAB  INS_LIST                                        ;
                    00000000G  00              02  FB 000B2      CALLS   #2, INS$EXECUTE_IN_EXEC_WITH_R_LOCK             ;
                               56              50  D0 000B9      MOVL    R0, STATUS                                      ;
                         0000V CF              00  FB 000BC      CALLS   #0, PRINTOUT                                    ; 0482
                                        04     A7  9F 000C1      PUSHAB  TMPBUF                                          ; 0484
                                               57  DD 000C4      PUSHL   R7                                              ;
                    00000000G  00              02  FB 000C6      CALLS   #2, LIB$FREE_VM                                 ;
                               03              50  E9 000CD      BLBC    STATUS, 7$                                      ;
                               50              56  D0 000D0      MOVL    STATUS, R0                                      ; 0486
                                               04  000D3 7$:     RET                                                     ; 0487

; Routine Size:  212 bytes,    Routine Base:  $CODE$ + 0028

;  292          0488  1
```

```
;  294     0489  1 %SBTTL  'INS_LIST';
;  295     0490  1
;  296     0491  1 ROUTINE INS_LIST ( KFE ) =
;  297     0492  2 BEGIN
;  298     0493  2 !+++
;  299     0494  2 !
;  300     0495  2 !   FUNCTIONAL DESCRIPTION:
;  301     0496  2 !
;  302     0497  2 !       Print the contents of either a specific KFE or all the KFE's.
;  303     0498  2 !
;  304     0499  2 !   INPUT:
;  305     0500  2 !
;  306     0501  2 !       kfe       = 0 : list all the KFE entries in all the lists.
;  307     0502  2 !                 = n : List the KFE entry at address "n".
;  308     0503  2 !
;  309     0504  2 !   IMPLICIT INPUT:
;  310     0505  2 !
;  311     0506  2 !       ins$gl_ctlmsk : INSTALL control flags determine whether to give an
;  312     0507  2 !                       abreviated or FULL listing.
;  313     0508  2 !       ins$g_outrab :  Record access block for output stream.
;  314     0509  2 !
;  315     0510  2 !   OUTPUT:
;  316     0511  2 !
;  317     0512  2 !       List the known file image list for a single entry
;  318     0513  2 !       or for every entry in all the lists.
;  319     0514  2 !
;  320     0515  2 !   IMPLICIT OUTPUT:
;  321     0516  2 !
;  322     0517  2 !       none
;  323     0518  2 !
;  324     0519  2 !   ROUTINE VALUE:
;  325     0520  2 !
;  326     0521  2 !---
;  327     0522  2 LOCAL
;  328     0523  2     STATUS;
;  329     0524  2
;  330     0525  2 !+++
;  331     0526  2 !
;  332     0527  2 !       Format and print the contents of the buffer
;  333     0528  2 !
;  334     0529  2 !---
;  335     0530  2
;  336     0531  2 IF .KFE EQL 0
;  337     0532  2 THEN
;  338     0533  2     STATUS = LIST_KFE_ENTRIES ()
;  339     0534  2 ELSE
;  340     0535  2     STATUS = LIST_KFE_ENTRY (.KFE);
;  341     0536  2
;  342     0537  2
;  343     0538  2 RETURN .STATUS;
;  344     0539  1 END;                                    ! routine  INS_LIST



                                        0000 00000 INS_LIST:
```

```
                                                 .WORD   Save nothing                    ; 0491
                                04   AC  D5 00002       TSTL    KFE                     ; 0531
                                     06  12 00005       BNEQ    1$
                          0000V  CF      00  FB 00007   CALLS   #0, LIST_KFE_ENTRIES    ; 0533
                                         04 0000C       RET
                                04   AC  DD 0000D 1$:   PUSHL   KFE                     ; 0535
                          0000V  CF      01  FB 00010   CALLS   #1, LIST_KFE_ENTRY
                                         04 00015       RET                             ; 0539

; Routine Size:  22 bytes,    Routine Base:  $CODE$ + 00FC

;  345          0540  1
```

```
;  347     0541  1 ROUTINE LIST_KFE_ENTRIES =
;  348     0542  1 !+++
;  349     0543  1 !
;  350     0544  1 !
;  351     0545  1 !---
;  352     0546  2 BEGIN
;  353     0547  2 LOCAL
;  354     0548  2     KFD : REF BBLOCK,
;  355     0549  2     KFE : REF BBLOCK;
;  356     0550  2
;  357     0551  2 BIND
;  358     0552  2     KFPB = EXE$GL_KNOWN_FILES : REF BBLOCK;
;  359     0553  2
;  360     0554  2 IF .KFPB EQL 0
;  361     0555  2 THEN
;  362     0556  3     BEGIN
;  363     0557  3     RETURN INS$_NOLIST;
;  364     0558  2     END;
;  365     0559  2
;  366     0560  2 IF .KFPB [KFPB$L_KFDLST] EQL 0
;  367     0561  2 THEN
;  368     0562  3     BEGIN
;  369     0563  3     RETURN INS$_EMPTYLST;
;  370     0564  2     END;
;  371     0565  2
;  372     0566  2 KFD = .KFPB [KFPB$L_KFDLST];
;  373     0567  2
;  374     0568  2 !
;  375     0569  2 !   Traverse the list of KFDs and format each KFD and all its KFEs.
;  376     0570  2 !   The KFD is the header block which contains the Device, directory and
;  377     0571  2 !   file type which several Known File Entries (KFE) share in common.
;  378     0572  2 !
;  379     0573  2 WHILE .KFD NEQ 0 DO
;  380     0574  3     BEGIN
;  381     0575  3     FORMAT_KFD (.KFD);
;  382     0576  3     KFE = .KFD [KFD$L_KFELIST];
;  383     0577  3
;  384     0578  3     !
;  385     0579  3     !   Format each KFE in the KFD's ordered list of KFEs
;  386     0580  3     !
;  387     0581  3     WHILE .KFE NEQ 0 DO
;  388     0582  4         BEGIN
;  389     0583  4         FORMAT_KFE (.KFE);
;  390     0584  4         KFE = .KFE [KFE$L_KFELINK];
;  391     0585  3         END;                        ! WHILE traversing KFD's ordered KFE list
;  392     0586  3
;  393     0587  3     KFD = .KFD [KFD$L_LINK];        ! Next KFD
;  394     0588  2     END;                            ! WHILE traversing KFD list
;  395     0589  2
;  396     0590  2 RETURN TRUE;
;  397     0591  1 END;


                                    000C 00000 LIST_KFE_ENTRIES:
```

```
                                                        .WORD   Save R2,R3                      ; 0541
                        50 00000000G    00   D0 00002           MOVL    KFPB, R0                ; 0554
                                        08   12 00009           BNEQ    1$                      ;
                        50 00000000G    8F   D0 0000B           MOVL    #INS$_NOLIST, R0        ; 0557
                                             04 00012           RET                             ;
                                        60   D5 00013 1$:       TSTL    (R0)                    ; 0560
                                        08   12 00015           BNEQ    2$                      ;
                        50 00000000G    8F   D0 00017           MOVL    #INS$_EMPTYLST, R0      ; 0563
                                             04 0001E           RET                             ;
                        52              60   D0 0001F 2$:       MOVL    (R0), KFD               ; 0566
                                        1F   13 00022 3$:       BEQL    6$                      ; 0573
                                        52   DD 00024           PUSHL   KFD                     ; 0575
                  0000V CF              01   FB 00026           CALLS   #1, FORMAT_KFD          ;
                        53        04    A2   D0 0002B           MOVL    4(KFD), KFE             ; 0576
                                        0D   13 0002F 4$:       BEQL    5$                      ; 0581
                                        53   DD 00031           PUSHL   KFE                     ; 0583
                  0000V CF              01   FB 00033           CALLS   #1, FORMAT_KFE          ;
                        53        04    A3   D0 00038           MOVL    4(KFE), KFE             ; 0584
                                        F1   11 0003C           BRB     4$                      ; 0581
                        52              62   D0 0003E 5$:       MOVL    (KFD), KFD              ; 0587
                                        DF   11 00041           BRB     3$                      ; 0573
                        50              01   D0 00043 6$:       MOVL    #1, R0                  ; 0590
                                             04 00046           RET                             ; 0591

; Routine Size:  71 bytes,    Routine Base:  $CODE$ + 0112

;  398          0592  1
```

```
;  400          0593  1 ROUTINE LIST_KFE_ENTRY (KFE) =
;  401          0594  1 !+++
;  402          0595  1 !
;  403          0596  1 !
;  404          0597  1 !---
;  405          0598  2 BEGIN
;  406          0599  2 MAP
;  407          0600  2     KFE : REF BBLOCK;
;  408          0601  2
;  409          0602  2 FORMAT_KFD (.KFE [KFE$L_KFD]);
;  410          0603  2
;  411          0604  2 FORMAT_KFE (.KFE);
;  412          0605  2
;  413          0606  2 RETURN TRUE;
;  414          0607  1 END;


                                        0004 00000 LIST_KFE_ENTRY:
                                                           .WORD   Save R2                                 ; 0593
                              52    04    AC  D0 00002           MOVL    KFE, R2                         ; 0602
                                    0C    A2  DD 00006           PUSHL   12(R2)                          ;
                    0000V     CF          01  FB 00009           CALLS   #1, FORMAT_KFD                  ;
                              52          52  DD 0000E           PUSHL   R2                              ; 0604
                    0000V     CF          01  FB 00010           CALLS   #1, FORMAT_KFE                  ;
                              50          01  D0 00015           MOVL    #1, R0                          ; 0606
                                              04 00018           RET                                     ; 0607

; Routine Size:  25 bytes,    Routine Base:  $CODE$ + 0159

;  415          0608  1
```

```
;  417     0609  1 ROUTINE FORMAT_KFD (KFD) =
;  418     0610  1 !+++
;  419     0611  1 !
;  420     0612  1 !
;  421     0613  1 !---
;  422     0614  2 BEGIN
;  423     0615  2 LITERAL
;  424     0616  2     INS_C_KFDPADLEN = 40;
;  425     0617  2
;  426     0618  2 LOCAL
;  427     0619  2     DEVNAM : BBLOCK [65],
;  428     0620  2     NEW_DEV : BBLOCK [65],
;  429     0621  2     DEVNAM_DSC : $BBLOCK [DSC$C_S_BLN],
;  430     0622  2     DDT_DSC : $BBLOCK [DSC$C_S_BLN],
;  431     0623  2     ITMLST : VECTOR [4,LONG]
;  432     0624  2         PRESET ( [0] = DVI$_LOGVOLNAM ^ 16 + 64,
;  433     0625  2                  [3] = 0),
;  434     0626  2     PAD;
;  435     0627  2
;  436     0628  2 MAP
;  437     0629  2     KFD : REF BBLOCK;
;  438     0630  2
;  439     0631  2 TERMINATE_LINE ();                              ! Blank line
;  440     0632  2
;  441     0633  2 IF .INS$GL_CTLMSK [INS$V_STRUCTURE]
;  442     0634  2 THEN
;  443     0635  2     !
;  444     0636  2     ! For a /STRUCTURE listing, simply display the device name in its
;  445     0637  2     ! raw form as it was stored in the KFD.
;  446     0638  2     !
;  447     0639  3     BEGIN
;  448     0640  3     DEVNAM_DSC [DSC$W_LENGTH] = .KFD [KFD$B_DEVLEN];
;  449     0641  3     DEVNAM_DSC [DSC$A_POINTER] = KFD [KFD$T_DDTSTR];
;  450     0642  3     END
;  451     0643  2 ELSE                                            ! not /STRUCTURE listing
;  452     0644  2     !
;  453     0645  2     ! Build a device name by extracting it from the DDTSTR field of the KFD
;  454     0646  2     ! and prefixing it with an underscore. The underscore tells $GETDVI that
;  455     0647  2     ! this is a device name and not to bother trying to translate the string.
;  456     0648  2     !
;  457     0649  3     BEGIN
;  458     0650  3     DEVNAM_DSC[DSC$A_POINTER] = DEVNAM;          ! Load addr of devnam string
;  459     0651  3     CH$WCHAR(%C' ',.DEVNAM_DSC[DSC$A_POINTER]); ! Device, not a logical name
;  460     0652  3     CH$MOVE(.KFD[KFD$B_DEVLEN],KFD[KFD$T_DDTSTR],.DEVNAM_DSC[DSC$A_POINTER]+1);
;  461     0653  3     DEVNAM_DSC[DSC$W_LENGTH] = .KFD[KFD$B_DEVLEN]+1; ! Calculate devnam string length
;  462     0654  3
;  463     0655  3     !
;  464     0656  3     ! Call GETDVI to convert the device name into the volume's logical name
;  465     0657  3     ! string. This achieves a less confusing/intimidating device name display
;  466     0658  3     ! for the users.
;  467     0659  3     !
;  468     0660  3     ITMLST [1] = NEW_DEV;                        ! Load output buffer address
;  469     0661  3     ITMLST [2] = DEVNAM_DSC[DSC$W_LENGTH];       ! Shove length into descriptor
;  470     0662  3     SYS$GETDVIW (0,0,DEVNAM_DSC,ITMLST,0,0,0,0);! Convert device name format
;  471     0663  3
;  472     0664  3     !
;  473     0665  3     ! Build the output descriptor for formatting below. If there was a
```

```
;  474          0666  3     ! volume logical name defined, then go ahead and use it. If not, then
;  475          0667  3     ! we must use the original device name.
;  476          0668  3     !
;  477          0669  3     IF .DEVNAM_DSC[DSC$W_LENGTH] EQL 0
;  478          0670  3     THEN
;  479          0671  4         BEGIN
;  480          0672  4         DEVNAM_DSC [DSC$W_LENGTH] = .KFD [KFD$B_DEVLEN];
;  481          0673  4         DEVNAM_DSC [DSC$A_POINTER] = KFD [KFD$T_DDTSTR];
;  482          0674  4         END
;  483          0675  3     ELSE
;  484          0676  4         BEGIN
;  485          0677  4         CH$WCHAR(':',NEW_DEV+.DEVNAM_DSC[DSC$W_LENGTH]);    ! Add and count colon
;  486          0678  4         DEVNAM_DSC[DSC$W_LENGTH] = .DEVNAM_DSC[DSC$W_LENGTH] + 1 ;
;  487          0679  4         DEVNAM_DSC[DSC$A_POINTER] = NEW_DEV ;               ! Finish descriptor
;  488          0680  3         END;
;  489          0681  2     END;
;  490          0682  2 !
;  491          0683  2 ! Now, format the output line.
;  492          0684  2 !
;  493          0685  2 DDT_DSC[DSC$W_LENGTH] = .KFD[KFD$B_DDTSTRLEN] - .KFD[KFD$B_DEVLEN];
;  494          0686  2 DDT_DSC[DSC$A_POINTER] = KFD[KFD$T_DDTSTR] + .KFD[KFD$B_DEVLEN];
;  495          0687  2 FORMAT_LINE (FAOCTL_DDT, DEVNAM_DSC, DDT_DSC);  ! Format the KFD output
;  496          0688  2
;  497          0689  2 IF .INS$GL_CTLMSK [INS$V_STRUCTURE]
;  498          0690  2 THEN
;  499          0691  3     BEGIN
;  500          0692  3     !
;  501          0693  3     !   Pad the buffer out to INS_C_KFDPADLEN characters
;  502          0694  3     !
;  503          0695  3     PAD = INS_C_KFDPADLEN - (INS$C_FAOBUFLEN - .INS$FAOBUFDESC [DSC$W_LENGTH]);
;  504          0696  3     IF .PAD LEQ 0
;  505          0697  3     THEN
;  506          0698  4         BEGIN
;  507          0699  4         TERMINATE_LINE ();                                 ! Print DDT string on first line
;  508          0700  4         PAD = INS_C_KFDPADLEN;
;  509          0701  3         END;
;  510          0702  3
;  511          0703  3     INS$FAOBUFDESC [DSC$W_LENGTH] = .INS$FAOBUFDESC [DSC$W_LENGTH] - .PAD; !length is size left in buffer
;  512          0704  3     INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOBUFDESC [DSC$A_POINTER] + .PAD;
;  513          0705  3
;  514          0706  3     FORMAT_TERMINATE_LINE (FAOCTL_KFDADR, .KFD,
;  515          0707  3         .KFD [KFD$W_SIZE], .KFD [KFD$W_REFCNT]);           ! Print KFD info
;  516          0708  2     END;
;  517          0709  2
;  518          0710  2 TERMINATE_LINE ();                                         ! Blank line if /STRUCTURE, else prints DDT string
;  519          0711  2
;  520          0712  2 RETURN TRUE;
;  521          0713  1 END;


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                002C0040  00248 P.ABG:  .LONG   2883648                                                 ;
                                     00# 0024C          .BYTE   0[8]                                                    ;
                                00000000  00254          .LONG   0                                                       ;
```

```
                                                        .PSECT  $CODE$,NOWRT,2

                                                01FC 00000 FORMAT_KFD:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8               ; 0609
                                58      0000V   CF  9E  00002           MOVAB   TERMINATE_LINE, R8
                                5E      FF58    CE  9E  00007           MOVAB   -168(SP), SP
                            6E      0000'       CF      10  28  0000C   MOVC3   #16, P.ABG, ITMLST        ; 0625
                                    68          00      FB  00012       CALLS   #0, TERMINATE_LINE      ; 0631
                                    56      04  AC      D0  00015       MOVL    KFD, R6                 ; 0640
                            06 00000000G    00          03  E1  00019   BBC     #3, INS$GL_CTLMSK+1, 1$ ; 0633
                                    57      0E  A6      9A  00021       MOVZBL  14(R6), R7              ; 0640
                                            3F          11  00025       BRB     2$
                                1C  AE      64  AE      9E  00027 1$:   MOVAB   DEVNAM, DEVNAM_DSC+4    ; 0650
                                    50      1C  AE      D0  0002C       MOVL    DEVNAM_DSC+4, R0        ; 0651
                                    60      5F  8F      90  00030       MOVB    #95, (R0)
                                    57      0E  A6      9A  00034       MOVZBL  14(R6), R7              ; 0652
                        01  A0          11  A6          57  28  00038   MOVC3   R7, 17(R6), 1(R0)
                        18  AE              57          01  A1  0003E   ADDW3   #1, R7, DEVNAM_DSC      ; 0653
                                04  AE          20  AE  9E  00043       MOVAB   NEW_DEV, ITMLST+4       ; 0660
                                08  AE          18  AE  9E  00048       MOVAB   DEVNAM_DSC, ITMLST+8    ; 0661
                                                    7E  7C  0004D       CLRQ    -(SP)                   ; 0662
                                                    7E  7C  0004F       CLRQ    -(SP)
                                                10  AE  9F  00051       PUSHAB  ITMLST
                                                2C  AE  9F  00054       PUSHAB  DEVNAM_DSC
                                                    7E  7C  00057       CLRQ    -(SP)
                                00000000G       00      08  FB  00059   CALLS   #8, SYS$GETDVIW
                                    50          18  AE  3C  00060       MOVZWL  DEVNAM_DSC, R0          ; 0669
                                                    0B  12  00064       BNEQ    3$
                                18  AE              57  B0  00066 2$:   MOVW    R7, DEVNAM_DSC          ; 0672
                                1C  AE          11  A6  9E  0006A       MOVAB   17(R6), DEVNAM_DSC+4    ; 0673
                                                    0D  11  0006F       BRB     4$                      ; 0669
                                20 AE40                 3A  90  00071 3$:   MOVB    #58, NEW_DEV[R0]        ; 0677
                                            18  AE      B6  00076       INCW    DEVNAM_DSC              ; 0678
                                1C  AE          20  AE  9E  00079       MOVAB   NEW_DEV, DEVNAM_DSC+4   ; 0679
                                    50          10  A6  9A  0007E 4$:   MOVZBL  16(R6), R0              ; 0685
                        10  AE                  57  A3  00082           SUBW3   R7, R0, DDT_DSC
                                14  AE          11 A746 9E  00087       MOVAB   17(R7)[R6], DDT_DSC+4   ; 0686
                                            10  AE      9F  0008D       PUSHAB  DDT_DSC                 ; 0687
                                            1C  AE      9F  00090       PUSHAB  DEVNAM_DSC
                                        0000'   CF      9F  00093       PUSHAB  FAOCTL_DDT
                                0000V   CF              03  FB  00097   CALLS   #3, FORMAT_LINE
                            2F 00000000G    00          03  E1  0009C   BBC     #3, INS$GL_CTLMSK+1, 6$ ; 0689
                                    52      0000'   CF  3C  000A4       MOVZWL  INS$FAOBUFDESC, PAD     ; 0695
                                    52      FF29    C2  9E  000A9       MOVAB   -215(R2), PAD
                                                        06  14  000AE   BGTR    5$                      ; 0696
                                    68                  00  FB  000B0   CALLS   #0, TERMINATE_LINE      ; 0699
                                    52                  28  D0  000B3   MOVL    #40, PAD                ; 0700
                                0000'   CF              52  A2  000B6 5$:   SUBW2   PAD, INS$FAOBUFDESC    ; 0703
                                0000'   CF              52  C0  000BB       ADDL2   PAD, INS$FAOBUFDESC+4  ; 0704
                                        7E      0C  A6  3C  000C0       MOVZWL  12(R6), -(SP)           ; 0707
                                        7E      08  A6  3C  000C4       MOVZWL  8(R6), -(SP)
                                                    56  DD  000C8       PUSHL   R6                      ; 0706
                                        0000'   CF  9F  000CA           PUSHAB  FAOCTL_KFDADR
                                0000V   CF              04  FB  000CE   CALLS   #4, FORMAT_TERMINATE_LINE
                                    68                  00  FB  000D3 6$:   CALLS   #0, TERMINATE_LINE      ; 0710
```

```
                                 50          01  D0 000D6          MOVL    #1, R0                          ; 0712
                                                 04 000D9          RET                                     ; 0713

; Routine Size:  218 bytes,    Routine Base:  $CODE$ + 0172

;  522          0714  1
```

```
;  524          0715  1 ROUTINE FORMAT_KFE (KFE) =
;  525          0716  1 !+++
;  526          0717  1 !
;  527          0718  1 !
;  528          0719  1 !---
;  529          0720  2 BEGIN
;  530          0721  2 MAP
;  531          0722  2     KFE : REF BBLOCK;
;  532          0723  2
;  533          0724  2 !
;  534          0725  2 !   Constants for setting file information block to get the file version
;  535          0726  2 !   number returned via a call to QIO.
;  536          0727  2 !
;  537          0728  2 LITERAL
;  538          0729  2     FIB_C_FID = 10,
;  539          0730  2     FIB_C_DID = 12,
;  540          0731  2     FIB_S_FID = 8,
;  541          0732  2     INS_C_CTLFLGSTR = 12,
;  542          0733  2     INS_C_KFEPADLEN = 20;
;  543          0734  2
;  544          0735  2 OWN
;  545          0736  2     FILVER : LONG,                                      ! Address to return file version
;  546          0737  2     ATRCTLBLK : BBLOCK [12]                             ! Attribute control block to get version number from
;  547          0738  2                 PRESET  ([ATR$W_SIZE] = ATR$S_FILVER,
;  548          0739  2                          [ATR$W_TYPE] = ATR$C_FILVER,   ! request file version
;  549          0740  2                          [ATR$L_ADDR] = FILVER
;  550          0741  2                         ),
;  551          0742  2
;  552          0743  2     FIB : BBLOCK [FIB_C_DID],
;  553          0744  2     FIB_DESC : BBLOCK [DSC$C_S_BLN]
;  554          0745  2                 PRESET ([DSC$W_LENGTH] = FIB_C_FID,
;  555          0746  2                         [DSC$A_POINTER] = FIB ),
;  556          0747  2
;  557          0748  2     !
;  558          0749  2     !   Control flag array to translate KFE flags to the ASCII
;  559          0750  2     !   to be formatted for output.
;  560          0751  2     !
;  561          0752  2     CTLFLG_ARRAY : VECTOR [2*INS_C_CTLFLGSTR] INITIAL (
;  562          0753  2                 KFE$M_OPEN,     CSTRING ('Open '),
;  563          0754  2                 KFE$M_HDRRES,   CSTRING ('Hdr '),
;  564          0755  2                 KFE$M_SHARED,   CSTRING ('Shar '),
;  565          0756  2                 KFE$M_PROCPRIV, CSTRING ('Prv '),
;  566          0757  2                 KFE$M_PROTECT,  CSTRING ('Prot '),
;  567          0758  2                 KFE$M_LIM,      CSTRING ('Lnkbl '),
;  568          0759  2                 KFE$M_COMPATMOD,CSTRING ('Cmode '),
;  569          0760  2                 KFE$M_SHMIDENT, CSTRING ('Shm '),
;  570          0761  2                 KFE$M_ACCOUNT,  CSTRING ('Acnt '),
;  571          0762  2                 KFE$M_NOPURGE,  CSTRING ('Nopurg '),
;  572          0763  2                 KFE$M_WRITEABLE,CSTRING ('Wrt '),
;  573          0764  2                 KFE$M_EXEONLY,  CSTRING ('Xonly')
;  574          0765  2                 );
;  575          0766  2
;  576          0767  2
;  577          0768  2 LOCAL
;  578          0769  2     FID : BBLOCK [FIB_S_FID],
;  579          0770  2     FLAGS,
;  580          0771  2     KFD : REF BBLOCK,
```

```
;  581     0772  2     PAD,                    ! Number of blanks to pad after filename
;  582     0773  2     QIO_STATUS,
;  583     0774  2     STATUS,
;  584     0775  2     WCB_SHRCNT,
;  585     0776  2     WCB_SIZ,
;  586     0777  2     WCB : REF BBLOCK;
;  587     0778  2
;  588     0779  2 KFD = .KFE [KFE$L_KFD];
;  589     0780  2
;  590     0781  2 !
;  591     0782  2 !   Print the File name
;  592     0783  2 !
;  593     0784  2 FORMAT_LINE (FAOCTL_FILNAM, KFE [KFE$B_FILNAMLEN]);
;  594     0785  2
;  595     0786  2 CH$FILL (0, FIB_S_FID, FID);    ! zero it out
;  596     0787  2 WCB = 0;
;  597     0788  2
;  598     0789  2 IF .KFE [KFE$V_OPEN]    ! If installed /OPEN, get info from window control block
;  599     0790  2 THEN
;  600     0791  3     BEGIN
;  601     0792  3     LOCAL
;  602     0793  3         FCB : REF BBLOCK;
;  603     0794  3
;  604     0795  3     WCB = .KFE [KFE$L_WCB];
;  605     0796  3     IF .WCB NEQ 0
;  606     0797  3     THEN
;  607     0798  4         BEGIN
;  608     0799  4         WCB_SIZ = .WCB [WCB$W_SIZE];
;  609     0800  4         WCB_SHRCNT = .WCB [WCB$W_REFCNT] - .KFE [KFE$W_GBLSECCNT] - 1;  ! Amount of file sharing
;  610     0801  4         FCB = .WCB [WCB$L_FCB];
;  611     0802  4         IF .FCB LSS 0
;  612     0803  4         THEN
;  613     0804  5             BEGIN
;  614     0805  5             CH$MOVE (FIB_S_FID, FCB [FCB$W_FID], FID)
;  615     0806  4             END;
;  616     0807  3         END;
;  617     0808  3     END
;  618     0809  3
;  619     0810  2 ELSE
;  620     0811  2     CH$MOVE (FIB_S_FID, KFE [KFE$W_FID], FID);
;  621     0812  2
;  622     0813  2
;  623     0814  2 !
;  624     0815  2 !   If we obtained the file id field then the file version can be obtained via
;  625     0816  2 !   a call to QIO.
;  626     0817  2 !
;  627     0818  2 IF NOT CH$FAIL (CH$FIND_NOT_CH (FIB_S_FID, FID, 0))     ! See if it is all zeros
;  628     0819  2 THEN
;  629     0820  3     BEGIN
;  630     0821  3     LOCAL
;  631     0822  3         CHANNEL : WORD,
;  632     0823  3         DEVNAM_DESC : BBLOCK [DSC$C_S_BLN],
;  633     0824  3         IOSB : BBLOCK [8];
;  634     0825  3
;  635     0826  3     CH$MOVE (FIB_S_FID, FID, FIB [FIB$W_FID] );
;  636     0827  3     !
;  637     0828  3     !   make descriptor of device name string
```

```
;  638          0829  3     !
;  639          0830  3     DEVNAM_DESC = .KFD [KFD$B_DEVLEN];
;  640          0831  3     DEVNAM_DESC [DSC$A_POINTER] = KFD [KFD$T_DDTSTR];
;  641          0832  3
;  642          0833  3     !
;  643          0834  3     !   Assign a channel so QIO can be called to get file version
;  644          0835  3     !
;  645          0836  3     STATUS = $ASSIGN ( DEVNAM = DEVNAM_DESC, CHAN = CHANNEL);
;  646          0837  3     IF NOT .STATUS THEN RETURN .STATUS;
;  647          0838  3     FILVER = 0;
;  648        P 0839  3     QIO_STATUS = $QIOW (FUNC = IO$_ACCESS, CHAN = .CHANNEL,          ! Get the file version
;  649          0840  3                         IOSB = IOSB, P1 = FIB_DESC, P5 = ATRCTLBLK);
;  650          0841  3
;  651          0842  3     EXECUTE ($DASSGN (CHAN = .CHANNEL) );       ! Deassign the channel
;  652          0843  3
;  653          0844  3     IF NOT .IOSB
;  654          0845  3     THEN
;  655          0846  4         BEGIN
;  656          0847  4         !
;  657          0848  4         !   Build a descriptor of the file name which is now in
;  658          0849  4         !   the output buffer and indicate that the file was not found
;  659          0850  4         !
;  660          0851  4         LITERAL
;  661          0852  4             DECODED_MSGBUF_LEN = 256,
;  662          0853  4             ERRFILNAM_BUFLEN = 31;
;  663          0854  4         LOCAL
;  664          0855  4             DECODED_MSGDSC : BBLOCK [DSC$C_S_BLN],
;  665          0856  4             DECODED_MSGBUF : BBLOCK [DECODED_MSGBUF_LEN],
;  666          0857  4             ERRFILNAM_DSC : BBLOCK [DSC$C_S_BLN],
;  667          0858  4             ERRFILNAM_BUF : BBLOCK [ERRFILNAM_BUFLEN];
;  668          0859  4
;  669          0860  4         ERRFILNAM_DSC [DSC$A_POINTER] = ERRFILNAM_BUF;
;  670          0861  4         ERRFILNAM_DSC = INS$C_FAOBUFLEN - .INS$FAOBUFDESC [DSC$W_LENGTH];
;  671          0862  4         IF .ERRFILNAM_DSC [DSC$W_LENGTH] GTR ERRFILNAM_BUFLEN
;  672          0863  4         THEN ERRFILNAM_DSC [DSC$W_LENGTH] = ERRFILNAM_BUFLEN;
;  673          0864  4
;  674          0865  4         CH$MOVE (.ERRFILNAM_DSC [DSC$W_LENGTH], .INS$FAOOUTBUF,
;  675          0866  4                 ERRFILNAM_BUF);
;  676          0867  4
;  677          0868  4         DECODED_MSGDSC = DECODED_MSGBUF_LEN;
;  678          0869  4         DECODED_MSGDSC [DSC$A_POINTER] = DECODED_MSGBUF;
;  679          0870  4         CH$FILL (0, DECODED_MSGBUF_LEN, DECODED_MSGBUF);
;  680          0871  4
;  681        P 0872  4         $GETMSG ( MSGID = INS$_NOVER,
;  682        P 0873  4                 MSGLEN = DECODED_MSGDSC,
;  683          0874  4                 BUFADR = DECODED_MSGDSC);
;  684          0875  4
;  685          0876  4         TERMINATE_LINE ();
;  686          0877  4         FORMAT_TERMINATE_LINE ( DECODED_MSGDSC, ERRFILNAM_DSC);
;  687          0878  4
;  688          0879  4         DECODED_MSGDSC = DECODED_MSGBUF_LEN;
;  689          0880  4         DECODED_MSGDSC [DSC$A_POINTER] = DECODED_MSGBUF;
;  690          0881  4         CH$FILL (0, DECODED_MSGBUF_LEN, DECODED_MSGBUF);
;  691          0882  4
;  692        P 0883  4         $GETMSG ( MSGID = .IOSB,
;  693        P 0884  4                 MSGLEN = DECODED_MSGDSC,
;  694          0885  4                 BUFADR = DECODED_MSGDSC);
```

```
;  695          0886  4
;  696          0887  4          FORMAT_LINE ( DECODED_MSGDSC);
;  697          0888  4          END
;  698          0889  4
;  699          0890  3      ELSE
;  700          0891  4          BEGIN
;  701          0892  4          FORMAT_LINE (FAOCTL_VERSION, .FILVER ); ! Format the version into output buffer
;  702          0893  3          END;
;  703          0894  2      END;
;  704          0895  2
;  705          0896  2 !
;  706          0897  2 !    Pad the buffer out to INS_C_KFEPADLEN characters
;  707          0898  2 !
;  708          0899  2 PAD = INS_C_KFEPADLEN - (INS$C_FAOBUFLEN - .INS$FAOBUFDESC [DSC$W_LENGTH]);
;  709          0900  2 IF .PAD LEQ 0
;  710          0901  2 THEN
;  711          0902  3      BEGIN
;  712          0903  3      TERMINATE_LINE ();
;  713          0904  3      PAD = INS_C_KFEPADLEN;
;  714          0905  2      END;
;  715          0906  2
;  716          0907  2 INS$FAOBUFDESC [DSC$W_LENGTH] = .INS$FAOBUFDESC [DSC$W_LENGTH] - .PAD; !length is size left in buffer
;  717          0908  2 INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOBUFDESC [DSC$A_POINTER] + .PAD;
;  718          0909  2
;  719          0910  2 !
;  720          0911  2 !    Decode KFE flags
;  721          0912  2 !
;  722          0913  3 BEGIN
;  723          0914  3      LOCAL
;  724          0915  3          BUFLEN,
;  725          0916  3          BUFPTR;
;  726          0917  3
;  727          0918  3      BUFLEN = .INS$FAOBUFDESC [DSC$W_LENGTH];
;  728          0919  3      BUFPTR = .INS$FAOBUFDESC [DSC$A_POINTER];
;  729          0920  3
;  730          0921  3      FLAGS = .KFE [KFE$W_FLAGS];
;  731          0922  3
;  732          0923  3      !
;  733          0924  3      !    Search the table, if the mask is set in the composite control
;  734          0925  3      !    flags  longword, then call FAOL with the corresponding descriptor
;  735          0926  3      !
;  736          0927  3      INCR I FROM 0 TO (2 * INS_C_CTLFLGSTR  -1) BY 2 DO
;  737          0928  4          BEGIN
;  738          0929  4          BIND
;  739          0930  4              MASK = CTLFLG_ARRAY [.I],
;  740          0931  4              CSTRNG = CTLFLG_ARRAY [.I] + 4,
;  741          0932  4              PADLEN = .CSTRNG : BYTE;
;  742          0933  4
;  743          0934  4          IF (.MASK AND .FLAGS) NEQ 0
;  744          0935  4          THEN
;  745          0936  5              BEGIN
;  746          0937  5              FORMAT_LINE (FAOCTL_FLAGS, .CSTRNG);
;  747          0938  5              BUFLEN = .INS$FAOBUFDESC [DSC$W_LENGTH];
;  748          0939  5              BUFPTR = .INS$FAOBUFDESC [DSC$A_POINTER];
;  749          0940  5              END
;  750          0941  4          ELSE
;  751          0942  5              BEGIN
```

```
;  752     0943  5              INS$FAOBUFDESC [DSC$W_LENGTH] = .INS$FAOBUFDESC [DSC$W_LENGTH] - .PADLEN;
;  753     0944  5              INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOBUFDESC [DSC$A_POINTER] + .PADLEN;
;  754     0945  4              END;
;  755     0946  3          END;
;  756     0947  3
;  757     0948  3      INS$FAOBUFDESC [DSC$W_LENGTH] = .BUFLEN;
;  758     0949  3      INS$FAOBUFDESC [DSC$A_POINTER] = .BUFPTR;
;  759     0950  2      END;
;  760     0951  2
;  761     0952  2  !
;  762     0953  2  !   Print extra info for a /FULL or /STRUCTURE listing
;  763     0954  2  !
;  764     0955  2  IF .INS$GL_CTLMSK [INS$V_FULL]
;  765     0956  2  THEN
;  766     0957  3      BEGIN
;  767     0958  3      TERMINATE_LINE ();                  ! Print file name or decoded flags
;  768     0959  3
;  769     0960  3      IF .INS$GL_CTLMSK [INS$V_STRUCTURE]
;  770     0961  3      THEN
;  771     0962  3          FORMAT_TERMINATE_LINE (FAOCTL_KFEADR, .KFE,
;  772     0963  3                                .KFE [KFE$W_SIZE], .KFE [KFE$B_HSHIDX]);
;  773     0964  3
;  774     0965  3      IF .KFE [KFE$V_COMPATMOD]   ! Mark as compatibility mode image
;  775     0966  3      THEN
;  776     0967  3          FORMAT_TERMINATE_LINE (FAOCTL_COMPAT_TYP, .KFE [KFE$W_AMECOD])
;  777     0968  3      ELSE
;  778     0969  3          FORMAT_TERMINATE_LINE (FAOCTL_USECNT, .KFE [KFE$L_USECNT]);
;  779     0970  3
;  780     0971  3      IF .KFE [KFE$V_OPEN]        ! If /OPEN
;  781     0972  3      THEN
;  782     0973  3          IF .KFE [KFE$V_COMPATMOD]
;  783     0974  3          THEN
;  784     0975  3              FORMAT_TERMINATE_LINE (FAOCTL_CMODCURR, .WCB_SHRCNT)
;  785     0976  3          ELSE
;  786     0977  3              FORMAT_TERMINATE_LINE (FAOCTL_SHRUSECNT,
;  787     0978  3                                     .WCB_SHRCNT, .KFE [KFE$W_SHRCNT] - 1);
;  788     0979  3
;  789     0980  3      IF .KFE [KFE$V_SHARED]      ! If /SHARED
;  790     0981  3      THEN
;  791     0982  3          FORMAT_TERMINATE_LINE (FAOCTL_GBLCNT, .KFE [KFE$W_GBLSECCNT]);
;  792     0983  3
;  793     0984  4      IF (.INS$GL_CTLMSK [INS$V_STRUCTURE] AND .WCB NEQ 0)        ! If installed /OPEN, print info on window
;  794     0985  3      THEN
;  795     0986  3          FORMAT_TERMINATE_LINE (FAOCTL_WINDOW, .WCB, .WCB_SIZ);
;  796     0987  3
;  797     0988  4      IF (.INS$GL_CTLMSK [INS$V_STRUCTURE] AND .KFE [KFE$V_HDRRES])       ! If header resident
;  798     0989  3      THEN
;  799     0990  4          BEGIN
;  800     0991  4          BIND
;  801     0992  4              KFRH = .KFE [KFE$L_IMGHDR] - KFRH$C_LENGTH : BBLOCK;
;  802     0993  4
;  803     0994  4          FORMAT_TERMINATE_LINE (FAOCTL_HEADER,
;  804     0995  4                                 .KFE [KFE$L_IMGHDR], .KFRH [KFE$W_SIZE]);
;  805     0996  3          END;
;  806     0997  3
;  807     0998  3      IF .KFE [KFE$V_PROCPRIV]
;  808     0999  3      THEN
```

```
;  809          1000  3          PRINT_PRIVS (KFE [KFE$Q_PROCPRIV]);
;  810          1001  3
;  811          1002  2      END;                ! Full listing
;  812          1003  2
;  813          1004  2 TERMINATE_LINE ();              ! If /FULL prints blank line, else prints file name
;  814          1005  2
;  815          1006  2 RETURN TRUE;
;  816          1007  1 END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                00000005 00258 P.ABH:   .BYTE   5
                                            20  6E  65  70  4F 00259     .ASCII  \Open \
                                                00000004 0025E P.ABI:   .BYTE   4
                                                20  72  64  48 0025F     .ASCII  \Hdr \
                                                00000005 00263 P.ABJ:   .BYTE   5
                                            20  72  61  68  53 00264     .ASCII  \Shar \
                                                00000004 00269 P.ABK:   .BYTE   4
                                                20  76  72  50 0026A     .ASCII  \Prv \
                                                00000005 0026E P.ABL:   .BYTE   5
                                            20  74  6F  72  50 0026F     .ASCII  \Prot \
                                                00000006 00274 P.ABM:   .BYTE   6
                                        20  6C  62  6B  6E  4C 00275     .ASCII  \Lnkbl \
                                                00000006 0027B P.ABN:   .BYTE   6
                                        20  65  64  6F  6D  43 0027C     .ASCII  \Cmode \
                                                00000004 00282 P.ABO:   .BYTE   4
                                                20  6D  68  53 00283     .ASCII  \Shm \
                                                00000005 00287 P.ABP:   .BYTE   5
                                                20  74  6E  63  41 00288     .ASCII  \Acnt \
                                                00000007 0028D P.ABQ:   .BYTE   7
                                    20  67  72  75  70  6F  4E 0028E     .ASCII  \Nopurg \
                                                00000004 00295 P.ABR:   .BYTE   4
                                                20  74  72  57 00296     .ASCII  \Wrt \
                                                00000005 0029A P.ABS:   .BYTE   5
                                            79  6C  6E  6F  58 0029B     .ASCII  \Xonly\

                                                        .PSECT  $OWN$,NOEXE,2

                                                0000C FILVER:   .BLKB   4
                                        0007    0002  00010 ATRCTLBLK:
                                                        .WORD   2, 7
                                          00000000' 00014           .ADDRESS FILVER
                                                00018             .BLKB   4
                                                0001C FIB:        .BLKB   12
                                            000A  00028 FIB_DESC:
                                                        .WORD   10
                                              00#   0002A         .BYTE   0[2]
                                          00000000' 0002C         .ADDRESS FIB
                                          00000008  00030 CTLFLG_ARRAY:
                                                        .LONG   8
                                          00000000' 00034         .ADDRESS P.ABH
                                          00000010  00038         .LONG   16
                                          00000000' 0003C         .ADDRESS P.ABI
                                          00000020  00040         .LONG   32
                                          00000000' 00044         .ADDRESS P.ABJ
```

```
                00000004  00048          .LONG   4                                       ;
                00000000' 0004C          .ADDRESS P.ABK                                  ;
                00000001  00050          .LONG   1                                       ;
                00000000' 00054          .ADDRESS P.ABL                                  ;
                00000002  00058          .LONG   2                                       ;
                00000000' 0005C          .ADDRESS P.ABM                                  ;
                00000080  00060          .LONG   128                                     ;
                00000000' 00064          .ADDRESS P.ABN                                  ;
                00000040  00068          .LONG   64                                      ;
                00000000' 0006C          .ADDRESS P.ABO                                  ;
                00000200  00070          .LONG   512                                     ;
                00000000' 00074          .ADDRESS P.ABP                                  ;
                00000100  00078          .LONG   256                                     ;
                00000000' 0007C          .ADDRESS P.ABQ                                  ;
                00000400  00080          .LONG   1024                                    ;
                00000000' 00084          .ADDRESS P.ABR                                  ;
                00000800  00088          .LONG   2048                                    ;
                00000000' 0008C          .ADDRESS P.ABS                                  ;

                                         .EXTRN  SYS$ASSIGN, SYS$QIOW
                                         .EXTRN  SYS$DASSGN, SYS$GETMSG

                                         .PSECT  $CODE$,NOWRT,2

                          0FFC 00000 FORMAT_KFE:
                                         .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 0715
                5E      FEB4   CE  9E 00002      MOVAB   -332(SP), SP                    ;
                58        04   AC  D0 00007      MOVL    KFF, R8                         ; 0779
                59        0C   A8  D0 0000B      MOVL    12(R8), KFD                     ;
                          36   A8  9F 0000F      PUSHAB  54(R8)                          ; 0784
                        0000'  CF  9F 00012      PUSHAB  FAOCTL_FILNAM                   ;
                      0000V    CF     02  FB 00016      CALLS   #2, FORMAT_LINE         ;
    08            00           6E     00  2C 0001B      MOVC5   #0, (SP), #0, #8, FID   ; 0786
                                      F8  AD   00020                                   ;
                                          56  D4 00022  CLRL    WCB                     ; 0787
                                    5A        10  A8 9E 00024  MOVAB   16(R8), R10      ; 0789
                      25                    6A     03  E1 00028  BBC     #3, (R10), 1$   ;
                                          56  18  A8 D0 0002C  MOVL    24(R8), WCB     ; 0795
                                          25  13  00030  BEQL    2$                      ; 0796
                                    5B        08  A6 3C 00032  MOVZWL  8(WCB), WCB_SIZ  ; 0799
                                    57        0E  A6 3C 00036  MOVZWL  14(WCB), R7      ; 0800
                                    50        12  A8 3C 0003A  MOVZWL  18(R8), R0      ;
                                    57            50  C2 0003E  SUBL2   R0, R7         ;
                                                  57  D7 00041  DECL    WCB_SHRCNT      ;
                                    50        18  A6 D0 00043  MOVL    24(WCB), FCB    ; 0801
                                                  0E  18 00047  BGEQ    2$              ; 0802
                F8      AD      24    A0              08  28 00049  MOVC3   #8, 36(FCB), FID ; 0805
                                                      06  11 0004F  BRB     2$          ; 0789
                F8      AD      18    A8              08  28 00051  1$:  MOVC3   #8, 24(R8), FID ; 0811
                F8      AD            08              00  3B 00057  2$:  SKPC    #0, #8, FID ; 0818
                                                      02  12 0005C  BNEQ    3$          ;
                                                      51  D4 0005E  CLRL    R1          ;
                                                      51  D5 00060  3$:  TSTL    R1      ;
                                                      03  12 00062  BNEQ    4$          ;
                                                  00FD    31 00064  BRW     10$         ;
              0000'     CF      F8    AD              08  28 00067  4$:  MOVC3   #8, FID, FIB+4 ; 0826
                                F0    AD        0E    A9 9A 0006E      MOVZBL  14(KFD), DEVNAM_DESC ; 0830
```

```
                        F4   AD      11   A9  9E 00073          MOVAB   17(R9), DEVNAM_DESC+4                    ; 0831
                                         7E  7C 00078          CLRQ    -(SP)                                     ; 0836
                                     08  AE  9F 0007A          PUSHAB  CHANNEL                                   ;
                                     F0  AD  9F 0007D          PUSHAB  DEVNAM_DESC                               ;
                 00000000G           00      FB 00080          CALLS   #4, SYS$ASSIGN                            ;
                                     30      E9 00087          BLBC    STATUS, 5$                                ; 0837
                               0000'     CF  D4 0008A          CLRL    FILVER                                    ; 0838
                                         7E  D4 0008E          CLRL    -(SP)                                     ; 0840
                               0000'     CF  9F 00090          PUSHAB  ATRCTLBLK                                 ;
                                         7E  7C 00094          CLRQ    -(SP)                                     ;
                                         7E  D4 00096          CLRL    -(SP)                                     ;
                               0000'     CF  9F 00098          PUSHAB  FIB_DESC                                  ;
                                         7E  7C 0009C          CLRQ    -(SP)                                     ;
                                     E8  AD  9F 0009E          PUSHAB  IOSB                                      ;
                                         32  DD 000A1          PUSHL   #50                                       ;
                                     7E  28  AE  3C 000A3          MOVZWL  CHANNEL, -(SP)                        ;
                                         7E  D4 000A7          CLRL    -(SP)                                     ;
                 00000000G           00      FB 000A9          CALLS   #12, SYS$QIOW                             ;
                                     7E      6E  3C 000B0          MOVZWL  CHANNEL, -(SP)                        ; 0842
                 00000000G           00      01  FB 000B3          CALLS   #1, SYS$DASSGN                        ;
                                     01      50  E8 000BA 5$:       BLBS    STATUS, 6$                            ;
                                             04  000BD           RET                                           ;
                                     03      F8  AD  E9 000BE 6$:       BLBC    IOSB, 7$                      ; 0844
                                           0092      31 000C2           BRW     9$                            ;
                                  28   AE      04  AE  9E 000C5 7$:       MOVAB   ERRFILNAM_BUF, ERRFILNAM_DSC+4; 0860
                                  24   AE      0000'   CF  3C 000CA          MOVZWL  INS$FAOBUFDESC, ERRFILNAM_DSC ; 0861
                            24    AE 000000FF   8F      24  AE  C3 000D0          SUBL3   ERRFILNAM_DSC, #255, ERRFILNAM_DSC
                                     1F      24  AE  B1 000DA          CMPW    ERRFILNAM_DSC, #31                ; 0862
                                             04  1B 000DE          BLEQU   8$                                    ;
                                  24   AE              1F  B0 000E0          MOVW    #31, ERRFILNAM_DSC            ; 0863
                            04    AE      0000'   DF      24  AE  28 000E4 8$:       MOVC3   ERRFILNAM_DSC, @INS$FAOOUTBUF, - ; 0865
                                                                                      ERRFILNAM_BUF               ;
                                  E0   AD      0100    8F  3C 000EC          MOVZWL  #256, DECODED_MSGDSC          ; 0868
                                  E4   AD        2C    AE  9E 000F2          MOVAB   DECODED_MSGBUF, DECODED_MSGDSC+4 ; 0869
                 0100 8F                  00              6E  00  2C 000F7          MOVC5   #0, (SP), #0, #256, DECODED_MSGBUF ; 0870
                                                        2C  AE      000FE
                                                     7E             0F  7D 00100          MOVQ    #15, -(SP)    ; 0874
                                                        E0  AD  9F 00103          PUSHAB  DECODED_MSGDSC        ;
                                                        E0  AD  9F 00106          PUSHAB  DECODED_MSGDSC        ;
                                                 00000000G      8F  DD 00109          PUSHL   #INS$_NOVER           ;
                                           00000000G            00  FB 0010F          CALLS   #5, SYS$GETMSG        ;
                                             0000V              CF  00  FB 00116          CALLS   #0, TERMINATE_LINE ; 0876
                                                        24  AE  9F 0011B          PUSHAB  ERRFILNAM_DSC         ; 0877
                                                        E0  AD  9F 0011E          PUSHAB  DECODED_MSGDSC        ;
                                             0000V      CF      02  FB 00121          CALLS   #2, FORMAT_TERMINATE_LINE
                                               E0   AD      0100    8F  3C 00126          MOVZWL  #256, DECODED_MSGDSC ; 0879
                                               E4   AD        2C    AE  9E 0012C          MOVAB   DECODED_MSGBUF, DECODED_MSGDSC+4 ; 0880
                 0100 8F                  00              6E  00  2C 00131          MOVC5   #0, (SP), #0, #256, DECODED_MSGBUF ; 0881
                                                        2C  AE      00138
                                                     7E             0F  7D 0013A          MOVQ    #15, -(SP)    ; 0885
                                                        E0  AD  9F 0013D          PUSHAB  DECODED_MSGDSC        ;
                                                        E0  AD  9F 00140          PUSHAB  DECODED_MSGDSC        ;
                                                        E8  AD  DD 00143          PUSHL   IOSB                  ;
                                           00000000G            00  FB 00146          CALLS   #5, SYS$GETMSG        ;
                                                        E0  AD  9F 0014D          PUSHAB  DECODED_MSGDSC        ; 0887
                                             0000V      CF      01  FB 00150          CALLS   #1, FORMAT_LINE       ;
                                                             0D  11 00155          BRB     10$                   ; 0844
```

```
                                0000'  CF   DD 00157 9$:     PUSHL   FILVER                                  ; 0892
                                0000'  CF   9F 0015B         PUSHAB  FAOCTL_VERSION                          ;
                     0000V      CF          02 FB 0015F      CALLS   #2, FORMAT_LINE                         ;
                                52     0000'CF   3C 00164 10$:  MOVZWL  INS$FAOBUFDESC, PAD                   ; 0899
                                52     FF15  C2  9E 00169       MOVAB   -235(R2), PAD                         ;
                                           08  14 0016E         BGTR    11$                                   ; 0900
                     0000V      CF          00 FB 00170      CALLS   #0, TERMINATE_LINE                      ; 0903
                                52          14 D0 00175      MOVL    #20, PAD                              ; 0904
                     0000'      CF          52 A2 00178 11$: SUBW2   PAD, INS$FAOBUFDESC                   ; 0907
                     0000'      CF          52 C0 0017D      ADDL2   PAD, INS$FAOBUFDESC+4                 ; 0908
                                55     0000'CF   3C 00182       MOVZWL  INS$FAOBUFDESC, BUFLEN                ; 0918
                                54     0000'CF   D0 00187       MOVL    INS$FAOBUFDESC+4, BUFPTR              ; 0919
                                59          6A 3C 0018C      MOVZWL  (R10), FLAGS                          ; 0921
                                52          D4 0018F         CLRL    I                                     ; 0934
                                53     0000'CF42 D0 00191 12$:  MOVL    CTLFLG_ARRAY+4[I], R3                 ; 0932
                                59     0000'CF42 D3 00197       BITL    CTLFLG_ARRAY[I], FLAGS                ; 0934
                                           1A  13 0019D         BEQL    13$                                   ;
                                0000'CF42  DD 0019F         PUSHL   CTLFLG_ARRAY+4[I]                       ; 0937
                                0000'  CF   9F 001A4         PUSHAB  FAOCTL_FLAGS                            ;
                     0000V      CF          02 FB 001A8      CALLS   #2, FORMAT_LINE                         ;
                                55     0000'  CF   3C 001AD     MOVZWL  INS$FAOBUFDESC, BUFLEN                ; 0938
                                54     0000'  CF   D0 001B2     MOVL    INS$FAOBUFDESC+4, BUFPTR              ; 0939
                                           10  11 001B7         BRB     14$                                   ; 0934
                                50              63 9A 001B9 13$: MOVZBL  (R3), R0                          ; 0943
                     0000'      CF          50 A2 001BC      SUBW2   R0, INS$FAOBUFDESC                      ;
                                50              63 9A 001C1      MOVZBL  (R3), R0                           ; 0944
                     0000'      CF          50 C0 001C4      ADDL2   R0, INS$FAOBUFDESC+4                    ;
          FFC2       52                  02  17 F1 001C9 14$: ACBL    #23, #2, I, 12$                       ; 0927
                     0000'      CF          55 B0 001CF      MOVW    BUFLEN, INS$FAOBUFDESC                  ; 0948
                     0000'      CF          54 D0 001D4      MOVL    BUFPTR, INS$FAOBUFDESC+4                ; 0949
               03 00000000G     00          02 E0 001D9      BBS     #2, INS$GL_CTLMSK+1, 15$                ; 0955
                                      00B7  31 E1 001E1      BRW     24$                                     ;
                     0000V      CF          00 FB 001E4 15$: CALLS   #0, TERMINATE_LINE                     ; 0958
               13 00000000G     00          03 E1 001E9      BBC     #3, INS$GL_CTLMSK+1, 16$                ; 0960
                                7E        08    A8 9A 001F1      MOVZBL  11(R8), -(SP)                       ; 0963
                                7E        08    A8 3C 001F5      MOVZWL  8(R8), -(SP)                        ;
                                          58    DD 001F9         PUSHL   R8                                  ; 0962
                                0000'     CF    9F 001FB         PUSHAB  FAOCTL_KFEADR                       ;
                     0000V      CF          04 FB 001FF      CALLS   #4, FORMAT_TERMINATE_LINE               ;
                                6A          95 00204 16$:    TSTB    (R10)                                   ; 0965
                                0A          18 00206         BGEQ    17$                                     ;
                                7E        2A    A8 3C 00208      MOVZWL  42(R8), -(SP)                       ; 0967
                                0000'     CF    9F 0020C         PUSHAB  FAOCTL_COMPAT_TYP                   ;
                                07          11 00210         BRB     18$                                     ;
                                14        A8    DD 00212 17$:    PUSHL   20(R8)                              ; 0969
                                0000'     CF    9F 00215         PUSHAB  FAOCTL_USECNT                       ;
                     0000V      CF          02 FB 00219 18$:    CALLS   #2, FORMAT_TERMINATE_LINE           ;
               22                6A          03 E1 0021E      BBC     #3, (R10), 20$                          ; 0971
                                6A          95 00222         TSTB    (R10)                                   ; 0975
                                0D          18 00224         BGEQ    19$                                     ;
                                57          DD 00226         PUSHL   WCB_SHRCNT                              ;
                                0000'     CF    9F 00228         PUSHAB  FAOCTL_CMODCURR                     ;
                     0000V      CF          02 FB 0022C      CALLS   #2, FORMAT_TERMINATE_LINE               ;
                                11          11 00231         BRB     20$                                     ;
                                7E        34    A8 3C 00233 19$:    MOVZWL  52(R8), -(SP)                       ; 0978
                                6E          D7 00237         DECL    (SP)                                    ;
                                57          DD 00239         PUSHL   WCB_SHRCNT                              ;
```

```
                                0000'  CF  9F 0023B          PUSHAB  FAOCTL_SHRUSECNT                        ; 0977
                      0000V     CF         03  FB 0023F      CALLS   #3, FORMAT_TERMINATE_LINE               ;
                 0D             6A         05  E1 00244 20$: BBC     #5, (R10), 21$                        ; 0980
                                7E      12 A8  3C 00248      MOVZWL  18(R8), -(SP)                         ; 0982
                                0000'  CF  9F 0024C          PUSHAB  FAOCTL_GBLCNT                         ;
                      0000V     CF         02  FB 00250      CALLS   #2, FORMAT_TERMINATE_LINE              ;
                 32 00000000G   00         03  E1 00255 21$: BBC     #3, INS$GL_CTLMSK+1, 23$              ; 0984
                                           56  D5 0025D      TSTL    WCB                                   ;
                                           0D  13 0025F      BEQL    22$                                   ;
                                0840   8F  BB 00261          PUSHR   #^M<R6,R11>                           ; 0986
                                0000'  CF  9F 00265          PUSHAB  FAOCTL_WINDOW                         ;
                      0000V     CF         03  FB 00269      CALLS   #3, FORMAT_TERMINATE_LINE              ;
                 19 00000000G   00         03  E1 0026E 22$: BBC     #3, INS$GL_CTLMSK+1, 23$              ; 0988
                 15             6A         04  E1 00276      BBC     #4, (R10), 23$                        ;
                 50       1C    A8         0C  C3 0027A      SUBL3   #12, 28(R8), R0                       ; 0992
                                7E      08 A0  3C 0027F      MOVZWL  8(R0), -(SP)                          ; 0995
                                1C A8  DD  00283             PUSHL   28(R8)                                ;
                                0000'  CF  9F 00286          PUSHAB  FAOCTL_HEADER                         ; 0994
                      0000V     CF         03  FB 0028A      CALLS   #3, FORMAT_TERMINATE_LINE              ;
                 08             6A         02  E1 0028F 23$: BBC     #2, (R10), 24$                        ; 0998
                                20 A8  9F  00293             PUSHAB  32(R8)                                ; 1000
                      0000V     CF         01  FB 00296      CALLS   #1, PRINT_PRIVS                       ;
                      0000V     CF         00  FB 0029B 24$: CALLS   #0, TERMINATE_LINE                    ; 1004
                                50         01  D0 002A0      MOVL    #1, R0                                ; 1006
                                           04  002A3         RET                                           ; 1007

; Routine Size:  676 bytes,    Routine Base:  $CODE$ + 024C
```

```
;  818          1008  1
;  819          1009  1 %SBTTL 'PRINT_PRIVS';
;  820          1010  1
;  821          1011  1 ROUTINE PRINT_PRIVS (PRIV_ADR) =
;  822          1012  1 !+++
;  823          1013  1 !
;  824          1014  1 !  FUNCTIONAL DESCRIPTION:
;  825          1015  1 !       Print the ASCII symbol for each privilege bit set in the quadword
;  826          1016  1 !       privilege mask, priv_adr.
;  827          1017  1 !
;  828          1018  1 !  INPUT:
;  829          1019  1 !       priv_adr = address of quadword privilege mask
;  830          1020  1 !
;  831          1021  1 !
;  832          1022  2 BEGIN
;  833          1023  2 LOCAL
;  834          1024  2     PLACE_HLDR,
;  835          1025  2     PRVS_TO_PRINT,
;  836          1026  2     SYMBOL_LEN,
;  837          1027  2     PRIV_MSK;
;  838          1028  2
;  839          1029  2 PLACE_HLDR = PRV$AB_NAMES;              ! point to start of privilege name table
;  840          1030  2 PRVS_TO_PRINT = FALSE;                  ! record status of buffer
;  841          1031  2 FORMAT_LINE ( FAOCTL_PRIVHD );         ! init buffer with header info and indentation
;  842          1032  2
;  843          1033  2
;  844          1034  2 WHILE .(.PLACE_HLDR) <0,8> NEQ 0 DO    ! Traverse down the table
;  845          1035  3     BEGIN
;  846          1036  3     PLACE_HLDR = .PLACE_HLDR + 1;       ! Second byte is privilege mask
;  847          1037  3     PRIV_MSK = .(.PLACE_HLDR) <0,8>;
;  848          1038  3     PLACE_HLDR = .PLACE_HLDR + 1;       ! Third byte is ASCII string count
;  849          1039  3     SYMBOL_LEN =.(.PLACE_HLDR) <0,8>;
;  850          1040  3
;  851          1041  3     IF .(.PRIV_ADR) <.PRIV_MSK,1>       ! Check if bit is set in quadword
;  852          1042  3     THEN
;  853          1043  4         BEGIN
;  854          1044  4         !
;  855          1045  4         !    The bit is set, put ASCII in buffer
;  856          1046  4         !
;  857          1047  4         PRVS_TO_PRINT = TRUE;           ! Remember that something is in buffer
;  858          1048  4         FORMAT_LINE ( FAOCTL_PRIV, .PLACE_HLDR);
;  859          1049  4         IF INS$C_FAOBUFLEN - .INS$FAOBUFDESC [DSC$W_LENGTH] GTR 70
;  860          1050  4         THEN
;  861          1051  5             BEGIN
;  862          1052  5             !
;  863          1053  5             !   Avoid too long a line.  If it is, print what we have and
;  864          1054  5             !   start a new line with a blank header offset
;  865          1055  5             !
;  866          1056  5             TERMINATE_LINE ();
;  867          1057  5             PRVS_TO_PRINT = FALSE;              ! Currently no privs in buffer
;  868          1058  5             FORMAT_LINE ( FAOCTL_PRIVHD2 );
;  869          1059  4             END;
;  870          1060  3         END;
;  871          1061  3
;  872          1062  3     !
;  873          1063  3     !   skip past count byte and ASCII privilege symbol
;  874          1064  3     !
```

```
;   875          1065  3     PLACE_HLDR = .PLACE_HLDR + 1 + .SYMBOL_LEN;
;   876          1066  3
;   877          1067  2     END;        ! while
;   878          1068  2
;   879          1069  2
;   880          1070  2 IF .PRVS_TO_PRINT       ! If there is something other than the header in the buffer
;   881          1071  2 THEN TERMINATE_LINE ()  ! Then print it
;   882          1072  2 ELSE
;   883          1073  3     BEGIN              ! otherwise reset buffer to forget about unused priv header
;   884          1074  3     INS$FAOBUFDESC [DSC$W_LENGTH] = INS$C_FAOBUFLEN;
;   885          1075  3     INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOOUTBUF;
;   886          1076  2     END;
;   887          1077  2
;   888          1078  2 RETURN TRUE;
;   889          1079  1 END;    ! routine  print_privs


                                00FC 00000 PRINT_PRIVS:
                                           .WORD   Save R2,R3,R4,R5,R6,R7                          ; 1011
                     57     0000V  CF  9E 00002     MOVAB   FORMAT_LINE, R7                         ;
                     56     0000'  CF  9E 00007     MOVAB   INS$FAOBUFDESC, R6                      ;
                     52 00000000G  00  9E 0000C     MOVAB   PRV$AB_NAMES, PLACE_HLDR                ; 1029
                                54  D4 00013     CLRL    PRVS_TO_PRINT                           ; 1030
                            0000'  CF  9F 00015     PUSHAB  FAOCTL_PRIVHD                           ; 1031
                     67             01  FB 00019     CALLS   #1, FORMAT_LINE                         ;
                                    62  95 0001C 1$:  TSTB    (PLACE_HLDR)                            ; 1034
                                    3E  13 0001E     BEQL    3$                                      ;
                                    52  D6 00020     INCL    PLACE_HLDR                              ; 1036
                                55      82  9A 00022     MOVZBL  (PLACE_HLDR)+, PRIV_MSK                 ; 1037
                                53      62  9A 00025     MOVZBL  (PLACE_HLDR), SYMBOL_LEN                ; 1039
                         2A        04   BC  55  E1 00028     BBC     PRIV_MSK, @PRIV_ADR, 2$                 ; 1041
                                54      01  D0 0002D     MOVL    #1, PRVS_TO_PRINT                       ; 1047
                                        52  DD 00030     PUSHL   PLACE_HLDR                              ; 1048
                                    0000'  CF  9F 00032     PUSHAB  FAOCTL_PRIV                             ;
                                67      02  FB 00036     CALLS   #2, FORMAT_LINE                         ;
                                50      66  3C 00039     MOVZWL  INS$FAOBUFDESC, R0                      ; 1049
                                50        46   A0  9E 0003C     MOVAB   70(R0), R0                      ;
                        000000FF   8F       50  D1 00040     CMPL    R0, #255                        ;
                                            0E  18 00047     BGEQ    2$                              ;
                             0000V  CF      00  FB 00049     CALLS   #0, TERMINATE_LINE              ; 1056
                                            54  D4 0004E     CLRL    PRVS_TO_PRINT                   ; 1057
                                    0000'  CF  9F 00050     PUSHAB  FAOCTL_PRIVHD2                  ; 1058
                                67          01  FB 00054     CALLS   #1, FORMAT_LINE                 ;
                                52        01 A342  9E 00057 2$:     MOVAB   1(SYMBOL_LEN)[PLACE_HLDR], PLACE_HLDR   ; 1065
                                          BE  11 0005C     BRB     1$                              ; 1034
                                07              54  E9 0005E 3$:     BLBC    PRVS_TO_PRINT, 4$       ; 1070
                             0000V  CF          00  FB 00061     CALLS   #0, TERMINATE_LINE      ; 1071
                                            09  11 00066     BRB     5$                              ;
                                66        FF   8F  9B 00068 4$:     MOVZBW  #255, INS$FAOBUFDESC            ; 1074
                                  04    A6        FC   A6  D0 0006C     MOVL    INS$FAOOUTBUF, INS$FAOBUFDESC+4 ; 1075
                                50              01  D0 00071 5$:     MOVL    #1, R0                  ; 1078
                                                04  04 00074     RET                             ; 1079

; Routine Size:  117 bytes,    Routine Base:  $CODE$ + 04F0
```

```
;  890          1080  1
```

```
;  892          1081  1 %SBTTL  'ouput to temporary buffer routines';
;  893          1082  1
;  894          1083  1 ROUTINE FORMAT_LINE (FAO_STRING, PARAMETER_LIST) =
;  895          1084  2 BEGIN
;  896          1085  2 !+++
;  897          1086  2 !
;  898          1087  2 !  FUNCTIONAL DESCRIPTION:
;  899          1088  2 !       Format an ASCII string and stuff it into the output buffer.
;  900          1089  2 !       Update the buffer pointers to reflect the new stuff in the
;  901          1090  2 !       buffer.
;  902          1091  2 !
;  903          1092  2 !  INPUT:
;  904          1093  2 !       fao_string =    Formatted Ascii Output control string for FAO
;  905          1094  2 !       parameter_list= List of stuff to have formatted into buffer.
;  906          1095  2 !
;  907          1096  2 !  IMPLICIT INPUT:
;  908          1097  2 !       Output buffer has been allocated and ins$faobufdesc is the
;  909          1098  2 !       descriptor for it.
;  910          1099  2 !
;  911          1100  2 !  OUTPUT:
;  912          1101  2 !       none
;  913          1102  2 !
;  914          1103  2 !  ROUTINE VALUE
;  915          1104  2 !       Success, or error status from SYS$FAOL
;  916          1105  2 !---
;  917          1106  2 LOCAL
;  918          1107  2     OUTLEN : WORD;
;  919          1108  2
;  920          1109  2 EXECUTE ( SYS$FAOL (.FAO_STRING, OUTLEN, INS$FAOBUFDESC, PARAMETER_LIST));       ! Format the buffer
;  921          1110  2 INS$FAOBUFDESC [DSC$W_LENGTH] = .INS$FAOBUFDESC [DSC$W_LENGTH] - .OUTLEN;       ! decrement space left in bu
;  922          1111  2 INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOBUFDESC [DSC$A_POINTER] + .OUTLEN;     ! Point to unused space left
;  923          1112  2 RETURN TRUE;
;  924          1113  1 END;                ! routine  FORMAT_LINE


                                                0000 00000 FORMAT_LINE:
                                                           .WORD    Save nothing                                ; 1083
                                          5E               04  C2 00002          SUBL2   #4, SP                      ;
                                                     08    AC  9F 00005          PUSHAB  PARAMETER_LIST              ; 1109
                                                   0000'   CF  9F 00008          PUSHAB  INS$FAOBUFDESC              ;
                                                     08    AE  9F 0000C          PUSHAB  OUTLEN                      ;
                                                     04    AC  DD 0000F          PUSHL   FAO_STRING                  ;
                                  00000000G        00        04  FB 00012          CALLS   #4, SYS$FAOL                ;
                                                   10        50  E9 00019          BLBC    STATUS, 1$                  ;
                                       0000'       CF        6E  A2 0001C          SUBW2   OUTLEN, INS$FAOBUFDESC      ; 1110
                                                   50        6E  3C 00021          MOVZWL  OUTLEN, R0                  ; 1111
                                       0000'       CF        50  C0 00024          ADDL2   R0, INS$FAOBUFDESC+4        ;
                                                   50        01  D0 00029          MOVL    #1, R0                      ; 1112
                                                             04  0002C 1$:        RET                                 ; 1113

; Routine Size:  45 bytes,    Routine Base:  $CODE$ + 0565

;  925          1114  1
```

```
;  927          1115  1 ROUTINE TERMINATE_LINE : NOVALUE =
;  928          1116  2 BEGIN
;  929          1117  2 !+++
;  930          1118  2 !
;  931          1119  2 !  FUNCTIONAL DESCRIPTION:
;  932          1120  2 !       Print the contents of the output buffer to sys$output and re-initialize
;  933          1121  2 !       the descriptor of the buffer, and zero the buffer.
;  934          1122  2 !
;  935          1123  2 !  INPUT:
;  936          1124  2 !       none
;  937          1125  2 !
;  938          1126  2 !  IMPLICIT INPUT:
;  939          1127  2 !       Output buffer has been allocated and ins$faobufdesc is the
;  940          1128  2 !       descriptor for it.
;  941          1129  2 !
;  942          1130  2 !  OUTPUT:
;  943          1131  2 !       Output the contents of ins$faooutbuf to sys$output
;  944          1132  2 !
;  945          1133  2 !  ROUTINE VALUE
;  946          1134  2 !       status from $PUT
;  947          1135  2 !---
;  948          1136  2
;  949          1137  2 LOCAL
;  950          1138  2     LINE_LEN;
;  951          1139  2
;  952          1140  2 LINE_LEN = INS$C_FAOBUFLEN - .INS$FAOBUFDESC [DSC$W_LENGTH];
;  953          1141  2 TMPBUF_PTR [0,0,8,0] = .LINE_LEN;
;  954          1142  2 TMPBUF_PTR = .TMPBUF_PTR + 1;
;  955          1143  2 CH$MOVE (.LINE_LEN, .INS$FAOOUTBUF, .TMPBUF_PTR);
;  956          1144  2 TMPBUF_PTR = .TMPBUF_PTR + .LINE_LEN;
;  957          1145  2
;  958          1146  2
;  959          1147  2 INS$FAOBUFDESC [DSC$W_LENGTH] = INS$C_FAOBUFLEN;
;  960          1148  2 INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOOUTBUF;
;  961          1149  2 CH$FILL (%C' ', INS$C_FAOBUFLEN, .INS$FAOOUTBUF);
;  962          1150  2 RETURN;
;  963          1151  1 END;                 ! Routine  TERMINATE_LINE

                                    03FC 00000 TERMINATE_LINE:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                    ;  1115
                                59      0000'   CF  9E  00002   MOVAB   INS$FAOBUFDESC, R9                     ;
                                58      0000'   CF  9E  00007   MOVAB   TMPBUF_PTR, R8                         ;
                                56              69  3C  0000C   MOVZWL  INS$FAOBUFDESC, LINE_LEN               ;  1140
                     56 000000FF    8F          56  C3  0000F   SUBL3   LINE_LEN, #255, LINE_LEN              ;
                           00       B8          56  90  00017   MOVB    LINE_LEN, @TMPBUF_PTR                  ;  1141
                                                68  D6  0001B   INCL    TMPBUF_PTR                             ;  1142
                                57      FC      A9  D0  0001D   MOVL    INS$FAOOUTBUF, R7                      ;  1143
                        00      B8              67  56  28  00021   MOVC3   LINE_LEN, (R7), @TMPBUF_PTR      ;
                                                68  56  C0  00026   ADDL2   LINE_LEN, TMPBUF_PTR             ;  1144
                                69      FF      8F  9B  00029   MOVZBW  #255, INS$FAOBUFDESC                 ;  1147
                                   04           A9  57  D0  0002D   MOVL    R7, INS$FAOBUFDESC+4            ;  1148
        00FF    8F                  20          6E  00  2C  00031   MOVC5   #0, (SP), #32, #255, (R7)         ;  1149
                                                    67      00038   
```

```
                                        04 00039          RET                                                    ; 1151

; Routine Size:  58 bytes,    Routine Base:  $CODE$ + 0592

;  964          1152  1
```

```
;  966          1153  1 ROUTINE FORMAT_TERMINATE_LINE (FAO_STRING,PARAMETER_LIST) : NOVALUE =
;  967          1154  2 BEGIN
;  968          1155  2 !+++
;  969          1156  2 ! FUNCTIONAL DESCRIPTION:
;  970          1157  2 !
;  971          1158  2 !       Call FORMAT_LINE to format the line, then call TERMINATE_LINE to
;  972          1159  2 !       terminate the line.
;  973          1160  2 !
;  974          1161  2 !---
;  975          1162  2 BUILTIN
;  976          1163  2     CALLG,
;  977          1164  2     AP;
;  978          1165  2
;  979          1166  2 CALLG(.AP,FORMAT_LINE);
;  980          1167  2 TERMINATE_LINE();
;  981          1168  2 RETURN;
;  982          1169  1 END;


                                0000 00000 FORMAT_TERMINATE_LINE:
                                           .WORD   Save nothing                    ; 1153
                    93   AF        6C  FA 00002   CALLG   (AP), FORMAT_LINE       ; 1166
                    BC   AF        00  FB 00006   CALLS   #0, TERMINATE_LINE      ; 1167
                                      04 0000A   RET                              ; 1169

; Routine Size:  11 bytes,    Routine Base:  $CODE$ + 05CC
```

```
;  984          1170  1 ROUTINE PRINTOUT =
;  985          1171  2 BEGIN
;  986          1172  2 !+++
;  987          1173  2 !
;  988          1174  2 !   FUNCTIONAL DESCRIPTION:
;  989          1175  2 !       Print the contents of the temporary buffer to sys$output
;  990          1176  2 !
;  991          1177  2 !   INPUT:
;  992          1178  2 !       none
;  993          1179  2 !
;  994          1180  2 !   IMPLICIT INPUT:
;  995          1181  2 !       Output buffer has been allocated and ins$faobufdesc is the
;  996          1182  2 !       descriptor for it.
;  997          1183  2 !
;  998          1184  2 !   OUTPUT:
;  999          1185  2 !       Output the contents of ins$faooutbuf to sys$output
; 1000          1186  2 !
; 1001          1187  2 !   ROUTINE VALUE
; 1002          1188  2 !       status from $PUT
; 1003          1189  2 !---
; 1004          1190  2
; 1005          1191  2 LOCAL
; 1006          1192  2     TMPBUF_USELEN,
; 1007          1193  2     STATUS;
; 1008          1194  2
; 1009          1195  2 TMPBUF_USELEN = .TMPBUF_PTR - .TMPBUF;
; 1010          1196  2 TMPBUF_PTR = .TMPBUF;
; 1011          1197  2
; 1012          1198  2 WHILE .TMPBUF_PTR - .TMPBUF LSS .TMPBUF_USELEN DO
; 1013          1199  3     BEGIN
; 1014          1200  3     LOCAL
; 1015          1201  3         SIZE;
; 1016          1202  3
; 1017          1203  3     SIZE = .(.TMPBUF_PTR) <0,8,0>;
; 1018          1204  3     INS$G_OUTRAB [RAB$W_RSZ] = .SIZE;
; 1019          1205  3     INS$G_OUTRAB [RAB$L_RBF] = .TMPBUF_PTR+1;
; 1020          1206  3     EXECUTE ($PUT (RAB = INS$G_OUTRAB));
; 1021          1207  3     TMPBUF_PTR = .TMPBUF_PTR + 1 + .SIZE;
; 1022          1208  2     END;
; 1023          1209  2
; 1024          1210  2 RETURN TRUE;
; 1025          1211  1 END;              ! Routine  PRINTOUT


                                                        .EXTRN  SYS$PUT

                                        003C 00000 PRINTOUT:
                                                        .WORD   Save R2,R3,R4,R5                                ; 1170
                              55 00000000G   00  9E 00002       MOVAB   INS$G_OUTRAB+34, R5                     ;
                              54     0000'   CF  9E 00009       MOVAB   TMPBUF_PTR, R4                          ;
                    53        64       FC   A4  C3 0000E       SUBL3   TMPBUF, TMPBUF_PTR, TMPBUF_USELEN       ; 1195
                              64       FC   A4  D0 00013       MOVL    TMPBUF, TMPBUF_PTR                      ; 1196
                              51            64  D0 00017 1$:    MOVL    TMPBUF_PTR, R1                          ; 1198
                    50        51       FC   A4  C3 0001A       SUBL3   TMPBUF, R1, R0                          ;
                              53            50  D1 0001F       CMPL    R0, TMPBUF_USELEN                       ;
                                            22  18 00022       BGEQ    2$                                      ;
```

```
                                    52              61  9A 00024          MOVZBL  (R1), SIZE                                      ; 1203
                                    65              52  B0 00027          MOVW    SIZE, INS$G_OUTRAB+34                           ; 1204
                                06  A5      01     A1  9E 0002A          MOVAB   1(R1), INS$G_OUTRAB+40                          ; 1205
                                        DE      A5  9F 0002F          PUSHAB  INS$G_OUTRAB                                    ; 1206
                           00000000G    00          01  FB 00032          CALLS   #1, SYS$PUT                                     ;
                                    0D              50  E9 00039          BLBC    STATUS, 3$                                      ;
                            50          64          52  C1 0003C          ADDL3   SIZE, TMPBUF_PTR, R0                            ; 1207
                                        64      01  A0  9E 00040          MOVAB   1(R0), TMPBUF_PTR                               ;
                                                    D1  11 00044          BRB     1$                                              ; 1198
                                        50          01  D0 00046 2$:      MOVL    #1, R0                                          ; 1210
                                                        04 00049 3$:      RET                                                     ; 1211

; Routine Size:  74 bytes,    Routine Base:  $CODE$ + 05D7

; 1026          1212  1
; 1027          1213  1
; 1028          1214  1 END                 ! Module inslist
; 1029          1215  0 ELUDOM



                                                                  .EXTRN  LIB$SIGNAL

;                                   PSECT SUMMARY
;
;       Name                        Bytes                       Attributes
;
;  $GLOBAL$                             12  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                               144  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                              672  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                             1569  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  .  ABS  .                             0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)



;                               Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619        74         0        1000        00:01.8




;                                   COMMAND QUALIFIERS

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

; Size:         1569 code + 828 data bytes
; Run Time:       00:31.3
```

```
; Elapsed Time:    01:38.9
; Lines/CPU Min:    2326
; Lexemes/CPU-Min: 19941
; Memory Used:  261 pages
; Compilation Complete
```

INSMAIN
LIS

INSDELETE
LIS

INSOLDCMD
LIS

INSPRSOLD
LIS

INSGLOBAL
LIS

INSKFSCAN
LIS

INSLIST
LIS

INSMSG
LIS