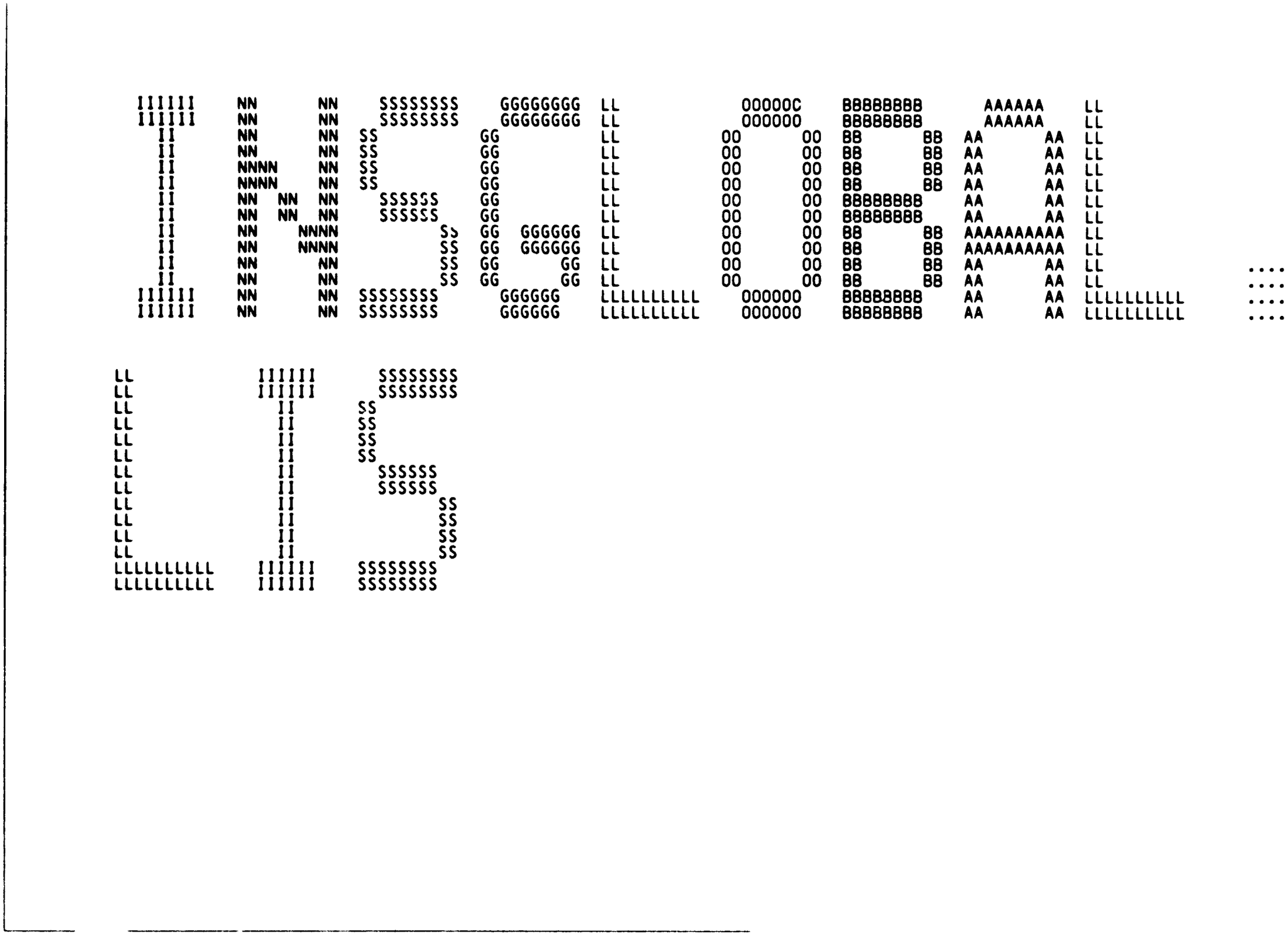
```
IIIIIIIII      NNN          NNN     SSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA     LLL
IIIIIIIII      NNN          NNN     SSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA     LLL
IIIIIIIII      NNN          NNN     SSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA     LLL
   III         NNN          NNN  SSS                   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 GGGGGGGG LL OOOOOO BBBBBBBB AAAAAA LL
IIIIII NN NN SSSSSSSS GGGGGGGG LL OOOOOO BBBBBBBB AAAAAA LL
II NN NN SS GG LL OO OO BB BB AA AA LL
II NN NN SS GG LL OO OO BB BB AA AA LL
II NNNN NN SS GG LL OO OO BB BB AA AA LL
II NNNN NN SS GG LL OO OO BB BB AA AA LL
II NN NN NN SSSSSS GG LL OO OO BBBBBBBB AA AA LL
II NN NN NN SSSSSS GG LL OO OO BBBBBBBB AA AA LL
II NN NNNN SS GG GGGGGG LL OO OO BB BB AAAAAAAAAA LL
II NN NNNN SS GG GGGGGG LL OO OO BB BB AAAAAAAAAA LL
II NN NN SS GG GG LL OO OO BB BB AA AA LL ....
II NN NN SS GG GG LL OO OO BB BB AA AA LL ....
IIIIII NN NN SSSSSSSS GGGGGG LLLLLLLLLL OOOOOO BBBBBBBB AA AA LLLLLLLLLL ....
IIIIII NN NN SSSSSSSS GGGGGG LLLLLLLLLL OOOOOO BBBBBBBB AA AA LLLLLLLLLL ....
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 INSGLOBAL (                                  ! List Global Sections
;     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 !       This module controls listing.
;    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, September 1981
;    43          0043  1 !
;    44          0044  1 ! Modified by:
;    45          0045  1 !
;    46          0046  1 !       V03-007 MSH0035         Michael S. Harvey       20-Apr-1984
;    47          0047  1 !               Correctly scan shared memory GSD cache so that all
;    48          0048  1 !               shared memories are included in the /GLOBAL display.
;    49          0049  1 !               Also, print out the global section summary for each
;    50          0050  1 !               shared memory instead of only the last shared memory
;    51          0051  1 !               displayed. Also, print out only the relevant shared
;    52          0052  1 !               memory global sections when a specific known file was
;    53          0053  1 !               specified. Also, prevent global section summary message
;    54          0054  1 !               from wrapping around the screen.
;    55          0055  1 !
;    56          0056  1 !       V03-006 MSH0020         Michael S. Harvey       8-Mar-1984
;    57          0057  1 !               Use correct length when extracting shared memory name
```

```
;    58          0058  1 !                 from the common data page.
;    59          0059  1 !
;    60          0060  1 !       V03-005 MSH0019         Michael S. Harvey       8-Mar-1984
;    61          0061  1 !                 Access KFE from EXEC mode instead of USER mode to
;    62          0062  1 !                 avoid an ACCVIO.
;    63          0063  1 !
;    64          0064  1 !       V03-004 MSH0004         Michael S. Harvey       26-Jan-1984
;    65          0065  1 !                 Add support for lengthened global section names in GSDs.
;    66          0066  1 !
;    67          0067  1 !       V03-003 BLS0256         Benn Schreiber          27-Dec-1983
;    68          0068  1 !                 Cleanup.  Allocate buffers once in main routine.
;    69          0069  1 !
;    70          0070  1 !       V03-002 RPG0002         Bob Grosso              21-Aug-1983
;    71          0071  1 !                 Restrict header printing in no sections to report.
;    72          0072  1 !
;    73          0073  1 !       V03-001 RPG0001         Bob Grosso              July 20, 1983
;    74          0074  1 !                 Print global sections by KFE if requested.
;    75          0075  1 !                 Extensive cleaning up of code.
;    76          0076  1 !
;    77          0077  1 !--
;    78          0078  1
;    79          0079  1 !
;    80          0080  1 ! Include files
;    81          0081  1 !
;    82          0082  1
;    83          0083  1 LIBRARY 'SYS$LIBRARY:LIB.L32';          ! VAX/VMS system definitions
;    84          0084  1
;    85          0085  1 REQUIRE
;    86          0086  1         'SRC$:INSPREFIX.REQ';
;    87          0228  1 REQUIRE
;    88          0229  1         'LIB$:INSDEF.R32';
;    89          0288  1
```

```
;    91          0289  1 %SBTTL  'Declarations';
;    92          0290  1 !
;    93          0291  1 ! Table of contents
;    94          0292  1 !
;    95          0293  1
;    96          0294  1 LINKAGE
;    97          0295  1     CALL_S_4 = CALL (STANDARD, REGISTER=4),     ! for calling copy routines which execute in kernel mode
;    98          0296  1     JSB_0_4 = JSB (REGISTER=0, REGISTER=4) :    ! For mutex handling routines
;    99          0297  1                 NOPRESERVE (1,2,3);
;   100          0298  1
;   101          0299  1 LINKAGE
;   102          0300  1     JSB_4_5_G6_10 = JSB (REGISTER = 4, REGISTER = 5, REGISTER = 10) :
;   103          0301  1                         GLOBAL (GSD_PTR = 6) NOPRESERVE (2,3);
;   104          0302  1
;   105          0303  1 !
;   106          0304  1 ! External routines
;   107          0305  1 !
;   108          0306  1
;   109          0307  1 EXTERNAL ROUTINE
;   110          0308  1     LIB$GET_VM,                                 ! Allocate buffer in virtual memory
;   111          0309  1     LIB$FREE_VM,                                ! Deallocate buffer in virtual memory
;   112          0310  1     LIB$PUT_OUTPUT,                             ! Print contents of formatted buffer
;   113          0311  1     MMG$GETNXTGSD : JSB_4_5_G6_10,              ! get address of next used GSD in memory
;   114          0312  1     MMG$VALIDATEGSD : JSB_4_5_G6_10,            ! Check if GSD is in use
;   115          0313  1     SCH$LOCKR   : JSB_0_4,                      ! Lock a Mutex for read access
;   116          0314  1     SCH$UNLOCK  : JSB_0_4,                      ! Unlock a Mutex
;   117          0315  1     SYS$FAOL : ADDRESSING_MODE (GENERAL);       ! Format ASCII output
;   118          0316  1
;   119          0317  1 EXTERNAL
;   120          0318  1     EXE$GL_GSDDELFL,                            ! delete pending list
;   121          0319  1     EXE$GL_GSDGRPFL,                            ! group list
;   122          0320  1     EXE$GL_GSDSYSFL,                            ! system list
;   123          0321  1     EXE$GL_SHBLIST,                             ! SHB listhead
;   124          0322  1     EXE$GL_GSDMTX,                              ! Global Section Descriptor Mutex
;   125          0323  1     INS$FAOOUTBUF,                              ! address of output buffer
;   126          0324  1     INS$FAOBUFDESC : BBLOCK [DSC$C_S_BLN],
;   127          0325  1     INS$G_OUTRAB : BBLOCK,                      ! Output record access block
;   128          0326  1     MMG$GL_SYSPHD,                              ! address of system process header
;   129          0327  1     SGN$GL_MAXGPGCT;                            ! Maximum global page count
;   130          0328  1
;   131          0329  1 FORWARD ROUTINE
;   132          0330  1     COPY_LISTS,                                 ! Kernel mode routine dispatcher
;   133          0331  1     COPY_GSD_LISTS : CALL_S_4,                  ! Copy Global Section Descriptors to user buffer
;   134          0332  1     LIST_GSDS,                                  ! list Global Section Descriptors
;   135          0333  1     COPY_SHM_LISTS : CALL_S_4,                  ! Copy shared memory lists to user buffer
;   136          0334  1     LIST_SHMS,                                  ! list shared memory lists
;   137          0335  1     DECODE_FLAGS,                               ! Decode flags and buffer their ASCII symbols
;   138          0336  1     EXEC_MODE_NAME_CHK,                         ! KFE and GSD name comparison routine
;   139          0337  1     INS$FAOL,                                   ! Format ASCII output line
;   140          0338  1     INS$OUTPUT_FAOBUF,                          ! Print contents of formatted buffer
;   141          0339  1     INS$FAO_AND_OUT;                            ! Format and output ascii line
;   142          0340  1
;   143          0341  1 OWN
;   144          0342  1     KFE,                                        ! KFE address of shared file for which
;   145          0343  1                                                 !  global sections are to be listed
;   146          0344  1     INS$A_BUFFER : VECTOR [2,LONG],             ! User buffer for copying GSD's to while in kernel mode
;   147          0345  1                                                 ! so the information can be formatted later while in user mode
```

```
;  148          0346  1      INS$L_BUFSIZ,                       ! Last size tried for buffer.
;  149          0347  1      GSD$W_SIZE2 : WORD INITIAL ($BYTEOFFSET(GSD$W_SIZE) + 2),
;  150          0348  1      MODEADDR : BYTE INITIAL (0);
;  151          0349  1
;  152          0350  1  EXTERNAL LITERAL
;  153          0351  1      INS$C_FAOBUFLEN;                    ! Size of the output print buffer
;  154          0352  1
;  155          0353  1  LITERAL
;  156          0354  1      INS$C_BUFSIZ = 8,                   ! size for first attempt to buffer kfi list
;  157          0355  1      INS$C_CTLFLGSTR = 6,                ! Number of entries in flags string array
;  158          0356  1      INS$C_NUMGSDLSTS = 4;               ! Four lists; system, group, delete pending and multiported memory
;  159          0357  1
;  160          0358  1  BUILTIN
;  161          0359  1      PROBEW;                             ! Check that buffer is writeable so we don't take an access violatio
;  162          0360  1
;  163          0361  1  BIND
;  164          0362  1      !
;  165          0363  1      !   FAO control strings to format the data
;  166          0364  1      !
;  167          0365  1      FAO_SYSTEM          = $DESCRIPTOR ('    '),
;  168          0366  1      FAO_GROUP           = $DESCRIPTOR ('=!OB'),
;  169          0367  1      FAO_PAGCNT          = $DESCRIPTOR (' Pagcnt/Refcnt=!UL/!UL'),
;  170          0368  1      FAO_PAGPFN          = $DESCRIPTOR (' Pagcnt/Basepfn=!UL/!XL'),
;  171          0369  1      FAO_SUMMARY         = $DESCRIPTOR ('!/    !UL Global Section!%S Used,  !UL/!UL Global Pages Used/Unused'
;  172          0370  1      FAO_SYSTITLE        = $DESCRIPTOR (' !/!_System Global Sections!/'),
;  173          0371  1      FAO_GRPTITLE        = $DESCRIPTOR (' !/!_Group Global Sections!/'),
;  174          0372  1      FAO_DELTITLE        = $DESCRIPTOR (' !/!_Delete Pending Global Sections!/');
;  175          0373  1
;  176          0374  1
```

```
;  178          0375  1 %SBTTL  'INS$GLOBAL';
;  179          0376  1
;  180          0377  1 GLOBAL ROUTINE INS$GLOBAL ( GLOBAL_KFE ) =
;  181          0378  2 BEGIN
;  182          0379  2 !+++
;  183          0380  2 !
;  184          0381  2 !  FUNCTIONAL DESCRIPTION:
;  185          0382  2 !
;  186          0383  2 !       Process the qualifier /GLOBAL by printing the system, group, delete
;  187          0384  2 !       pending and multiport memory global section lists.  Enter kernel mode
;  188          0385  2 !       to access the lists, copying them to a user buffer, which can then be
;  189          0386  2 !       formatted for output.  Print the output to sys$output.
;  190          0387  2 !
;  191          0388  2 !  INPUT:
;  192          0389  2 !
;  193          0390  2 !       none
;  194          0391  2 !
;  195          0392  2 !  IMPLICITS INPUT:
;  196          0393  2 !
;  197          0394  2 !       An output channel to sys$output has been opened and ins$g_outrab
;  198          0395  2 !       locates the record access block.
;  199          0396  2 !
;  200          0397  2 !  OUTPUT:
;  201          0398  2 !
;  202          0399  2 !       The formatted global section lists to sys$output.
;  203          0400  2 !
;  204          0401  2 !  ROUTINE VALUE:
;  205          0402  2 !
;  206          0403  2 !       Success or a service error.
;  207          0404  2 !---
;  208          0405  2 BUILTIN
;  209          0406  2     NULLPARAMETER;                      ! Check if parameter was included
;  210          0407  2
;  211          0408  2 LOCAL
;  212          0409  2     STATUS;
;  213          0410  2
;  214          0411  2 IF NOT NULLPARAMETER (1)
;  215          0412  2 THEN KFE = .GLOBAL_KFE
;  216          0413  2 ELSE KFE = 0;
;  217          0414  2
;  218          0415  2 STATUS = COPY_LISTS (COPY_GSD_LISTS);   ! Copy the system, group and delete pending global section lists fro
;  219          0416  2                                         ! space to the user buffer.
;  220          0417  2 IF NOT .STATUS THEN RETURN .STATUS;
;  221          0418  2
;  222          0419  2 !
;  223          0420  2 !   Initialize the buffer and its descriptor
;  224          0421  2 !
;  225          0422  2 CH$FILL (%C' ',INS$C_FAOBUFLEN, .INS$FAOOUTBUF);
;  226          0423  2 INS$FAOBUFDESC [DSC$W_LENGTH] = INS$C_FAOBUFLEN;
;  227          0424  2 INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOOUTBUF;
;  228          0425  2 INS$G_OUTRAB [RAB$L_RBF] = .INS$FAOOUTBUF;
;  229          0426  2
;  230          0427  2
;  231          0428  2 STATUS = LIST_GSDS (.INS$A_BUFFER);      ! Format the info in the user buffer and print it out.
;  232          0429  2 IF NOT .STATUS
;  233          0430  2 THEN
;  234          0431  2     RETURN .STATUS;
```

```
;  235          0432  2
;  236          0433  2
;  237          0434  2
;  238          0435  2 !
;  239          0436  2 !       Print Shared Memory Sections
;  240          0437  2 !
;  241          0438  2 STATUS = COPY_LISTS (COPY_SHM_LISTS);   ! Copy the multiport memory global section lists to user buffer.
;  242          0439  2 IF NOT .STATUS
;  243          0440  2 THEN
;  244          0441  2     RETURN .STATUS;
;  245          0442  2
;  246          0443  2 !
;  247          0444  2     Initialize the buffer and its descriptor
;  248          0445  2 !
;  249          0446  2 CH$FILL (%C' ',INS$C_FAOBUFLEN, .INS$FAOOUTBUF);
;  250          0447  2 INS$FAOBUFDESC [DSC$W_LENGTH] = INS$C_FAOBUFLEN;
;  251          0448  2 INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOOUTBUF;
;  252          0449  2 INS$G_OUTRAB [RAB$L_RBF] = .INS$FAOOUTBUF;
;  253          0450  2
;  254          0451  2 STATUS = LIST_SHMS (.INS$A_BUFFER);     ! Format the info in the user buffer and print it out.
;  255          0452  2
;  256          0453  2 RETURN .STATUS;
;  257          0454  1 END;    ! routine  INS$GLOBAL

                                                                .TITLE  INSGLOBAL
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                  20 20 20 20   00000 P.AAB:  .ASCII  \    \                                    ;
                                                     00000004   00004 P.AAA:  .LONG   4                                        ;
                                                     00000000'  00008         .ADDRESS P.AAB                                  ;
                                                  42 4F 21 3D   0000C P.AAD:  .ASCII  \=!OB\                                  ;
                                                     00000004   00010 P.AAC:  .LONG   4                                        ;
                                                     00000000'  00014         .ADDRESS P.AAD                                  ;
3D 74 6E 63 66 65 52 2F 74 6E 63 67 61 50 20   00018 P.AAF:  .ASCII  \ Pagcnt/Refcnt=!UL/!UL\                                ;
                        4C 55 21 2F 4C 55 21   00027                                                                         ;
                                               0002E         .BLKB   2                                                       ;
                                                     00000016   00030 P.AAE:  .LONG   22                                       ;
                                                     00000000'  00034         .ADDRESS P.AAF                                  ;
6E 66 70 65 73 61 42 2F 74 6E 63 67 61 50 20   00038 P.AAH:  .ASCII  \ Pagcnt/Basepfn=!UL/!XL\                               ;
                        4C 58 21 2F 4C 55 21 3D   00047                                                                      ;
                                               0004F         .BLKB   1                                                       ;
                                                     00000017   00050 P.AAG:  .LONG   23                                       ;
                                                     00000000'  00054         .ADDRESS P.AAH                                  ;
61 62 6F 6C 47 20 4C 55 21 20 20 20 20 2F 21   00058 P.AAJ:  .ASCII  \!/    !UL Global Section!%S Used,  !UL/!\                 ;
73 55 20 53 25 21 6E 6F 69 74 63 65 53 20 6C   00067                                                                         ;
               21 2F 4C 55 21 20 20 2C 64 65   00076                                                                         ;
73 65 67 61 50 20 6C 61 62 6F 6C 47 20 4C 55   00080         .ASCII  \UL Global Pages Used/Unused\                           ;
            64 65 73 75 6E 55 2F 64 65 73 55 20   0008F                                                                      ;
                                               0009B         .BLKB   1                                                       ;
                                                     00000043   0009C P.AAI:  .LONG   67                                       ;
                                                     00000000'  000A0         .ADDRESS P.AAJ                                  ;
6F 6C 47 20 6D 65 74 73 79 53 5F 21 2F 21 20   000A4 P.AAL:  .ASCII  \ !/!_System Global Sections!/\                         ;
   2F 21 73 6E 6F 69 74 63 65 53 20 6C 61 62   000B3                                                                         ;
                                               000C1         .BLKB   3
```

```
                                                0000001D  000C4 P.AAK:   .LONG    29                                      ;
                                                00000000' 000C8          .ADDRESS P.AAL                                   ;
62 6F 6C 47 20 70 75 6F 72 47 5F 21 2F 21 20    000CC P.AAN:   .ASCII   \ !/!_Group Global Sections!/\                   ;
       2F 21 73 6E 6F 69 74 63 65 53 20 6C 61   000DB                                                                   ;
                                                0000001C  000E8 P.AAM:   .LONG    28                                    ;
                                                00000000' 000EC          .ADDRESS P.AAN                                   ;
6E 65 50 20 65 74 65 6C 65 44 5F 21 2F 21 20    000F0 P.AAP:   .ASCII   \ !/!_Delete Pending Global Sections!/\          ;
63 65 53 20 6C 61 62 6F 6C 47 20 67 6E 69 64    000FF                                                                   ;
                        2F 21 73 6E 6F 69 74    0010E                                                                   ;
                                                00115          .BLKB    3                                               ;
                                                00000025  00118 P.AAO:   .LONG    37                                    ;
                                                00000000' 0011C          .ADDRESS P.AAP                                   ;

                                                                .PSECT   $OWN$,NOEXE,2

                                                00000 KFE:      .BLKB    4
                                                00004 INS$A_BUFFER:
                                                                .BLKB    8
                                                0000C INS$L_BUFSIZ:
                                                                .BLKB    4
                                             000A  00010 GSD$W_SIZE2:
                                                                .WORD    10                                              ;
                                               00   00012 MODEADDR:
                                                                .BYTE    0                                               ;

                                                        FAO_SYSTEM=          P.AAA
                                                        FAO_GROUP=           P.AAC
                                                        FAO_PAGCNT=          P.AAE
                                                        FAO_PAGPFN=          P.AAG
                                                        FAO_SUMMARY=         P.AAI
                                                        FAO_SYSTITLE=        P.AAK
                                                        FAO_GRPTITLE=        P.AAM
                                                        FAO_DELTITLE=        P.AAO
                                                                .EXTRN   LIB$GET_VM, LIB$FREE_VM
                                                                .EXTRN   LIB$PUT_OUTPUT, MMG$GETNXTGSD
                                                                .EXTRN   MMG$VALIDATEGSD
                                                                .EXTRN   SCH$LOCKR, SCH$UNLOCK
                                                                .EXTRN   SYS$FAOL, EXE$GL_GSDDELFL
                                                                .EXTRN   EXE$GL_GSDGRPFL
                                                                .EXTRN   EXE$GL_GSDSYSFL
                                                                .EXTRN   EXE$GL_SHBLIST, EXE$GL_GSDMTX
                                                                .EXTRN   INS$FAOOUTBUF, INS$FAOBUFDESC
                                                                .EXTRN   INS$G_OUTRAB, MMG$GL_SYSPHD
                                                                .EXTRN   SGN$GL_MAXGPGCT
                                                                .EXTRN   INS$C_FAOBUFLEN

                                                                .PSECT   $CODE$,NOWRT,2

                                                        0FFC 00000      .ENTRY   INS$GLOBAL, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0377
                                                                                 R10,R11                                    ;
5B 00000000G   00   9E 00002                           MOVAB    INS$G_OUTRAB+40, R11                                    ;
5A 00000000G   8F   D0 00009                           MOVL     #INS$C_FAOBUFLEN, R10                                   ;
59 00000000G   00   9E 00010                           MOVAB    INS$FAOOUTBUF, R9                                       ;
58 00000000G   00   9E 00017                           MOVAB    INS$FAOBUFDESC, R8                                      ;
               6C   95 0001E                           TSTB     (AP)                                                    ; 0411
               0D   13 00020                           BEQL     1$                                                      ;
         04    AC   D5 00022                           TSTL     4(AP)                                                   ;
```

```
                                08  13 00025          BEQL    1$                              ;
              0000'  CF     04  AC  D0 00027          MOVL    GLOBAL_KFE, KFE                 ; 0412
                                04  11 0002D          BRB     2$                              ;
                         0000'  CF  D4 0002F 1$:      CLRL    KFE                             ; 0413
                         0000V  CF  9F 00033 2$:      PUSHAB  COPY_GSD_LISTS                  ; 0415
              0000V  CF         01  FB 00037          CALLS   #1, COPY_LISTS                  ;
                     57         50  D0 0003C          MOVL    R0, STATUS                      ;
                     50         57  E9 0003F          BLBC    STATUS, 3$                      ; 0417
                     56         69  D0 00042          MOVL    INS$FAOOUTBUF, R6               ; 0422
      5A        20   6E         00  2C 00045          MOVC5   #0, (SP), #32, R10, (R6)        ;
                                66     0004A                                                  ;
                     68         5A  B0 0004B          MOVW    R10, INS$FAOBUFDESC             ; 0423
                04   A8         56  D0 0004E          MOVL    R6, INS$FAOBUFDESC+4            ; 0424
                     6B         56  D0 00052          MOVL    R6, INS$G_OUTRAB+40             ; 0425
                         0000'  CF  DD 00055          PUSHL   INS$A_BUFFER                    ; 0428
              0000V  CF         01  FB 00059          CALLS   #1, LIST_GSDS                   ;
                     57         50  D0 0005E          MOVL    R0, STATUS                      ;
                     2E         57  E9 00061          BLBC    STATUS, 3$                      ; 0429
                         0000V  CF  9F 00064          PUSHAB  COPY_SHM_LISTS                  ; 0438
              0000V  CF         01  FB 00068          CALLS   #1, COPY_LISTS                  ;
                     57         50  D0 0006D          MOVL    R0, STATUS                      ;
                     1F         57  E9 00070          BLBC    STATUS, 3$                      ; 0439
                     56         69  D0 00073          MOVL    INS$FAOOUTBUF, R6               ; 0446
      5A        20   6E         00  2C 00076          MOVC5   #0, (SP), #32, R10, (R6)        ;
                                66     0007B                                                  ;
                     68         5A  B0 0007C          MOVW    R10, INS$FAOBUFDESC             ; 0447
                04   A8         56  D0 0007F          MOVL    R6, INS$FAOBUFDESC+4            ; 0448
                     6B         56  D0 00083          MOVL    R6, INS$G_OUTRAB+40             ; 0449
                         0000'  CF  DD 00086          PUSHL   INS$A_BUFFER                    ; 0451
              0000V  CF         01  FB 0008A          CALLS   #1, LIST_SHMS                   ;
                     57         50  D0 0008F          MOVL    R0, STATUS                      ;
                     50         57  D0 00092 3$:      MOVL    STATUS, R0                      ; 0453
                                    04 00095          RET                                     ; 0454

; Routine Size:  150 bytes,    Routine Base:  $CODE$ + 0000

;  258          0455  1
```

```
;  260          0456  1 %SBTTL  'copy_lists';
;  261          0457  1
;  262          0458  1 ROUTINE COPY_LISTS ( COPY__ROUTINE ) =
;  263          0459  2 BEGIN
;  264          0460  2 !+++
;  265          0461  2 !
;  266          0462  2 !  FUNCTIONAL DESCRIPTION:
;  267          0463  2 !       Set up the arguement list for the call to change mode dispatcher.
;  268          0464  2 !
;  269          0465  2 !  INPUT:
;  270          0466  2 !       copy__routine = Name of routine which will execute in kernel mode.
;  271          0467  2 !
;  272          0468  2 !---
;  273          0469  2 BIND ROUTINE
;  274          0470  2     COPY_ROUTINE = .COPY__ROUTINE : call_s_4;
;  275          0471  2
;  276          0472  2 LOCAL
;  277          0473  2     CMK_ARGLST : VECTOR [2,LONG],
;  278          0474  2     CMK_STATUS;                                         ! status from $CMKRNL call
;  279          0475  2
;  280          0476  2 !
;  281          0477  2 !   copy the GSD data base to user buffer so that it need not be processed in kernal mode
;  282          0478  2 !
;  283          0479  2 CMK_ARGLST [0] = 1;
;  284          0480  2 INS$L_BUFSIZ = INS$C_BUFSIZ;                                 ! BUFFER SIZE FOR FIRST COPY ATTEMPT
;  285          0481  2 REPORT ( $EXPREG ( PAGCNT = INS$C_BUFSIZ, RETADR = INS$A_BUFFER));
;  286          0482  2 CMK_ARGLST [1] = .INS$A_BUFFER;
;  287          0483  2
;  288          0484  3 WHILE (CMK_STATUS =
;  289          0485  4         $CMKRNL (ROUTIN = COPY_ROUTINE, ARGLST = CMK_ARGLST)
;  290          0486  2       ) EQL SS$_ACCVIO DO                                   ! If buffer too small
;  291          0487  3     BEGIN                                                   ! double the size of the buffer and try agai
;  292          0488  3     REPORT ($DELTVA ( INADR = INS$A_BUFFER ) ); ! return the buffer
;  293          0489  3     INS$L_BUFSIZ = 2 * .INS$L_BUFSIZ;                       ! double the allocation size
;  294          0490  3     REPORT ($EXPREG ( PAGCNT = .INS$L_BUFSIZ, RETADR = INS$A_BUFFER));
;  295          0491  3     CMK_ARGLST [1] = .INS$A_BUFFER;
;  296          0492  2     END;
;  297          0493  2
;  298          0494  2 IF NOT .CMK_STATUS THEN RETURN .CMK_STATUS;
;  299          0495  2 RETURN TRUE;
;  300          0496  1 END;    ! routine  copy_lists


                                                .EXTRN  SYS$EXPREG, SYS$CMKRNL
                                                .EXTRN  SYS$DELTVA

                                003C 00000 COPY_LISTS:
                                                .WORD   Save R2,R3,R4,R5                ; 0458
                     55 00000000G   00   9E 00002       MOVAB   LIB$SIGNAL, R5          ;
                     54 00000000G   00   9E 00009       MOVAB   SYS$EXPREG, R4          ;
                     53      0000'  CF   9E 00010       MOVAB   INS$A_BUFFER, R3        ;
                     5E             04   C2 00015       SUBL2   #4, SP                  ;
                                    01   DD 00018       PUSHL   #1                      ; 0479
                08   A3             08   D0 0001A       MOVL    #8, INS$L_BUFSIZ        ; 0480
                                    7E   7C 0001E       CLRQ    -(SP)                   ; 0481
                                    53   DD 00020       PUSHL   R3                      ;
```

```
                                    08   DD 00022          PUSHL   #8                                                      ;
                                64  04   FB 00024          CALLS   #4, SYS$EXPREG                                          ;
                                05  50   E8 00027          BLBS    STATUS, 2$                                              ;
                                    50   DD 0002A 1$:      PUSHL   STATUS                                                  ;
                                65  01   FB 0002C          CALLS   #1, LIB$SIGNAL                                          ;
                            04  AE  63   D0 0002F 2$:      MOVL    INS$A_BUFFER, CMK_ARGLST+4                      ;0482
                                5E   DD 00033              PUSHL   SP                                              ;0485
                                04  AC   DD 00035          PUSHL   COPY_ROUTINE                                            ;
                00000000G       00   02   FB 00038          CALLS   #2, SYS$CMKRNL                                          ;
                                52   50   D0 0003F          MOVL    R0, CMK_STATUS                                          ;
                                0C   52   D1 00042          CMPL    CMK_STATUS, #12                                 ;0486
                                     26   12 00045          BNEQ    4$                                                      ;
                                     7E   7C 00047          CLRQ    -(SP)                                           ;0488
                                     53   DD 00049          PUSHL   R3                                                      ;
                00000000G       00   03   FB 0004B          CALLS   #3, SYS$DELTVA                                          ;
                                05   50   E8 00052          BLBS    STATUS, 3$                                              ;
                                     50   DD 00055          PUSHL   STATUS                                                  ;
                                65   01   FB 00057          CALLS   #1, LIB$SIGNAL                                          ;
                            08  A3   02   C4 0005A 3$:      MULL2   #2, INS$L_BUFSIZ                                ;0489
                                     7E   7C 0005E          CLRQ    -(SP)                                           ;0490
                                     53   DD 00060          PUSHL   R3                                                      ;
                                08  A3   DD 00062          PUSHL   INS$L_BUFSIZ                                            ;
                                64   04   FB 00065          CALLS   #4, SYS$EXPREG                                          ;
                                BF   50   E9 00068          BLBC    STATUS, 1$                                              ;
                                     C2   11 0006B          BRB     2$                                              ;0491
                                04   52   E8 0006D 4$:      BLBS    CMK_STATUS, 5$                                  ;0494
                                50   52   D0 00070          MOVL    CMK_STATUS, R0                                          ;
                                          04 00073          RET                                                             ;
                                50        01   D0 00074 5$:      MOVL    #1, R0                                  ;0495
                                          04 00077          RET                                                     ;0496

; Routine Size:  120 bytes,    Routine Base:  $CODE$ + 0096

;  301          0497  1
```

```
;  303          0498  1 %SBTTL  'copy_gsd_lists';
;  304          0499  1
;  305          0500  1 ROUTINE COPY_GSD_LISTS (BUF, PCB_ADR) : CALL_S_4 =
;  306          0501  1 !+++
;  307          0502  1 !
;  308          0503  1 !   FUNCTIONAL DESCRIPTION:
;  309          0504  1 !        Execute in Kernal mode to copy GSD data base to user buffer.
;  310          0505  1 !
;  311          0506  1 !   INPUT:
;  312          0507  1 !        buf     = address of user buffer to copy GSD data to
;  313          0508  1 !        pcb_adr = address of process control block, courtesy of the change mode
;  314          0509  1 !                  dispatcher
;  315          0510  1 !
;  316          0511  1 !   ROUTINE VALUE:
;  317          0512  1 !        ss$_accvio :    user buffer was too small
;  318          0513  1 !        success :       lists were copied to buffer
;  319          0514  1 !--
;  320          0515  2 BEGIN
;  321          0516  2
;  322          0517  2 LABEL
;  323          0518  2     LOCKED;             ! Exit block with error status so mutex can be unlocked
;  324          0519  2
;  325          0520  2 !
;  326          0521  2 !   Table of pointers to the GSD lists
;  327          0522  2 !
;  328          0523  2 OWN
;  329          0524  2     GSD_LISTS : VECTOR [INS$C_NUMGSDLSTS] INITIAL (
;  330          0525  2                 EXE$GL_GSDSYSFL,
;  331          0526  2                 EXE$GL_GSDGRPFL,
;  332          0527  2                 EXE$GL_GSDDELFL,
;  333          0528  2                 0);
;  334          0529  2
;  335          0530  2 LOCAL
;  336          0531  2     GBLSEC_TAB : REF VECTOR [,LONG],            ! global section table
;  337          0532  2     PROBE_LEN : WORD,                           ! length for PROBEW
;  338          0533  2     STATUS;
;  339          0534  2
;  340          0535  2 MAP
;  341          0536  2     MMG$GL_SYSPHD : REF BBLOCK;                 ! System process header
;  342          0537  2
;  343          0538  2 SCH$LOCKR (EXE$GL_GSDMTX, .PCB_ADR);            ! Lock the mutex
;  344          0539  2
;  345          0540  2 LOCKED:
;  346          0541  3 BEGIN                           ! block LOCKED: permits unlocking of mutex on error exit
;  347          0542  3 STATUS = SS$_NORMAL;            ! preset success status
;  348          0543  3
;  349          0544  3 GBLSEC_TAB = .MMG$GL_SYSPHD  + .MMG$GL_SYSPHD [PHD$L_PSTBASOFF];  ! Base address of global section table
;  350          0545  3
;  351          0546  3 INCR I FROM 0 TO INS$C_NUMGSDLSTS-2 BY 1 DO     ! For each global section list
;  352          0547  4     BEGIN
;  353          0548  4     LOCAL
;  354          0549  4         GSD_ENTRY : REF BBLOCK,                 ! Global section descriptor entry
;  355          0550  4         GSD_LSTHEAD;                            ! GSD listhead
;  356          0551  4
;  357          0552  4     GSD_LSTHEAD = .GSD_LISTS [.I];
;  358          0553  4
;  359          0554  4     IF PROBEW ( MODEADDR, GSD$W_SIZE2, .BUF)    ! room in the buffer for list marker?
```

```
;  360          0555  4     THEN
;  361          0556  4         BUF = CH$FILL( -1, .GSD$W_SIZE2, .BUF)  ! Mark beginning of new list
;  362          0557  4     ELSE
;  363          0558  4         LEAVE LOCKED WITH (STATUS = SS$_ACCVIO);
;  364          0559  4
;  365          0560  4     GSD_ENTRY = ..GSD_LSTHEAD;                  ! first entry in list
;  366          0561  4     WHILE .GSD_ENTRY NEQ .GSD_LSTHEAD DO        ! follow list until back to listhead
;  367          0562  5         BEGIN
;  368          0563  5         IF .GSD_ENTRY [GSD$B_TYPE] eql DYN$C_EXTGSD     ! Is this an extended GSD?
;  369          0564  5         THEN
;  370          0565  6             BEGIN
;  371          0566  6             !
;  372          0567  6             !   Extended GSD
;  373          0568  6             !
;  374          0569  6             PROBE_LEN = .GSD_ENTRY [GSD$W_SIZE] + 4;            ! size of GSD and longword for section table
;  375          0570  6             IF PROBEW ( MODEADDR, PROBE_LEN, .BUF)
;  376          0571  6             THEN
;  377          0572  6                 BUF = CH$COPY (.GSD_ENTRY [GSD$W_SIZE], .GSD_ENTRY, 0, ! copy the entry and fill last longwo
;  378          0573  6                                .PROBE_LEN, .BUF)
;  379          0574  6             ELSE
;  380          0575  6                 LEAVE LOCKED WITH (STATUS = SS$_ACCVIO);
;  381          0576  6             END                                 ! extended GSD
;  382          0577  5         ELSE
;  383          0578  5             !
;  384          0579  5             !   Not an extended GSD
;  385          0580  5             !
;  386          0581  6             BEGIN
;  387          0582  6             LOCAL
;  388          0583  6                 GST_ENTRY_ADR,
;  389          0584  6                 GST_INDEX;
;  390          0585  6
;  391          0586  6             PROBE_LEN = .GSD_ENTRY [GSD$W_SIZE] + 4 + SEC$C_LENGTH;     ! Check if there is room for GSD, gl
;  392          0587  6             IF NOT PROBEW ( MODEADDR, PROBE_LEN, .BUF)          !   address, and global section table entry
;  393          0588  6             THEN LEAVE LOCKED WITH (STATUS = SS$_ACCVIO);
;  394          0589  6             BUF = CH$MOVE (.GSD_ENTRY [GSD$W_SIZE], .GSD_ENTRY, .BUF);
;  395          0590  6
;  396          0591  6             !
;  397          0592  6             !   Get Global Section Table Index
;  398          0593  6             !   WORD must be sign extended
;  399          0594  6             !
;  400          0595  6             GST_INDEX = .(GSD_ENTRY [GSD$W_GSTX])<0,16,1>;
;  401          0596  6
;  402          0597  6             GST_ENTRY_ADR = GBLSEC_TAB [.GST_INDEX];            ! address of global section table entry
;  403          0598  6             .BUF = .GST_ENTRY_ADR;                              ! buffer address of GST entry
;  404          0599  6             BUF = .BUF + 4;                                     ! skip over storage of GST entry address
;  405          0600  6
;  406          0601  6             BUF = CH$MOVE (SEC$C_LENGTH, .GST_ENTRY_ADR, .BUF); ! copy the GST entry
;  407          0602  5             END;            ! non extended GSD
;  408          0603  5         GSD_ENTRY = ..GSD_ENTRY;
;  409          0604  4         END;                ! WHILE traversing list
;  410          0605  3     END;                    ! INCRementing through list heads
;  411          0606  3
;  412          0607  3     !
;  413          0608  3     !   Check if there is room in the buffer to pad the end with zeros
;  414          0609  3     !   to mark end of lists.
;  415          0610  3     !
;  416          0611  3     IF PROBEW ( MODEADDR, GSD$W_SIZE2, .BUF)
```

```
;  417          0612  3     THEN
;  418          0613  3         BUF = CH$FILL( 0, .GSD$W_SIZE2, .BUF)
;  419          0614  3     ELSE
;  420          0615  3         LEAVE LOCKED WITH (STATUS = SS$_ACCVIO);
;  421          0616  3
;  422          0617  2 END;    ! block LOCKED:
;  423          0618  2
;  424          0619  2 SCH$UNLOCK (EXE$GL_GSDMTX, .PCB_ADR);   ! Unlock the mutex
;  425          0620  2
;  426          0621  2 RETURN .STATUS;
;  427          0622  1 END;    ! routine copy_gsd_lists


                                                        .PSECT  $OWN$,NOEXE,2

                                                00013   .BLKB   1
                    00000000G 00000000G 00000000G 00014 GSD_LISTS:
                                                        .ADDRESS EXE$GL_GSDSYSFL, EXE$GL_GSDGRPFL, -        ;
                                                                EXE$GL_GSDDELFL
                                00000000  00020         .LONG   0


                                                        .PSECT  $CODE$,NOWRT,2

                                    0FFC 00000 COPY_GSD_LISTS:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 0500
                          5E            04  C2 00002          SUBL2   #4, SP                              ;
                          5B            54  D0 00005          MOVL    R4, R11                             ;
                          50 00000000G  00  9E 00008          MOVAB   EXE$GL_GSDMTX, R0                   ; 0538
                          54            5B  D0 0000F          MOVL    PCB_ADR, R4                         ;
                             00000000G  00  16 00012          JSB     SCH$LOCKR                           ;
                          58            01  D0 00018          MOVL    #1, STATUS                          ; 0542
                          50 00000000G  00  D0 0001B          MOVL    MMG$GL_SYSPHD, R0                   ; 0544
                          6E       20 B040  9E 00022          MOVAB   @32(R0)[R0], GBLSEC_TAB             ;
                                        57  D4 00027          CLRL    I                                   ; 0546
                          5A      0000'CF47  D0 00029 1$:     MOVL    GSD_LISTS[I], GSD_LSTHEAD           ; 0552
                    04    BC      0000'  CF  0D 0002F         PROBEW  MODEADDR, GSD$W_SIZE2, @BUF         ; 0554
                                        78  13 00038          BEQL    6$                                  ;
0000'  CF     FF    8F            6E        00  2C 0003A          MOVC5   #0, (SP), #-1, GSD$W_SIZE2, @BUF    ; 0556
                                      04    BC      00042       ;
                          04    AC            53  D0 00044          MOVL    R3, BUF                         ;
                          56                  6A  D0 00048          MOVL    (GSD_LSTHEAD), GSD_ENTRY        ; 0560
                          5A                  56  D1 0004B 2$:      CMPL    GSD_ENTRY, GSD_LSTHEAD          ; 0561
                                              55  13 0004E          BEQL    5$                              ;
                                        28    0A  A6  91 00050          CMPB    10(GSD_ENTRY), #40          ; 0563
                                              18  12 00054          BNEQ    3$                              ;
                          59    08      A6        04  A1 00056          ADDW3   #4, 8(GSD_ENTRY), PROBE_LEN ; 0569
                    04    BC            59  0000' CF  0D 0005B          PROBEW  MODEADDR, PROBE_LEN, @BUF   ; 0570
                                              5F  13 00062          BEQL    7$                              ;
          59              00            66    08  A6  2C 00064          MOVC5   8(GSD_ENTRY), (GSD_ENTRY), #0, PROBE_LEN, -     ; 0573
                                      04    BC      0006A       ;           @BUF                            ;
                                              2E  11 0006C          BRB     4$                              ;
                          59    08      A6        24  A1 0006E 3$:      ADDW3   #36, 8(GSD_ENTRY), PROBE_LEN ; 0586
                    04    BC            59  0000' CF  0D 00073          PROBEW  MODEADDR, PROBE_LEN, @BUF   ; 0587
                                              47  13 0007A          BEQL    7$                              ;
```

```
                04   BC                66        08   A6  28 0007C          MOVC3   8(GSD_ENTRY), (GSD_ENTRY), @BUF          ; 0589
                          04   AC             53   D0  00082               MOVL    R3, BUF                                   ;
                               50        16   A6  32 00086                CVTWL   22(GSD_ENTRY), GST_INDEX                  ; 0595
                               50        00 BE40  DE 0008A                MOVAL   @GBLSEC_TAB[GST_INDEX], GST_ENTRY_ADR     ; 0597
                          04   BC             50   D0  0008F               MOVL    GST_ENTRY_ADR, @BUF                       ; 0598
                          04   AC             04   C0  00093               ADDL2   #4, BUF                                   ; 0599
                04   BC                60        20   28  00097            MOVC3   #32, (GST_ENTRY_ADR), @BUF                ; 0601
                          04   AC             53   D0  0009C 4$:           MOVL    R3, BUF                                   ;
                               56             66   D0  000A0               MOVL    (GSD_ENTRY), GSD_ENTRY                    ; 0603
                                              A6   11  000A3               BRB     2$                                        ; 0561
                     80                       57        02   F3  000A5 5$:  AOBLEQ  #2, I, 1$                               ; 0546
                04   BC      0000'  CF  0000'  CF        0D  000A9         PROBEW  MODE:ADDR, GSD$W_SIZE2, @BUF              ; 0611
                                              0F   13  000B2 6$:           BEQL    7$                                        ;
    0000'  CF        00                       6E        00   2C  000B4      MOVC5   #0, (SP), #0, GSD$W_SIZE2, @BUF           ; 0613
                                                   04   BC  000BB           ;                                                 ;
                          04   AC             53   D0  000BD               MOVL    R3, BUF                                   ;
                                              03   11  000C1               BRB     8$                                        ;
                                    58        0C   D0  000C3 7$:           MOVL    #12, STATUS                               ; 0615
                                    50 00000000G  00   9E  000C6 8$:       MOVAB   EXE$GL_GSDMTX, R0                         ; 0619
                                    54            5B   D0  000CD           MOVL    PCB_ADR, R4                               ;
                                       00000000G  00   16  000D0           JSB     SCH$UNLOCK                                ;
                                    50            58   D0  000D6           MOVL    STATUS, R0                                ; 0621
                                                  04   D0  000D9           RET                                               ; 0622

; Routine Size:  218 bytes,    Routine Base:  $CODE$ + 010E

;  428          0623  1
```

```
;  430          0624  1 %SBTTL 'list_gsds';
;  431          0625  1
;  432          0626  1 ROUTINE LIST_GSDS (GSD_BUFFER) =
;  433          0627  2 BEGIN
;  434          0628  2 !+++
;  435          0629  2 !
;  436          0630  2 !  FUNCTIONAL DESCRIPTION:
;  437          0631  2 !       format the Global section information in the user buffer and
;  438          0632  2 !       print it to sys$output.
;  439          0633  2 !
;  440          0634  2 !  INPUT:
;  441          0635  2 !       gsd_buffer =    address of user buffer which contains GSD info.
;  442          0636  2 !
;  443          0637  2 !  OUTPUT:
;  444          0638  2 !       Format and put to sys$output the system, group and delete pending
;  445          0639  2 !       global section lists.
;  446          0640  2 !
;  447          0641  2 !---
;  448          0642  2
;  449          0643  2 OWN
;  450          0644  2     !
;  451          0645  2     !   Table of control strings to print the list name
;  452          0646  2     !
;  453          0647  2     TITLE_DESCS :       VECTOR [3] INITIAL (
;  454          0648  2                 FAO_SYSTITLE,
;  455          0649  2                 FAO_GRPTITLE,
;  456          0650  2                 FAO_DELTITLE);
;  457          0651  2
;  458          0652  2 LOCAL
;  459          0653  2     BUFFER : REF BBLOCK,
;  460          0654  2     GPT_ENTRIES,        ! Number of GPT entries used in a list
;  461          0655  2     GPT_UNUSED,         ! Number of GPT entries unused in a list
;  462          0656  2     MATCHES,            ! boolean
;  463          0657  2     PRINT_TITLE,        ! boolean to record if title should be printed
;  464          0658  2     SECTIONS,           ! Count of global sections
;  465          0659  2     STATUS,
;  466          0660  2     TITLE;              ! Index of which title to use
;  467          0661  2
;  468          0662  2
;  469          0663  2 BUFFER = .GSD_BUFFER;
;  470          0664  2
;  471          0665  2 !
;  472          0666  2 !    Initialize counters
;  473          0667  2 !
;  474          0668  2 SECTIONS = 0;
;  475          0669  2 GPT_ENTRIES = 0;
;  476          0670  2 TITLE = -1;
;  477          0671  2
;  478          0672  2 !
;  479          0673  2 !   Zeros in the size field signal the end of the buffered data.
;  480          0674  2 !   Minus ones signal the start of a new list.
;  481          0675  2 !
;  482          0676  2 UNTIL .BUFFER [GSD$W_SIZE] EQL 0 DO
;  483          0677  3     BEGIN
;  484          0678  3
;  485          0679  3     BUFFER = .BUFFER + $BYTEOFFSET (GSD$W_SIZE) + 2;    ! Skip over minus ones
;  486          0680  3     !
```

```
;  487     0681  3     !   Setup to output appropriate title
;  488     0682  3     !
;  489     0683  3     PRINT_TITLE = TRUE;
;  490     0684  3     TITLE = .TITLE + 1;                                 ! Increment title to index next title descriptor
;  491     0685  3
;  492     0686  3     MATCHES = FALSE;
;  493     0687  3
;  494     0688  3     WHILE .(BUFFER [GSD$W_SIZE])<0,16,1> GTR 0 DO       ! For each entry in this list
;  495     0689  4         BEGIN
;  496     0690  4         LOCAL
;  497     0691  4             GSTE_ADR : REF BBLOCK,                      ! address in buffer of Global section table entry
;  498     0692  4             PAGES;
;  499     0693  4         MAP
;  500     0694  4             KFE : REF $BBLOCK;
;  501     0695  4
;  502     0696  4         IF .KFE EQL 0                                   ! If printing all                                    6
;  503     0697  4         THEN MATCHES = TRUE                                                                                  7
;  504     0698  4         ELSE
;  505     0699  5             BEGIN
;  506     0700  5             LOCAL
;  507     0701  5                 CME_ARGLIST : VECTOR [3,LONG];
;  508     0702  5
;  509     0703  5             CME_ARGLIST [0] = 2;                                                                             6
;  510     0704  5             CME_ARGLIST [1] = .KFE;                     ! Pass KFE address
;  511     0705  5             CME_ARGLIST [2] = .BUFFER;                  ! Pass GSD address
;  512     0706  5
;  513     0707  5             MATCHES = $CMEXEC (ROUTIN=EXEC_MODE_NAME_CHK, ARGLST=CME_ARGLIST);
;  514     0708  4             END;                                                                                             6
;  515     0709  4
;  516     0710  4         IF .MATCHES                                     ! If this is one to print
;  517     0711  4         THEN
;  518     0712  5             BEGIN                                                                                            2
;  519     0713  5             IF .PRINT_TITLE                             ! Title hasn't already been printed
;  520     0714  5             THEN
;  521     0715  6                 BEGIN
;  522     0716  6                 BIND
;  523     0717  6                     TITLE_DESC = TITLE_DESCS [.TITLE];                                                       6
;  524     0718  6                                                                                                              2
;  525     0719  6                 INS$FAO_AND_OUT ( .TITLE_DESC); ! Load title into output buffer
;  526     0720  6                 PRINT_TITLE = FALSE;                                                                         7
;  527     0721  5                 END;
;  528     0722  5
;  529     0723  5             SECTIONS = .SECTIONS + 1;                   ! Count the global sections
;  530     0724  5
;  531     0725  5             DECODE_FLAGS (.BUFFER);                     ! Format the ASCII symbols for the flags             2
;  532     0726  5                                                                                                              6
;  533     0727  5             INS$FAOL (                                  ! print either SYS or GRP
;  534     0728  6                 ( IF (.BUFFER [GSD$W_FLAGS] AND SEC$M_SYSGBL) NEQ 0                                          2
;  535     0729  6                       THEN FAO_SYSTEM
;  536     0730  6                       ELSE FAO_GROUP )
;  537     0731  5                 , .BUFFER [GSD$W_PCBGRP]);
;  538     0732  5
;  539     0733  4             END;
;  540     0734  4         IF .BUFFER [GSD$B_TYPE] EQL DYN$C_EXTGSD        ! Extended GSD?
;  541     0735  4         THEN
;  542     0736  5             BEGIN       ! Extended GSD
;  543     0737  5             PAGES = .BUFFER [GSD$L_PAGES];              ! Pages in section
```

```
;  544          0738  5            GPT_ENTRIES = .GPT_ENTRIES + ((.PAGES + 3) AND NOT 1);      ! Count 2 zero GPT entries and round
;  545          0739  5                                                                        !  even number of longwords.
;  546          0740  5            IF .MATCHES                                  ! If this is one to print
;  547          0741  5            THEN
;  548          0742  5                INS$FAO_AND_OUT( FAO_PAGPFN, .PAGES, .BUFFER [GSD$L_BASEPFN]);
;  549          0743  5
;  550          0744  5            BUFFER = .BUFFER [GSD$W_SIZE] + .BUFFER + 4;        ! skip 4 bytes for the empty Section Table e
;  551          0745  5            END
;  552          0746  4        ELSE
;  553          0747  5            BEGIN       ! non-extended
;  554          0748  5            GSTE_ADR = .BUFFER [GSD$W_SIZE] + .BUFFER + 4;
;  555          0749  5            PAGES = .GSTE_ADR [SEC$L_PAGCNT];
;  556          0750  5            GPT_ENTRIES = .GPT_ENTRIES + ((.PAGES + 3) AND NOT 1);      ! Count 2 zero GPT entries and round
;  557          0751  5                                                                        !  even number of longwords.
;  558          0752  5            IF .MATCHES                                  ! If this is one to print
;  559          0753  5            THEN
;  560          0754  5                INS$FAO_AND_OUT( FAO_PAGCNT, .PAGES, .GSTE_ADR [SEC$L_REFCNT]);
;  561          0755  5            BUFFER = .BUFFER [GSD$W_SIZE] + .BUFFER + 4 + SEC$C_LENGTH;
;  562          0756  4            END;
;  563          0757  4
;  564          0758  4        MATCHES = FALSE;
;  565          0759  3        END;            ! While entries in this list
;  566          0760  3
;  567          0761  2    END;
;  568          0762  2
;  569          0763  2 !
;  570          0764  2 !   Print the summary lines
;  571          0765  2 !
;  572          0766  2 IF .KFE EQL 0
;  573          0767  2 THEN
;  574          0768  3    BEGIN
;  575          0769  3    GPT_UNUSED = .SGN$GL_MAXGPGCT - .GPT_ENTRIES;                    ! Number of unused = max available -
;  576          0770  3    INS$FAO_AND_OUT( FAO_SUMMARY, .SECTIONS, .GPT_ENTRIES, .GPT_UNUSED);
;  577          0771  3    END
;  578          0772  2 ELSE
;  579          0773  2    INS$OUTPUT_FAOBUF ();
;  580          0774  2
;  581          0775  2 RETURN TRUE;
;  582          0776  1 END;            ! routine  list_gsds


                                                    .PSECT  $OWN$,NOEXE,2

                00000000' 00000000' 00000000' 00024 TITLE_DESCS:
                                                    .ADDRESS FAO_SYSTITLE, FAO_GRPTITLE, FAO_DELTITLE   ;

                                                    .EXTRN  SYS$CMEXEC

                                                    .PSECT  $CODE$,NOWRT,2

                                    07FC 00000 LIST_GSDS:
                                                    .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10            ; 0626
                          5A      0000V  CF  9E 00002       MOVAB   INS$FAO_AND_OUT, R10                ;
                          5E             0C  C2 00007       SUBL2   #12, SP                             ;
                          54       04    AC  D0 0000A       MOVL    GSD_BUFFER, BUFFER                  ; 0663
                                         58  D4 0000E       CLRL    SECTIONS                            ; 0668
```

```
                                56  D4 00010          CLRL    GPT_ENTRIES                             ; 0669
                           55   01  CE 00012          MNEGL   #1, TITLE                               ; 0670
                      08   A4   B5 00015 1$:         TSTW    8(BUFFER)                               ; 0676
                                03  12 00018          BNEQ    2$                                      ;
                             00C7  31 0001A          BRW     14$                                     ;
                           54   0A  C0 0001D 2$:      ADDL2   #10, BUFFER                             ; 0679
                           59   01  D0 00020          MOVL    #1, PRINT_TITLE                         ; 0683
                                55  D6 00023          INCL    TITLE                                   ; 0684
                                57  D4 00025 3$:      CLRL    MATCHES                                 ; 0686
                      08   A4   B5 00027          TSTW    8(BUFFER)                               ; 0688
                                E9  15 0002A          BLEQ    1$                                      ;
                           50   0000' CF D0 0002C          MOVL    KFE, R0                                 ; 0696
                                05  12 00031          BNEQ    4$                                      ;
                           57   01  D0 00033          MOVL    #1, MATCHES                             ; 0697
                                1B  11 00036          BRB     5$                                      ;
                           6E   02  D0 00038 4$:      MOVL    #2, CME_ARGLIST                         ; 0703
                      04   AE   50  D0 0003B          MOVL    R0, CME_ARGLIST+4                       ; 0704
                      08   AE   54  D0 0003F          MOVL    BUFFER, CME_ARGLIST+8                   ; 0705
                                5E  DD 00043          PUSHL   SP                                      ; 0707
                           0000V CF  9F 00045          PUSHAB  EXEC_MODE_NAME_CHK                      ;
                 00000000G 00   02  FB 00049          CALLS   #2, SYS$CMEXEC                          ;
                           57   50  D0 00050          MOVL    R0, MATCHES                             ;
                           32   57  E9 00053 5$:      BLBC    MATCHES, 9$                             ; 0710
                           0A   59  E9 00056          BLBC    PRINT_TITLE, 6$                         ; 0713
                           0000'CF45 DD 00059          PUSHL   TITLE_DESCS[TITLE]                      ; 0719
                           6A   01  FB 0005E          CALLS   #1, INS$FAO_AND_OUT                     ;
                                59  D4 00061          CLRL    PRINT_TITLE                             ; 0720
                                58  D6 00063 6$:      INCL    SECTIONS                                ; 0723
                                54  DD 00065          PUSHL   BUFFER                                  ; 0725
                      0000V CF   01  FB 00067          CALLS   #1, DECODE_FLAGS                        ;
                           7E   0E  A4 3C 0006C          MOVZWL  14(BUFFER), -(SP)                       ; 0731
                                20  A4 B5 00070          TSTW    32(BUFFER)                              ; 0728
                                07  18 00073          BGEQ    7$                                      ;
                           50   0000' CF 9E 00075          MOVAB   FAO_SYSTEM, R0                          ;
                                05  11 0007A          BRB     8$                                      ;
                           50   0000' CF 9E 0007C 7$:      MOVAB   FAO_GROUP, R0                           ;
                                50  DD 00081 8$:      PUSHL   R0                                      ;
                      0000V CF   02  FB 00083          CALLS   #2, INS$FAOL                            ;
                           28   0A  A4 91 00088 9$:      CMPB    10(BUFFER), #40                         ; 0734
                                28  12 0008C          BNEQ    11$                                     ;
                           52   28  A4 D0 0008E          MOVL    40(BUFFER), PAGES                       ; 0737
                           51   03  A2 9E 00092          MOVAB   3(R2), R1                               ; 0738
                           51      8A 00096          BICB2   #1, R1                                  ;
                           56      C0 00099          ADDL2   R1, GPT_ENTRIES                         ;
                           0C      E9 0009C          BLBC    MATCHES, 10$                            ; 0740
                                24 A4 DD 0009F          PUSHL   36(BUFFER)                              ; 0742
                                52  DD 000A2          PUSHL   PAGES                                   ;
                           0000' CF 9F 000A4          PUSHAB  FAO_PAGPFN                              ;
                           6A      03  FB 000A8          CALLS   #3, INS$FAO_AND_OUT                     ;
                           50   08  A4 3C 000AB 10$:     MOVZWL  8(BUFFER), R0                           ; 0744
                           54   04 A440 9E 000AF          MOVAB   4(BUFFER)[R0], BUFFER                   ;
                                2B  11 000B4          BRB     13$                                     ; 0734
                           53   08  A4 3C 000B6 11$:     MOVZWL  8(BUFFER), R3                           ; 0748
                           50   04 A443 9E 000BA          MOVAB   4(BUFFER)[R3], GSTE_ADR                 ;
                           52   1C  A0 D0 000BF          MOVL    28(GSTE_ADR), PAGES                     ; 0749
                           51   03  A2 9E 000C3          MOVAB   3(R2), R1                               ; 0750
                           51      01  8A 000C7          BICB2   #1, R1                                  ;
```

```
                                56              51  C0 000CA          ADDL2   R1, GPT_ENTRIES                                                 ;
                                0C              57  E9 000CD          BLBC    MATCHES, 12$                                                    ; 0752
                                          18    A0  DD 000D0          PUSHL   24(GSTE_ADR)                                                    ; 0754
                                                52  DD 000D3          PUSHL   PAGES                                                           ;
                                          0000' CF  9F 000D5          PUSHAB  FAO_PAGCNT                                                      ;
                                6A              03  FB 000D9          CALLS   #3, INS$FAO_AND_OUT                                             ;
                                54        24 A443   9E 000DC 12$:      MOVAB   36(BUFFER)[R3], BUFFER                                          ; 0755
                                             FF41   31 000E1 13$:      BRW     3$                                                              ; 0758
                                          0000' CF  D5 000E4 14$:      TSTL    KFE                                                             ; 0766
                                                17  12 000E8          BNEQ    15$                                                             ;
                          50 00000000G  00      56  C3 000EA          SUBL3   GPT_ENTRIES, SGN$GL_MAXGPGCT, GPT_UNUSED                        ; 0769
                                                50  DD 000F2          PUSHL   GPT_UNUSED                                                      ; 0770
                                                56  DD 000F4          PUSHL   GPT_ENTRIES                                                     ;
                                                58  DD 000F6          PUSHL   SECTIONS                                                        ;
                                          0000' CF  9F 000F8          PUSHAB  FAO_SUMMARY                                                     ;
                                6A              04  FB 000FC          CALLS   #4, INS$FAO_AND_OUT                                             ;
                                                05  11 000FF          BRB     16$                                                             ; 0766
                             0000V CF           00  FB 00101 15$:      CALLS   #0, INS$OUTPUT_FAOBUF                                           ; 0773
                                50              01  D0 00106 16$:      MOVL    #1, R0                                                          ; 0775
                                                    04 00109          RET                                                                     ; 0776

; Routine Size:  266 bytes,    Routine Base:  $CODE$ + 01E8

;  583          0777  1
```

```
;  585          0778  1 %SBTTL  'exec_mode_name_chk';
;  586          0779  1
;  587          0780  1 ROUTINE EXEC_MODE_NAME_CHK (KFE, BUFFER) =
;  588          0781  2 BEGIN
;  589          0782  2 !+++
;  590          0783  2 !
;  591          0784  2 !   FUNCTIONAL DESCRIPTION:
;  592          0785  2 !        Execute in Executive mode to compare the name string in the
;  593          0786  2 !        supplied KFE to the global section name string in the
;  594          0787  2 !        supplied GSD.
;  595          0788  2 !
;  596          0789  2 !   INPUT:
;  597          0790  2 !        KFE = Address of KFE
;  598          0791  2 !        BUFFER = Address of GSD
;  599          0792  2 !
;  600          0793  2 !   OUTPUT:
;  601          0794  2 !        Routine value = TRUE if the KFE file name matches the global section name - the suffix
;  602          0795  2 !                      = FALSE if otherwise
;  603          0796  2 !
;  604          0797  2 !---
;  605          0798  2 MAP
;  606          0799  2     KFE : REF $BBLOCK,
;  607          0800  2     BUFFER : REF BBLOCK;
;  608          0801  2
;  609          0802  2 LOCAL
;  610          0803  2     MATCHES;
;  611          0804  2
;  612          0805  2 MATCHES = FALSE;                                ! Assume that the names are different
;  613          0806  2
;  614          0807  2 IF .BUFFER [GSD$B_TYPE] NEQ DYN$C_EXTGSD
;  615          0808  2 THEN
;  616          0809  2     !
;  617          0810  2     ! Compare the KFE's name string with the global section name less the installed suffix
;  618          0811  2     !
;  619          0812  3     (IF CH$COMPARE ( .KFE [KFE$B_FILNAMLEN], KFE [KFE$T_FILNAM],
;  620          0813  3       .(BUFFER [GSD$T_GSDNAM]) <0,8,0> - 4, BUFFER [GSD$T_GSDNAM] + 1) EQL 0
;  621          0814  3     THEN
;  622          0815  2         MATCHES = TRUE);
;  623          0816  2
;  624          0817  2 RETURN .MATCHES;
;  625          0818  1 END;                 ! routine exec_mode_name_chk


                                003C 00000 EXEC_MODE_NAME_CHK:
                                           .WORD   Save R2,R3,R4,R5                                ; 0780
                                55  D4 00002  CLRL    MATCHES                                     ; 0805
                          51    AC  D0 00004  MOVL    BUFFER, R1                                  ; 0807
                          28 0A A1  91 00008  CMPB    10(R1), #40                                 ;
                                26  13 0000C  BEQL    2$                                          ;
                          50    04  AC  D0 0000E  MOVL    KFE, R0                               ; 0812
                          53    36  A0  9A 00012  MOVZBL  54(R0), R3                             ;
                          52    22  A1  9A 00016  MOVZBL  34(R1), R2                             ; 0813
                          52        04  C2 0001A  SUBL2   #4, R2                                ;
                          54        01  D0 0001D  MOVL    #1, R4                                ; 0812
```

```
                52              00      37    A0              53  2D 00020          CMPC5   R3, 55(R0), #0, R2, 35(R1)          ;
                                                        23      A1     00026                                                      ;
                                                              03  1A 00028          BGTRU   1$                                  ;
                                                        54    01  D9 0002A          SBWC    #1, R4                              ;
                                                              54  D5 0002D 1$:      TSTL    R4                                  ; 0813
                                                              03  12 0002F          BNEQ    2$                                  ;
                                                        55    01  D0 00031          MOVL    #1, MATCHES                         ; 0815
                                                        50    55  D0 00034 2$:      MOVL    MATCHES, R0                         ; 0817
                                                              04  00037          RET                                         ; 0818

; Routine Size:  56 bytes,    Routine Base:  $CODE$ + 02F2
```

```
;  627          0819  1 %SBTTL  'copy_shm_lists';
;  628          0820  1
;  629          0821  1 ROUTINE COPY_SHM_LISTS (BUF, PCB_ADR) : CALL_S_4 =
;  630          0822  1 !+++
;  631          0823  1 !
;  632          0824  1 !  FUNCTIONAL DESCRIPTION:
;  633          0825  1 !       Execute in Kernal mode to copy SHM data base to user buffer.
;  634          0826  1 !
;  635          0827  1 !  INPUT:
;  636          0828  1 !       buf     = address of user buffer to copy GSD data to
;  637          0829  1 !       pcb_adr = address of process control block, courtesy of the change mode
;  638          0830  1 !                 dispatcher
;  639          0831  1 !
;  640          0832  1 !  ROUTINE VALUE:
;  641          0833  1 !       ss$_accvio :    user buffer was too small
;  642          0834  1 !       success :       lists were copied to buffer
;  643          0835  1 !--
;  644          0836  2 BEGIN
;  645          0837  2
;  646          0838  2 GLOBAL REGISTER
;  647          0839  2     GSD_PTR = 6 : REF BBLOCK;
;  648          0840  2
;  649          0841  2 LOCAL
;  650          0842  2     COM_DATA_PAGE : REF BBLOCK,         ! store commom data page address
;  651          0843  2     NAME,
;  652          0844  2     PROBE_LEN : WORD,                   ! length for PROBEW
;  653          0845  2     SHM_CTLBLK : REF BBLOCK,            ! shared memory control block
;  654          0846  2     STATUS;
;  655          0847  2
;  656          0848  2 SHM_CTLBLK = EXE$GL_SHBLIST;
;  657          0849  2
;  658          0850  2 !
;  659          0851  2 !   Loop thru all the shared memories
;  660          0852  2 !
;  661          0853  2 WHILE ((SHM_CTLBLK = ..SHM_CTLBLK) NEQ 0) DO
;  662          0854  3     BEGIN
;  663          0855  3
;  664          0856  3     !
;  665          0857  3     !   Check if enough room in buffer and mark start of new list by
;  666          0858  3     !   filling size field with minus ones.
;  667          0859  3     !
;  668          0860  3     IF PROBEW ( MODEADDR, GSD$W_SIZE2, .BUF)
;  669          0861  3     THEN
;  670          0862  3         BUF = CH$FILL (-1, .GSD$W_SIZE2, .BUF)
;  671          0863  3     ELSE
;  672          0864  3         RETURN SS$_ACCVIO;
;  673          0865  3
;  674          0866  3     !
;  675          0867  3     !    If this shared memory is not connected, then skip it
;  676          0868  3     !
;  677          0869  3     IF (SHB$M_CONNECT AND .SHM_CTLBLK [SHB$B_FLAGS]) NEQ 0
;  678          0870  3     THEN
;  679          0871  4         BEGIN           ! shared memory is connected
;  680          0872  4
;  681          0873  4         !
;  682          0874  4         !    Check if there is room in the buffer for
;  683          0875  4         !       longword for port number
```

```
;  684          0876  4          !       longword for number of global pages
;  685          0877  4          !       longword for maximum number of GSD's in this list
;  686          0878  4          !       16 bytes for the ASCIC name of the shared memory
;  687          0879  4          !    and move them in.
;  688          0880  4          !
;  689          0881  4          PROBE_LEN = 4 + 4 + 4 + 16;
;  690          0882  4          IF NOT PROBEW ( MODEADDR, PROBE_LEN, .BUF)
;  691          0883  4          THEN RETURN SS$_ACCVIO;
;  692          0884  4
;  693          0885  4          .BUF = .SHM_CTLBLK [SHB$B_PORT];
;  694          0886  4          BUF = .BUF + 4;
;  695          0887  4
;  696          0888  4          COM_DATA_PAGE = .SHM_CTLBLK [SHB$L_DATAPAGE];
;  697          0889  4
;  698          0890  4          .BUF = .COM_DATA_PAGE [SHD$L_GSPAGCNT];
;  699          0891  4          BUF = .BUF + 4;
;  700          0892  4
;  701          0893  4          .BUF = .COM_DATA_PAGE [SHD$W_GSDMAX];
;  702          0894  4          BUF = .BUF + 4;
;  703          0895  4
;  704          0896  4          NAME = .COM_DATA_PAGE + $BYTEOFFSET (SHD$T_NAME);
;  705          0897  4          BUF = CH$MOVE (16, .NAME, .BUF);
;  706          0898  4
;  707          0899  4          !
;  708          0900  4          !   Get address of first gsd in this shared memory
;  709          0901  4          !   and check if it is in use.
;  710          0902  4          !
;  711          0903  4          GSD_PTR = .COM_DATA_PAGE [SHD$L_GSDPTR] + .COM_DATA_PAGE;
;  712          0904  4          STATUS = MMG$VALIDATEGSD ( .SHM_CTLBLK, .COM_DATA_PAGE, .NAME);
;  713          0905  4
;  714          0906  4          !
;  715          0907  4          !   For all the used GSDs
;  716          0908  4          !
;  717          0909  4          WHILE (.GSD_PTR NEQ 0) DO
;  718          0910  5              BEGIN
;  719          0911  5              !
;  720          0912  5              !   get info on each gsd
;  721          0913  5              !
;  722          0914  5
;  723          0915  5              PROBE_LEN = .GSD_PTR [GSD$W_SIZE];
;  724          0916  5              IF NOT PROBEW ( MODEADDR, PROBE_LEN, .BUF)
;  725          0917  5              THEN RETURN SS$_ACCVIO;
;  726          0918  5
;  727          0919  5              BUF = CH$MOVE (.PROBE_LEN, .GSD_PTR, .BUF); ! Copy the GSD                ;
;  728          0920  5
;  729          0921  5              !
;  730          0922  5              !   Get the address of the next used GSD in shared memory                 ;
;  731          0923  5              !
;  732          0924  5              STATUS = MMG$GETNXTGSD ( .SHM_CTLBLK, .COM_DATA_PAGE, .NAME);
;  733          0925  4              END;
;  734          0926  4          !
;  735          0927  4          !   Mark end of GSDs for this port
;  736          0928  4          !
;  737          0929  4          IF NOT PROBEW ( MODEADDR, GSD$W_SIZE2, .BUF)
;  738          0930  4          THEN RETURN SS$_ACCVIO;
;  739          0931  4          BUF = CH$FILL (0, .GSD$W_SIZE2, .BUF);
;  740          0932  4
```

```
;  741          0933  3          END;     ! IF connected
;  742          0934  2      END;         ! While there are more Shared Memory Control Blocks
;  743          0935  2
;  744          0936  2 !
;  745          0937  2 !   Fill size field with zeros to mark the end of the buffer
;  746          0938  2 !
;  747          0939  2 IF NOT PROBEW ( MODEADDR, GSD$W_SIZE2, .BUF)
;  748          0940  2 THEN RETURN SS$_ACCVIO;
;  749          0941  2 BUF = CH$FILL (0, .GSD$W_SIZE2, .BUF);
;  750          0942  2
;  751          0943  2 RETURN SS$_NORMAL;
;  752          0944  1 END;     ! routine copy_shm_lists
```

```
                                                    0FFC 00000 COPY_SHM_LISTS:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 0821
                                    57 00000000G    00   9E 00002      MOVAB   EXE$GL_SHBLIST, SHM_CTLBLK          ; 0848
                                    57              67   D0 00009 1$:   MOVL    (SHM_CTLBLK), SHM_CTLBLK            ; 0853
                                                    03   12 0000C      BNEQ    2$                                  ;
                                                 00A6   31 0000E      BRW     7$                                  ;
                        04  BC  0000'  CF  0000'    CF   0D 00011 2$:   PROBEW  MODEADDR, GSD$W_SIZE2, @BUF         ; 0860
                                                    6A   13 0001A      BEQL    4$                                  ;
    0000'  CF       FF  8F          6E              00   2C 0001C      MOVC5   #0, (SP), #-1, GSD$W_SIZE2, @BUF    ; 0862
                                                 04 BC      00024                                                   ;
                            04  AC                  53   D0 00026      MOVL    R3, BUF                             ;
                                DB      0B          A7   E9 0002A      BLBC    11(SHM_CTLBLK), 1$                  ; 0869
                                59                  1C   B0 0002E      MOVW    #28, PROBE_LEN                      ; 0881
                    04  BC              59  0000'   CF   0D 00031      PROBEW  MODEADDR, PROBE_LEN, @BUF           ; 0882
                                                    6B   13 00038      BEQL    6$                                  ;
                            04  BC          15      A7   9A 0003A      MOVZBL  21(SHM_CTLBLK), @BUF                ; 0885
                            04  AC                  04   C0 0003F      ADDL2   #4, BUF                             ; 0886
                                58          04      A7   D0 00043      MOVL    4(SHM_CTLBLK), COM_DATA_PAGE        ; 0888
                            04  BC          10      A8   D0 00047      MOVL    16(COM_DATA_PAGE), @BUF             ; 0890
                            04  AC                  04   C0 0004C      ADDL2   #4, BUF                             ; 0891
                            04  BC          18      A8   3C 00050      MOVZWL  24(COM_DATA_PAGE), @BUF             ; 0893
                            04  AC                  04   C0 00055      ADDL2   #4, BUF                             ; 0894
                                5A          20      A8   9E 00059      MOVAB   32(R8), NAME                        ; 0896
                    04  BC              6A          10   28 0005D      MOVC3   #16, (NAME), @BUF                   ; 0897
                            04  AC                  53   D0 00062      MOVL    R3, BUF                             ;
                        56              58      04  A8   C1 00066      ADDL3   4(COM_DATA_PAGE), COM_DATA_PAGE, GSD_PTR ; 0903
                                        54          57   7D 0006B      MOVQ    SHM_CTLBLK, R4                      ; 0904
                                    00000000G       00   16 0006E      JSB     MMG$VALIDATEGSD                     ;
                                        5B          50   D0 00074 3$:   MOVL    R0, STATUS                          ;
                                                    56   D5 00077      TSTL    GSD_PTR                             ; 0909
                                                    21   13 00079      BEQL    5$                                  ;
                                        59      08  A6   B0 0007B      MOVW    8(GSD_PTR), PROBE_LEN               ; 0915
                    04  BC              59  0000'   CF   0D 0007F      PROBEW  MODEADDR, PROBE_LEN, @BUF           ; 0916
                                                    3A   13 00086 4$:   BEQL    8$                                  ;
                    04  BC              66          59   28 00088      MOVC3   PROBE_LEN, (GSD_PTR), @BUF          ; 0919
                            04  AC                  53   D0 0008D      MOVL    R3, BUF                             ;
                                        54          57   7D 00091      MOVQ    SHM_CTLBLK, R4                      ; 0924
                                    00000000G       00   16 00094      JSB     MMG$GETNXTGSD                       ;
                                                    D8   11 0009A      BRB     3$                                  ;
    04  BC      0000'   CF  0000'    CF   0D 0009C 5$:   PROBEW  MODEADDR, GSD$W_SIZE2, @BUF         ; 0929
```

```
                                            1B  13 000A5 6$:    BEQL    8$
   0000'  CF         00          6E         00  2C 000A7        MOVC5   #0, (SP), #0, GSD$W_SIZE2, @BUF            ; 0931
                                      04    BC     000AE
                              04     AC         53  D0 000B0        MOVL    R3, BUF
                                              FF52  31 000B4        BRW     1$                                          ; 0853
                04     BC     0000'  CF    0000'  CF  0D 000B7 7$:    PROBEW  MODEADDR, GSD$W_SIZE2, @BUF                ; 0939
                                            04  12 000C0        BNEQ    9$
                                     50     0C  D0 000C2 8$:    MOVL    #12, R0                                     ; 0940
                                                04 000C5        RET
   0000'  CF         00          6E         00  2C 000C6 9$:    MOVC5   #0, (SP), #0, GSD$W_SIZE2, @BUF            ; 0941
                                      04    BC     000CD
                              04     AC         53  D0 000CF        MOVL    R3, BUF
                                     50         01  D0 000D3        MOVL    #1, R0                                      ; 0943
                                                04 000D6        RET                                                 ; 0944

; Routine Size:  215 bytes,    Routine Base:  $CODE$ + 052A

;  753          0945  1
```

```
;  755     0946  1 %SBTTL  'lists_shms';
;  756     0947  1
;  757     0948  1 ROUTINE LIST_SHMS (SHM_BUFFER) =
;  758     0949  1 !+++
;  759     0950  1 !
;  760     0951  1 !  FUNCTIONAL DESCRIPTION:
;  761     0952  1 !       Format and display to sys$output the shared memory GSD's
;  762     0953  1 !
;  763     0954  1 !  INPUT:
;  764     0955  1 !       shm_buffer =    buffer containing copies of shared memory GSD list
;  765     0956  1 !
;  766     0957  1 !---
;  767     0958  2 BEGIN
;  768     0959  2 OWN
;  769     0960  2     SPACES : LONG INITIAL (CSTRING ('        ')),
;  770     0961  2     DELPEND : LONG INITIAL (CSTRING (' DELPEND'));
;  771     0962  2
;  772     0963  2 LOCAL
;  773     0964  2     BUFFER : REF BBLOCK,
;  774     0965  2     GPT_ENTRIES,
;  775     0966  2     GSD_MAX,
;  776     0967  2     GSP_UNUSED,             ! Count of pages for global sections
;  777     0968  2     MATCHES,                ! Boolean used in controlling display
;  778     0969  2     PORT_NUMBER,            ! Number of memory port
;  779     0970  2     PRINT_TITLE,            ! Boolean used in controlling title
;  780     0971  2     SECTIONS,               ! Number of global sections
;  781     0972  2     STATUS;
;  782     0973  2
;  783     0974  2 BIND
;  784     0975  2     FAO_SHMTITLE        = $DESCRIPTOR (' !/!_Global Sections in Multiport Memory "!AC"!/'),
;  785     0976  2     FAO_CRPORT          = $DESCRIPTOR (' !AC Creator Port=!UB'),
;  786     0977  2     FAO_BASEPFN         = $DESCRIPTOR ('!_Basepfn/Pagcnt=!XL/!UL'),
;  787     0978  2     FAO_PROCCNT         = $DESCRIPTOR ('!_Port !UB PTE Refcnt=!UL'),
;  788     0979  2     FAO_SUMMARY1        = $DESCRIPTOR ('!7    !UL Global Page!%S Used,  !UL Global Page!%S Unused'),
;  789     0980  2     FAO_SUMMARY2        = $DESCRIPTOR ('    !UL Global Section!%S Used,  !UL Global Section!%S Unused');
;  790     0981  2
;  791     0982  2
;  792     0983  2 BUFFER = .SHM_BUFFER;
;  793     0984  2
;  794     0985  2 !
;  795     0986  2 !   If there were no shared memories then return success
;  796     0987  2 !
;  797     0988  2 IF .(BUFFER [GSD$W_SIZE])<0,16,1> EQL 0 THEN RETURN TRUE;
;  798     0989  2
;  799     0990  2 !
;  800     0991  2 !   For each shared memory
;  801     0992  2 !
;  802     0993  2 UNTIL .(BUFFER [GSD$W_SIZE])<0,16,1> EQL 0 DO   ! <0,16,1> Ensures signed compare
;  803     0994  3     BEGIN
;  804     0995  3     LOCAL
;  805     0996  3         TBUF : BBLOCK [16];                         ! Title buffer
;  806     0997  3
;  807     0998  3     PRINT_TITLE = TRUE;                             ! Initialize title control indicator
;  808     0999  3
;  809     1000  3     BUFFER = .BUFFER + $BYTEOFFSET (GSD$W_SIZE) + 2;    ! skip over -1's
;  810     1001  3
;  811     1002  3     SECTIONS = 0;
```

```
;  812          1003  3      GPT_ENTRIES = 0;
;  813          1004  3
;  814          1005  3      PORT_NUMBER = ..BUFFER;
;  815          1006  3      BUFFER = .BUFFER + 4;
;  816          1007  3
;  817          1008  3      GSP_UNUSED = ..BUFFER;
;  818          1009  3      BUFFER = .BUFFER + 4;
;  819          1010  3
;  820          1011  3      GSD_MAX = ..BUFFER;
;  821          1012  3      BUFFER = .BUFFER + 4;
;  822          1013  3
;  823          1014  3      CH$MOVE(16, .BUFFER, TBUF);                    ! Copy shared memory name
;  824          1015  3      BUFFER = .BUFFER + 16;
;  825          1016  3
;  826          1017  3      UNTIL .(BUFFER [GSD$W_SIZE])<0,16,1>  EQL 0 DO      ! loop through all GSD's
;  827          1018  4          BEGIN
;  828          1019  4          LOCAL
;  829          1020  4              BASPFNADR,
;  830          1021  4              PORT,
;  831          1022  4              PROC_REF_CNTS,
;  832          1023  4              PTECNTADR;
;  833          1024  4
;  834          1025  4          MAP
;  835          1026  4              KFE : REF $BBLOCK;
;  836          1027  4
;  837          1028  4          IF .KFE EQL 0
;  838          1029  4          THEN MATCHES = TRUE
;  839          1030  4          ELSE
;  840          1031  5              BEGIN
;  841          1032  5              !
;  842          1033  5              ! See if the KFE and GSD have matching names, less suffix
;  843          1034  5              !
;  844          1035  5              LOCAL
;  845          1036  5                  CME_ARGLIST : VECTOR [3,LONG];
;  846          1037  5
;  847          1038  5              CME_ARGLIST [0] = 2;
;  848          1039  5              CME_ARGLIST [1] = .KFE;
;  849          1040  5              CME_ARGLIST [2] = .BUFFER;
;  850          1041  5
;  851          1042  5              MATCHES = $CMEXEC (ROUTIN=EXEC_MODE_NAME_CHK, ARGLST=CME_ARGLIST);
;  852          1043  5
;  853          1044  4              END;
;  854          1045  4
;  855          1046  4          IF .MATCHES
;  856          1047  4          THEN
;  857          1048  5              BEGIN
;  858          1049  5              IF .PRINT_TITLE
;  859          1050  5              THEN
;  860          1051  6                  BEGIN                          ! Print title if not already done
;  861          1052  6                  INS$FAO_AND_OUT (FAO_SHMTITLE, TBUF);
;  862          1053  6                  PRINT_TITLE = FALSE;
;  863          1054  5                  END;
;  864          1055  5
;  865          1056  5              SECTIONS = .SECTIONS + 1;
;  866          1057  5              DECODE_FLAGS (.BUFFER);
;  867          1058  5              INS$FAO_AND_OUT(FAO_CRPORT,
;  868          1059  6                  (IF .BUFFER [GSD$V_DELPEND]
```

```
;  869          1060  6              THEN .DELPEND
;  870          1061  5              ELSE .SPACES ),
;  871          1062  5              .BUFFER [GSD$B_CREATPORT]);
;  872          1063  5
;  873          1064  5          BASPFNADR = BUFFER [GSD$L_BASPFN1];          ! Address of first PFN base
;  874          1065  5          INCR I FROM 0 TO GSD$C_PFNBASMAX - 1 DO      ! For the maximum number of PFN bases allowed
;  875          1066  6              BEGIN
;  876          1067  6              BIND
;  877          1068  6                  PFN_BASE = .BASPFNADR,               ! PFN base
;  878          1069  6                  BASE_CNT = .BASPFNADR + 4;           ! count of pages at this base
;  879          1070  6
;  880          1071  6              IF .BASE_CNT  EQL 0 THEN EXITLOOP;
;  881          1072  6              GPT_ENTRIES = .GPT_ENTRIES + .BASE_CNT; ! count of global section pages used
;  882          1073  6              INS$FAO_AND_OUT(FAO_BASEPFN, .PFN_BASE, .BASE_CNT);
;  883          1074  6              BASPFNADR = .BASPFNADR + 8;              ! skip over count to next PFN base
;  884          1075  5              END;
;  885          1076  5
;  886          1077  5          PROC_REF_CNTS = .BUFFER [GSD$B_PROCCNT];     ! Processor reference counts
;  887          1078  5          PTECNTADR = BUFFER [GSD$L_PTECNT1];          ! Address of first reference count
;  888          1079  5          PORT = 0;
;  889          1080  5          WHILE (PROC_REF_CNTS = .PROC_REF_CNTS - 1) GEQ 0 DO ! For all active ports
;  890          1081  6              BEGIN
;  891          1082  6              LOCAL
;  892          1083  6                  PTE_CNT;
;  893          1084  6
;  894          1085  6              PTE_CNT = ..PTECNTADR;                   ! get processor reference count
;  895          1086  6              IF .PORT EQL .PORT_NUMBER                 ! Is this the port for this processor ?
;  896          1087  6              THEN PTE_CNT = .PTE_CNT - 1;             !   Then subtract one for the GSD lock
;  897          1088  6              IF .PTE_CNT GTR 0                        ! skip if none mapped to section
;  898          1089  6              THEN
;  899          1090  6                  INS$FAO_AND_OUT(FAO_PROCCNT, .PORT, .PTE_CNT);
;  900          1091  6
;  901          1092  6                  PTECNTADR = .PTECNTADR + 4;           ! Go to address of next processor reference count
;  902          1093  6                  PORT = .PORT + 1;                    ! Do for all active ports
;  903          1094  5                  END;
;  904          1095  4              END;
;  905          1096  4
;  906          1097  4          BUFFER = .BUFFER + .BUFFER [GSD$W_SIZE];
;  907          1098  4
;  908          1099  3          END;                                         ! Until no more GSDs
;  909          1100  3
;  910          1101  3      IF .KFE EQL 0
;  911          1102  3      THEN
;  912          1103  4          BEGIN
;  913          1104  4          !
;  914          1105  4          !   Print title if not already done
;  915          1106  4          !
;  916          1107  4          IF .PRINT_TITLE
;  917          1108  4          THEN
;  918          1109  4              INS$FAO_AND_OUT (FAO_SHMTITLE, TBUF);
;  919          1110  4          !
;  920          1111  4          !   Print summary lines
;  921          1112  4          !
;  922          1113  4          GSP_UNUSED = .GSP_UNUSED - .GPT_ENTRIES;
;  923          1114  4          INS$FAO_AND_OUT( FAO_SUMMARY1, .GPT_ENTRIES, .GSP_UNUSED);
;  924          1115  4          INS$FAO_AND_OUT( FAO_SUMMARY2, .SECTIONS, (.GSD_MAX - .SECTIONS));
;  925          1116  3          END;
```

```
;  926          1117  3
;  927          1118  3     BUFFER = .BUFFER + .GSD$W_SIZE2                    ! skip over zeros
;  928          1119  2     END;
;  929          1120  2
;  930          1121  2 RETURN TRUE;
;  931          1122  1 END;    ! Routine  list_shms


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                    08  00120 P.AAQ:  .BYTE   8
                                20  20  20  20  20  20  20  20  00121        .ASCII  \        \
                                                    08  00129 P.AAR:  .BYTE   8
                                44  4E  45  50  4C  45  44  20  0012A        .ASCII  \ DELPEND\
63  65  53  20  6C  61  62  6F  6C  47  5F  21  2F  21  20  00132 P.AAT:  .ASCII  \ !/!_Global Sections in Multiport Memory\
70  69  74  6C  75  4D  20  6E  69  20  73  6E  6F  69  74  00141
                    79  72  6F  6D  65  4D  20  74  72  6F  00150
                            2F  21  22  43  41  21  22  20  0015A        .ASCII  \ "!AC"!/\
                                                        00162        .BLKB   2
                                        00000030  00164 P.AAS:  .LONG   48
                                        00000000' 00168        .ADDRESS P.AAT
6F  50  20  72  6F  74  61  65  72  43  20  43  41  21  20  0016C P.AAV:  .ASCII  \ !AC Creator Port=!UB\
                                42  55  21  3D  74  72  0017B
                                                        00181        .BLKB   3
                                        00000015  00184 P.AAU:  .LONG   21
                                        00000000' 00188        .ADDRESS P.AAV
6E  63  67  61  50  2F  6E  66  70  65  73  61  42  5F  21  0018C P.AAX:  .ASCII  \!_Basepfn/Pagcnt=!XL/!UL\
                        4C  55  21  2F  4C  58  21  3D  74  0019B
                                        00000018  001A4 P.AAW.  .LONG   24
                                        00000000' 001A8        .ADDRESS P.AAX
20  45  54  50  20  42  55  21  20  74  72  6F  50  5F  21  001AC P.AAZ:  .ASCII  \!_Port !UB PTE Refcnt=!UL\
                        4C  55  21  3D  74  6E  63  66  65  52  001BB
                                                        001C5        .BLKB   3
                                        00000019  001C8 P.AAY:  .LONG   25
                                        00000000' 001CC        .ADDRESS P.AAZ
61  62  6F  6C  47  20  4C  55  21  20  20  20  20  2F  21  001D0 P.ABB:  .ASCII  \!/    !UL Global Page!%S Used,  !UL Glob\
2C  64  65  73  55  20  53  25  21  65  67  61  50  20  6C  001DF
                    62  6F  6C  47  20  4C  55  21  20  20  001EE
73  75  6E  55  20  53  25  21  65  67  61  50  20  6C  61  001F8        .ASCII  \al Page!%S Unused\
                                                    64  65  00207
                                                        00209        .BLKB   3
                                        00000039  0020C P.ABA:  .LONG   57
                                        00000000' 00210        .ADDRESS P.ABB
20  6C  61  62  6F  6C  47  20  4C  55  21  20  20  20  20  00214 P.ABD:  .ASCII  \    !UL Global Section!%S Used,  !UL Glo\
64  65  73  55  20  53  25  21  6E  6F  69  74  63  65  53  00223
                    6F  6C  47  20  4C  55  21  20  20  2C  00232
20  53  25  21  6E  6F  69  74  63  65  53  20  6C  61  62  0023C        .ASCII  \bal Section!%S Unused\
                                64  65  73  75  6E  55  0024B
                                                        00251        .BLKB   3
                                        0000003D  00254 P.ABC:  .LONG   61
                                        00000000' 00258        .ADDRESS P.ABD

                                                        .PSECT  $OWN$,NOEXE,2

                                        00000000' 00030 SPACES: .ADDRESS P.AAQ
                                        00000000' 00034 DELPEND:.ADDRESS P.AAR
```

```
                                      FAO_SHMTITLE=      P.AAS
                                      FAO_CRPORT=        P.AAU
                                      FAO_BASEPFN=       P.AAW
                                      FAO_PROCCNT=       P.AAY
                                      FAO_SUMMARY1=      P.ABA
                                      FAO_SUMMARY2=      P.ABC

                                              .PSECT  $CODE$,NOWRT,2

                           0FFC 00000 LIST_SHMS:
                                              .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 0948
              5E          24 C2 00002         SUBL2   #36, SP                                 ;
              56     04   AC D0 00005         MOVL    SHM_BUFFER, BUFFER                      ; 0983
                     08   A6 B5 00009         TSTW    8(BUFFER)                               ; 0988
                          03 13 0000C         BEQL    2$                                      ;
                     08   A6 B5 0000E 1$:     TSTW    8(BUFFER)                               ; 0993
                          03 12 00011 2$:     BNEQ    3$                                      ;
                        0122 31 00013         BRW     21$                                     ;
              56          0A C0 00016 3$:     ADDL2   #10, BUFFER                             ; 1000
                          57 D4 00019         CLRL    SECTIONS                                ; 1002
              58          01 7D 0001B         MOVQ    #1, PRINT_TITLE                         ; 0998
              5B          86 D0 0001E         MOVL    (BUFFER)+, PORT_NUMBER                  ; 1005
              5A          86 D0 00021         MOVL    (BUFFER)+, GSP_UNUSED                   ; 1008
         04   AE          86 D0 00024         MOVL    (BUFFER)+, GSD_MAX                      ; 1011
 14   AE      66          10 28 00028         MOVC3   #16, (BUFFER), TBUF                     ; 1014
              56          10 C0 0002D         ADDL2   #16, BUFFER                             ; 1015
                     08   A6 B5 00030 4$:     TSTW    8(BUFFER)                               ; 1017
                          03 12 00033         BNEQ    5$                                      ;
                        00C1 31 00035         BRW     18$                                     ;
              50   0000'  CF D0 00038 5$:     MOVL    KFE, R0                                 ; 1028
                          05 12 0003D         BNEQ    6$                                      ;
              6E          01 D0 0003F         MOVL    #1, MATCHES                             ; 1029
                          1D 11 00042         BRB     7$                                      ;
         08   AE          02 D0 00044 6$:     MOVL    #2, CME_ARGLIST                         ; 1038
         0C   AE          50 D0 00048         MOVL    R0, CME_ARGLIST+4                       ; 1039
         10   AE          56 D0 0004C         MOVL    BUFFER, CME_ARGLIST+8                   ; 1040
                     08   AE 9F 00050         PUSHAB  CME_ARGLIST                             ; 1042
                   FE9A   CF 9F 00053         PUSHAB  EXEC_MODE_NAME_CHK                      ;
   00000000G  00          02 FB 00057         CALLS   #2, SYS$CMEXEC                          ;
              6E          50 D0 0005E         MOVL    R0, MATCHES                             ;
              03          6E E8 00061 7$:     BLBS    MATCHES, 8$                             ; 1046
                        0088 31 00064         BRW     17$                                     ;
              0E          58 E9 00067 8$:     BLBC    PRINT_TITLE, 9$                         ; 1049
                     14   AE 9F 0006A         PUSHAB  TBUF                                    ; 1052
                   0000'  CF 9F 0006D         PUSHAB  FAO_SHMTITLE                            ;
       0000V  CF          02 FB 00071         CALLS   #2, INS$FAO_AND_OUT                     ;
                          58 D4 00076         CLRL    PRINT_TITLE                             ; 1053
                          57 D6 00078 9$:     INCL    SECTIONS                                ; 1056
                          56 DD 0007A         PUSHL   BUFFER                                  ; 1057
       0000V  CF          01 FB 0007C         CALLS   #1, DECODE_FLAGS                        ;
              7E     52   A6 9A 00081         MOVZBL  82(BUFFER), -(SP)                       ; 1062
      06      66          02 E1 00085         BBC     #2, (BUFFER), 10$                       ; 1059
                   0000'  CF DD 00089         PUSHL   DELPEND                                 ; 1060
                          04 11 0008D         BRB     11$                                     ;
                   0000'  CF DD 0008F 10$:    PUSHL   SPACES                                  ; 1061
```

```
                                0000'  CF  9F 00093 11$:    PUSHAB  FAO_CRPORT                      ; 1058
                         0000V  CF         03  FB 00097          CALLS   #3, INS$FAO_AND_OUT
                            52     54     A6  9E 0009C          MOVAB   84(R6), BASPFNADR            ; 1064
                                          53  D4 000A0          CLRL    I                            ; 1065
                            50     04     A2  D0 000A2 12$:     MOVL    4(BASPFNADR), R0             ; 1071
                                          17  13 000A6          BEQL    13$
                            59            50  C0 000A8          ADDL2   R0, GPT_ENTRIES              ; 1072
                                          50  DD 000AB          PUSHL   R0                           ; 1073
                                          62  DD 000AD          PUSHL   (BASPFNADR)
                                0000'  CF  9F 000AF          PUSHAB  FAO_BASEPFN
                         0000V  CF         03  FB 000B3          CALLS   #3, INS$FAO_AND_OUT
                            52            08  C0 000B8          ADDL2   #8, BASPFNADR                ; 1074
                     E3     53            03  F3 000BB          AOBLEQ  #3, I, 12$                   ; 1065
                            54     51     A6  9A 000BF 13$:     MOVZBL  81(BUFFER), PROC_REF_CNTS    ; 1077
                            52     74     A6  9E 000C3          MOVAB   116(R6), PTECNTADR           ; 1078
                                          53  D4 000C7          CLRL    PORT                         ; 1079
                                          54  D7 000C9 14$:     DECL    PROC_REF_CNTS                ; 1080
                                          22  19 000CB          BLSS    17$
                            50            62  D0 000CD          MOVL    (PTECNTADR), PTE_CNT         ; 1085
                            58            53  D1 000D0          CMPL    PORT, PORT_NUMBER            ; 1086
                                          02  12 000D3          BNEQ    15$
                                          50  D7 000D5          DECL    PTE_CNT                      ; 1087
                                          50  D5 000D7 15$:     TSTL    PTE_CNT                      ; 1088
                                          0D  15 000D9          BLEQ    16$
                                          50  DD 000DB          PUSHL   PTE_CNT                      ; 1090
                                          53  DD 000DD          PUSHL   PORT
                                0000'  CF  9F 000DF          PUSHAB  FAO_PROCCNT
                         0000V  CF         03  FB 000E3          CALLS   #3, INS$FAO_AND_OUT
                            52            04  C0 000E8 16$:     ADDL2   #4, PTECNTADR                ; 1092
                                          53  D6 000EB          INCL    PORT                         ; 1093
                                          DA  11 000ED          BRB     14$                          ; 1080
                            50     08     A6  3C 000EF 17$:     MOVZWL  8(BUFFER), R0                ; 1097
                            56            50  C0 000F3          ADDL2   R0, BUFFER
                                      FF37  31 000F6          BRW     4$                           ; 1017
                                0000'  CF  D5 000F9 18$:     TSTL    KFE                          ; 1101
                                          2E  12 000FD          BNEQ    20$
                                      0C  58  E9 000FF          BLBC    PRINT_TITLE, 19$             ; 1107
                                    14    AE  9F 00102          PUSHAB  TBUF                         ; 1109
                                0000'  CF  9F 00105          PUSHAB  FAO_SHMTITLE
                         0000V  CF         02  FB 00109          CALLS   #2, INS$FAO_AND_OUT
                            5A            59  C2 0010E 19$:     SUBL2   GPT_ENTRIES, GSP_UNUSED      ; 1113
                            7E            59  7D 00111          MOVQ    GPT_ENTRIES, -(SP)           ; 1114
                                0000'  CF  9F 00114          PUSHAB  FAO_SUMMARY1
                         0000V  CF         03  FB 00118          CALLS   #3, INS$FAO_AND_OUT
                     7E     04     AE        57  C3 0011D          SUBL3   SECTIONS, GSD_MAX, -(SP)     ; 1115
                                          57  DD 00122          PUSHL   SECTIONS
                                0000'  CF  9F 00124          PUSHAB  FAO_SUMMARY2
                         0000V  CF         03  FB 00128          CALLS   #3, INS$FAO_AND_OUT
                            50  0000'  CF  3C 0012D 20$:     MOVZWL  GSD$W_SIZE2, R0              ; 1118
                            56            50  C0 00132          ADDL2   R0, BUFFER
                                      FED6  31 00135          BRW     1$
                            50            01  D0 00138 21$:     MOVL    #1, R0                       ; 1121
                                              04 0013B          RET                                  ; 1122

; Routine Size:  316 bytes,    Routine Base:  $CODE$ + 0401
```

```
;  932          lists_shms
                1123  1
```

```
;  934          1124  1 %SBTTL  'Decode_flags';
;  935          1125  1
;  936          1126  1 ROUTINE DECODE_FLAGS ( GSD ) =
;  937          1127  2 BEGIN
;  938          1128  2 !+++
;  939          1129  2 !
;  940          1130  2 !  FUNCTIONAL DESCRIPTION:
;  941          1131  2 !       Format the ASCII symbols in the output buffer
;  942          1132  2 !       for the gsd flags that are set.
;  943          1133  2 !
;  944          1134  2 !  INPUT:
;  945          1135  2 !       gsd = address in buffer of GSD flags word
;  946          1136  2 !
;  947          1137  2 !---
;  948          1138  2 MAP
;  949          1139  2     GSD : REF BBLOCK;
;  950          1140  2
;  951          1141  2 !
;  952          1142  2 !   Indices to ctlflg_array
;  953          1143  2 !
;  954          1144  2 LITERAL
;  955          1145  2     SEC_I_WRT = 0,
;  956          1146  2     SEC_I_CRF = 2,
;  957          1147  2     SEC_I_DZRO = 4,
;  958          1148  2     SEC_I_PRM = 6,
;  959          1149  2     SEC_I_TMP = 7,
;  960          1150  2     SEC_I_SYS = 8,
;  961          1151  2     SEC_I_GRP = 9;
;  962          1152  2
;  963          1153  2 OWN
;  964          1154  2
;  965          1155  2     !
;  966          1156  2     !   Table of counted ASCII strings, one string to be output
;  967          1157  2     !   if flag is set, another if it is not set
;  968          1158  2     !
;  969          1159  2     CTLFLAG_ARRAY : VECTOR [2 * INS$C_CTLFLGSTR]  INITIAL (
;  970          1160  2                 CSTRING ('WRT'), CSTRING ('   '),
;  971          1161  2                 CSTRING ('CRF'), CSTRING ('   '),
;  972          1162  2                 CSTRING ('DZRO'), CSTRING ('    '),
;  973          1163  2                 CSTRING ('PRM'), CSTRING ('TMP'),
;  974          1164  2                 CSTRING ('SYS'), CSTRING ('GRP')
;  975          1165  2                 ),
;  976          1166  2     PFN_CSTR : LONG INITIAL( CSTRING ('PFN '));
;  977          1167  2
;  978          1168  2 BIND
;  979          1169  2  FAO_GBLNAM   = $DESCRIPTOR ('!#AC(!XL) !AC !AC !AC !AC !AC'),
;  980          1170  2  FAO_GBLNAM_L = $DESCRIPTOR ('!#AC!/                (!XL) !AC !AC !AC !AC !AC');
;  981          1171  2 !
;  982          1172  2 !   Place the correct ASCII symbols in the output buffer.
;  983          1173  2 !
;  984          1174  2 INS$FAOL (
;  985          1175  3         (IF .GSD [GSD$B_TYPE] EQL DYN$C_EXTGSD
;  986          1176  3                 then
;  987          1177  4                 (IF .(GSD [GSD$T_PFNGSDNAM]) <0,8,0> LEQ 15
;  988          1178  4                         then FAO_GBLNAM
;  989          1179  4                         else FAO_GBLNAM_L)
;  990          1180  3                 else
```

```
;  991          1181  4                 (IF .(GSD [GSD$T_GSDNAM]) <0,8,0> LEQ 15
;  992          1182  4                       then FAO_GBLNAM
;  993          1183  2                       else FAO_GBLNAM_L) ),
;  994          1184  3           (IF .GSD [GSD$B_TYPE] EQL DYN$C_EXTGSD
;  995          1185  3                 then
;  996          1186  4                 (IF .(GSD [GSD$T_PFNGSDNAM]) <0,8,0> LEQ 15
;  997          1187  4                       then 15
;  998          1188  4                       else .(GSD [GSD$T_PFNGSDNAM]) <0,8,0>)
;  999          1189  3                 else
; 1000          1190  4                 (IF .(GSD [GSD$T_GSDNAM]) <0,8,0> LEQ 15
; 1001          1191  4                       then 15
; 1002          1192  2                       else .(GSD [GSD$T_GSDNAM]) <0,8,0>) ),
; 1003          1193  3           (IF .GSD [GSD$B_TYPE] EQL DYN$C_EXTGSD
; 1004          1194  3                 then GSD [GSD$T_PFNGSDNAM]
; 1005          1195  2                 else GSD [GSD$T_GSDNAM]),
; 1006          1196  2           .GSD [GSD$L_IDENT],
; 1007          1197  3           (IF (.GSD [GSD$W_FLAGS] AND SEC$M_WRT) NEQ 0
; 1008          1198  3                 then .CTLFLAG_ARRAY [SEC_I_WRT]
; 1009          1199  2                 else .CTLFLAG_ARRAY [SEC_I_WRT + 1]),
; 1010          1200  3           (IF (.GSD [GSD$W_FLAGS] AND SEC$M_CRF) NEQ 0
; 1011          1201  3                 then .CTLFLAG_ARRAY [SEC_I_CRF]
; 1012          1202  2                 else .CTLFLAG_ARRAY [SEC_I_CRF + 1]),
; 1013          1203  3           (IF (.GSD [GSD$W_FLAGS] AND SEC$M_DZRO) NEQ 0
; 1014          1204  3                 THEN
; 1015          1205  4                     BEGIN
; 1016          1206  4                     IF .GSD [GSD$B_TYPE] EQL DYN$C_EXTGSD
; 1017          1207  4                     THEN .PFN_CSTR
; 1018          1208  4                     ELSE .CTLFLAG_ARRAY [SEC_I_DZRO]
; 1019          1209  4                     END
; 1020          1210  2                 ELSE .CTLFLAG_ARRAY [SEC_I_DZRO + 1]),
; 1021          1211  3           (IF (.GSD [GSD$W_FLAGS] AND SEC$M_PERM) NEQ 0
; 1022          1212  3                 THEN .CTLFLAG_ARRAY [SEC_I_PRM]
; 1023          1213  2                 ELSE .CTLFLAG_ARRAY [SEC_I_TMP]),
; 1024          1214  3           (IF (.GSD [GSD$W_FLAGS] AND SEC$M_SYSGBL) NEQ 0
; 1025          1215  3                 THEN .CTLFLAG_ARRAY [SEC_I_SYS]
; 1026          1216  2                 ELSE .CTLFLAG_ARRAY [SEC_I_GRP])        );
; 1027          1217  2
; 1028          1218  2 RETURN TRUE;
; 1029          1219  1 END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                 03  0025C P.ABE:  .BYTE   3                                    ;
                                              54 52  57  0025D        .ASCII  \WRT\                               ;
                                                 03  00260 P.ABF:  .BYTE   3                                    ;
                                              20 20  20  00261        .ASCII  \   \                               ;
                                                 03  00264 P.ABG:  .BYTE   3                                    ;
                                              46 52  43  00265        .ASCII  \CRF\                               ;
                                                 03  00268 P.ABH:  .BYTE   3                                    ;
                                              20 20  20  00269        .ASCII  \   \                               ;
                                                 04  0026C P.ABI:  .BYTE   4                                    ;
                                           4F 52 5A  44  0026D        .ASCII  \DZRO\                              ;
                                                 04  00271 P.ABJ:  .BYTE   4                                    ;
                                           20 20 20  20  00272        .ASCII  \    \                              ;
                                                 03  00276 P.ABK:  .BYTE   3                                    ;
```

```
                                                      4D  52  50  00277          .ASCII  \PRM\                          ;
                                                              03  0027A P.ABL:   .BYTE   3                              ;
                                                      50  4D  54  0027B          .ASCII  \TMP\                          ;
                                                              03  0027E P.ABM:   .BYTE   3                              ;
                                                      53  59  53  0027F          .ASCII  \SYS\                          ;
                                                              03  00282 P.ABN:   .BYTE   3                              ;
                                                      50  52  47  00283          .ASCII  \GRP\                          ;
                                                              04  00286 P.ABO:   .BYTE   4                              ;
                                                  20  4E  46  50  00287          .ASCII  \PFN \                         ;
21  20  43  41  21  20  29  4C  58  21  28  43  41  23  21  0028B P.ABQ:   .ASCII  \!#AC(!XL) .AC .AC !AC !AC !AC\  ;
    43  41  21  20  43  41  21  20  43  41  21  20  43  41  0029A                                                   ;
                                                   0000001D  002A8 P.ABP:   .LONG   29                             ;
                                                   00000000' 002AC          .ADDRESS P.ABQ                         ;
20  20  20  20  20  20  20  20  20  2F  21  43  41  23  21  002B0 P.ABS:   .ASCII  \!#AC!/           (!XL) !AC !AC !AC !\  ;
43  41  21  20  29  4C  58  21  28  20  20  20  20  20  20  002BF                                                   ;
                        21  20  43  41  21  20  43  41  21  20  002CE                                               ;
                                        43  41  21  20  43  41  002D8          .ASCII  \AC !AC\                     ;
                                                            002DE          .BLKB   2
                                                   0000002E  002E0 P.ABR:   .LONG   46                             ;
                                                   00000000' 002E4          .ADDRESS P.ABS                         ;

                                                                            .PSECT  $OWN$,NOEXE,2

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00038 CTLFLAG_ARRAY:
                                                                            .ADDRESS P.ABE, P.ABF, P.ABG, P.ABH, P.ABI, -  ;
                    00000000' 00000000' 00000000' 00000000' 00050                    P.ABJ, P.ABK, P.ABL, P.ABM, P.ABN      ;
                                                            00060          .BLKB   8
                                                   00000000' 00068 PFN_CSTR:
                                                                            .ADDRESS P.ABO                         ;

                                                                   FAO_GBLNAM=          P.ABP
                                                                   FAO_GBLNAM_L=        P.ABR


                                                                            .PSECT  $CODE$,NOWRT,2

                                                       000C  00000 DECODE_FLAGS:
                                                                            .WORD   Save R2,R3                     ; 1126
                          53       0000'  CF  9E  00002          MOVAB   CTLFLAG_ARRAY+32, R3                   ;
                          50          04  AC  D0  00007          MOVL    GSD, R0                                ; 1214
                          52          20  A0  3C  0000B          MOVZWL  32(R0), R2                             ;
                                              52  B5  0000F          TSTW    R2                                     ;
                                              04  18  00011          BGEQ    1$                                     ;
                                              63  DD  00013          PUSHL   CTLFLAG_ARRAY+32                       ; 1215
                                              03  11  00015          BRB     2$                                     ;
                                      04  A3  DD  00017 1$:      PUSHL   CTLFLAG_ARRAY+36                       ; 1216
              05          52              0E  E1  0001A 2$:      BBC     #14, R2, 3$                            ; 1211
                                      F8  A3  DD  0001E          PUSHL   CTLFLAG_ARRAY+24                       ; 1212
                                              03  11  00021          BRB     4$                                     ;
                                      FC  A3  DD  00023 3$:      PUSHL   CTLFLAG_ARRAY+28                       ; 1213
              14          52              02  E1  00026 4$:      BBC     #2, R2, 7$                             ; 1203
                          28          0A  A0  91  0002A          CMPB    10(R0), #40                            ; 1206
                                              06  12  0002E          BNEQ    5$                                     ;
                          51          10  A3  D0  00030          MOVL    PFN_CSTR, R1                           ; 1207
                                              04  11  00034          BRB     6$                                     ;
                          51          F0  A3  D0  00036 5$:      MOVL    CTLFLAG_ARRAY+16, R1                   ; 1208
                                              51  DD  0003A 6$:      PUSHL   R1                                     ; 1205
```

```
                     03  11 0003C          BRB     8$                                          ;
                F4   A3  DD 0003E 7$:       PUSHL   CTLFLAG_ARRAY+20                            ; 1210
          05    52   01  E1 00041 8$:       BBC     #1, R2, 9$                                  ; 1200
                E8   A3  DD 00045          PUSHL   CTLFLAG_ARRAY+8                             ; 1201
                     03  11 00048          BRB     10$                                         ;
                EC   A3  DD 0004A 9$:       PUSHL   CTLFLAG_ARRAY+12                            ; 1202
          05    52   03  E1 0004D 10$:      BBC     #3, R2, 11$                                 ; 1197
                E0   A3  DD 00051          PUSHL   CTLFLAG_ARRAY                               ; 1198
                     03  11 00054          BRB     12$                                         ;
                E4   A3  DD 00056 11$:      PUSHL   CTLFLAG_ARRAY+4                             ; 1199
                18   A0  DD 00059 12$:      PUSHL   24(R0)                                      ; 1196
                     52  D4 0005C          CLRL    R2                                          ; 1193
          28    0A   A0  91 0005E          CMPB    10(R0), #40                                 ;
                     08  12 00062          BNEQ    13$                                         ;
                     52  D6 00064          INCL    R2                                          ;
          51   30   A0  9E 00066          MOVAB   48(R0), R1                                  ; 1194
                     04  11 0006A          BRB     14$                                         ;
          51   22   A0  9E 0006C 13$:      MOVAB   34(R0), R1                                  ; 1195
                     51  DD 00070 14$:      PUSHL   R1                                          ;
          0C         52  E9 00072          BLBC    R2, 15$                                     ; 1193
          0F   30   A0  91 00075          CMPB    48(R0), #15                                 ; 1186
                     0C  1B 00079          BLEQU   16$                                         ;
          51   30   A0  9A 0007B          MOVZBL  48(R0), R1                                  ; 1188
                     0F  11 0007F          BRB     18$                                         ; 1186
          0F   22   A0  91 00081 15$:      CMPB    34(R0), #15                                 ; 1190
                     05  1A 00085          BGTRU   17$                                         ;
          51         0F  D0 00087 16$:      MOVL    #15, R1                                     ;
                     04  11 0008A          BRB     18$                                         ;
          51   22   A0  9A 0008C 17$:      MOVZBL  34(R0), R1                                  ; 1192
                     51  DD 00090 18$:      PUSHL   R1                                          ; 1190
          06         52  E9 00092          BLBC    R2, 19$                                     ; 1184
          0F   30   A0  91 00095          CMPB    48(R0), #15                                 ; 1177
                     04  11 00099          BRB     20$                                         ;
          0F   22   A0  91 0009B 19$:      CMPB    34(R0), #15                                 ; 1181
                     07  1A 0009F 20$:      BGTRU   21$                                         ;
          50   0000'  CF  9E 000A1          MOVAB   FAO_GBLNAM, R0                              ;
                     05  11 000A6          BRB     22$                                         ;
          50   0000'  CF  9E 000A8 21$:      MOVAB   FAO_GBLNAM_L, R0                            ;
                     50  DD 000AD 22$:      PUSHL   R0                                          ;
     0000V CF        09  FB 000AF          CALLS   #9, INS$FAOL                                ; 1175
          50         01  D0 000B4          MOVL    #1, R0                                      ; 1218
                     04  000B7          RET                                                 ; 1219

; Routine Size:  184 bytes,    Routine Base:  $CODE$ + 053D

; 1030          1220  1
```

```
;  1032          1221  1 %SBTTL  'output routines ';
;  1033          1222  1
;  1034          1223  1 GLOBAL ROUTINE INS$FAOL (FAO_STRING, PARAMETER_LIST) =
;  1035          1224  2 BEGIN
;  1036          1225  2 !+++
;  1037          1226  2 !
;  1038          1227  2 !  FUNCTIONAL DESCRIPTION:
;  1039          1228  2 !       Format an ASCII string and stuff it into the output buffer.
;  1040          1229  2 !       Update the buffer pointers to reflect the new stuff in the
;  1041          1230  2 !       buffer.
;  1042          1231  2 !
;  1043          1232  2 !  INPUT:
;  1044          1233  2 !       fao_string =    Formatted Ascii Output control string for FAO
;  1045          1234  2 !       parameter_list= List of stuff to have formatted into buffer.
;  1046          1235  2 !
;  1047          1236  2 !  IMPLICIT INPUT:
;  1048          1237  2 !       Output buffer has been allocated and ins$faobufdesc is the
;  1049          1238  2 !       descriptor for it.
;  1050          1239  2 !
;  1051          1240  2 !  OUTPUT:
;  1052          1241  2 !       none
;  1053          1242  2 !
;  1054          1243  2 !  ROUTINE VALUE
;  1055          1244  2 !       Success, or error status from SYS$FAOL
;  1056          1245  2 !---
;  1057          1246  2 LOCAL
;  1058          1247  2     OUTLEN : WORD;
;  1059          1248  2
;  1060          1249  2 EXECUTE ( SYS$FAOL (.FAO_STRING, OUTLEN, INS$FAOBUFDESC, PARAMETER_LIST));      ! Format the buffer
;  1061          1250  2 INS$FAOBUFDESC [DSC$W_LENGTH] = .INS$FAOBUFDESC [DSC$W_LENGTH] - .OUTLEN;      ! decrement space left in bu
;  1062          1251  2 INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOBUFDESC [DSC$A_POINTER] + .OUTLEN;    ! Point to unused space left
;  1063          1252  2 RETURN TRUE;
;  1064          1253  1 END;              ! global routine  INS$FAOL


                                                    0004 00000          .ENTRY  INS$FAOL, Save R2                               ; 1223
                                  52 00000000G   00  9E 00002          MOVAB   INS$FAOBUFDESC, R2                              ;
                                  5E             04  C2 00009          SUBL2   #4, SP                                          ;
                                           08    AC  9F 0000C          PUSHAB  PARAMETER_LIST                                  ; 1249
                                                 52  DD 0000F          PUSHL   R2                                              ;
                                           08    AE  9F 00011          PUSHAB  OUTLEN                                          ;
                                           04    AC  DD 00014          PUSHL   FAO_STRING                                      ;
                              00000000G    00        04  FB 00017          CALLS   #4, SYS$FAOL                                    ;
                                           0D        50  E9 0001E          BLBC    STATUS, 1$                                      ;
                                           62        6E  A2 00021          SUBW2   OUTLEN, INS$FAOBUFDESC                          ; 1250
                                           50        6E  3C 00024          MOVZWL  OUTLEN, R0                                      ; 1251
                                     04    A2        50  C0 00027          ADDL2   R0, INS$FAOBUFDESC+4                            ;
                                           50        01  D0 0002B          MOVL    #1, R0                                          ; 1252
                                                     04  0002E 1$:        RET                                                     ; 1253

; Routine Size:  47 bytes,    Routine Base:  $CODE$ + 05F5

; 1065          1254  1
```

```
;  1067          1255  1 GLOBAL ROUTINE INS$OUTPUT_FAOBUF =
;  1068          1256  2 BEGIN
;  1069          1257  2 !+++
;  1070          1258  2 !
;  1071          1259  2 !  FUNCTIONAL DESCRIPTION:
;  1072          1260  2 !       Print the contents of the output buffer to sys$output and re-initialize
;  1073          1261  2 !       the descriptor of the buffer, and zero the buffer.
;  1074          1262  2 !
;  1075          1263  2 !  INPUT:
;  1076          1264  2 !       none
;  1077          1265  2 !
;  1078          1266  2 !  IMPLICIT INPUT:
;  1079          1267  2 !       Output buffer has been allocated and ins$faobufdesc is the
;  1080          1268  2 !       descriptor for it.
;  1081          1269  2 !
;  1082          1270  2 !  OUTPUT:
;  1083          1271  2 !       Output the contents of ins$faooutbuf to sys$output
;  1084          1272  2 !
;  1085          1273  2 !  ROUTINE VALUE
;  1086          1274  2 !       status from $PUT
;  1087          1275  2 !---
;  1088          1276  2
;  1089          1277  2 LOCAL
;  1090          1278  2     STATUS;
;  1091          1279  2
;  1092          1280  2 INS$G_OUTRAB [RAB$W_RSZ] = INS$C_FAOBUFLEN - .INS$FAOBUFDESC [DSC$W_LENGTH];
;  1093          1281  2 STATUS = $PUT (RAB = INS$G_OUTRAB);
;  1094          1282  2 INS$FAOBUFDESC [DSC$W_LENGTH] = INS$C_FAOBUFLEN;
;  1095          1283  2 INS$FAOBUFDESC [DSC$A_POINTER] = .INS$FAOOUTBUF;
;  1096          1284  2 CH$FILL (%C' ', INS$C_FAOBUFLEN, .INS$FAOOUTBUF);
;  1097          1285  2 RETURN .STATUS;
;  1098          1286  1 END;                ! Global Routine  INS$OUTPUT_FAOBUF


                                                                .EXTRN  SYS$PUT

                                                                .ENTRY  INS$OUTPUT_FAOBUF, Save R2,R3,R4,R5,R6,R7   ; 1255
                                                    57 00000000G  00    9E  00002   MOVAB   INS$FAOBUFDESC, R7
                                    00000000G  00       0000G   8F      67  A3  00009   SUBW3   INS$FAOBUFDESC, #INS$C_FAOBUFLEN, -   ; 1280
                                                                                            INS$G_OUTRAB+34
                                                        00000000G   00      9F  00013   PUSHAB  INS$G_OUTRAB                         ; 1281
                                                00000000G   00              01  FB  00019   CALLS   #1, SYS$PUT
                                                            56              50  D0  00020   MOVL    R0, STATUS
                                                            67      0000G   8F  B0  00023   MOVW    #INS$C_FAOBUFLEN, INS$FAOBUFDESC    ; 1282
                                                            50  00000000G   00  D0  00028   MOVL    INS$FAOOUTBUF, R0                   ; 1283
                                                        04  A7              50  D0  0002F   MOVL    R0, INS$FAOBUFDESC+4
    0000G  8F               20                          6E              00  2C  00033   MOVC5   #0, (SP), #32, #INS$C_FAOBUFLEN, (R0)   ; 1284
                                                                            60      0003A
                                                            50              56  D0  0003B   MOVL    STATUS, R0                          ; 1285
                                                                            04  0003E   RET                                         ; 1286

; Routine Size:  63 bytes,    Routine Base:  $CODE$ + 0624
```

```
;  1100          1287  1 ROUTINE INS$FAO_AND_OUT(FAO_STRING, PARAMETER_LIST) =
;  1101          1288  2 BEGIN
;  1102          1289  2 !+++
;  1103          1290  2 !
;  1104          1291  2 !   FUNCTIONAL DESCRIPTION:
;  1105          1292  2 !        Format and output an ASCII string and stuff it into the output buffer.
;  1106          1293  2 !        Update the buffer pointers to reflect the new stuff in the
;  1107          1294  2 !        buffer.
;  1108          1295  2 !
;  1109          1296  2 !   INPUT:
;  1110          1297  2 !        fao_string =    Formatted Ascii Output control string for FAO
;  1111          1298  2 !        parameter_list= List of stuff to have formatted into buffer.
;  1112          1299  2 !
;  1113          1300  2 !   IMPLICIT INPUT:
;  1114          1301  2 !        Output buffer has been allocated and ins$faobufdesc is the
;  1115          1302  2 !        descriptor for it.
;  1116          1303  2 !
;  1117          1304  2 !   OUTPUT:
;  1118          1305  2 !        none
;  1119          1306  2 !
;  1120          1307  2 !   ROUTINE VALUE
;  1121          1308  2 !        Success, or error status from SYS$FAOL
;  1122          1309  2 !---
;  1123          1310  2 BUILTIN
;  1124          1311  2     AP,
;  1125          1312  2     CALLG;
;  1126          1313  2
;  1127          1314  2 EXECUTE(CALLG(.AP,INS$FAOL));
;  1128          1315  2 RETURN INS$OUTPUT_FAOBUF()
;  1129          1316  1 END;


                                        0000 00000 INS$FAO_AND_OUT:
                                                   .WORD   Save nothing                          ;  1287
                                8C   AF            6C  FA 00002         CALLG   (AP), INS$FAOL   ;  1314
                                     04            50  E9 00006         BLBC    STATUS, 1$       ;
                                B4   AF            00  FB 00009         CALLS   #0, INS$OUTPUT_FAOBUF ;  1315
                                                       04 0000D 1$:     RET                      ;  1316

; Routine Size:  14 bytes,    Routine Base:  $CODE$ + 0663

;  1130          1317  1 END
;  1131          1318  0 ELUDOM          ! INSGLOBAL


                                                   .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY
;
;       Name                      Bytes                         Attributes
;
```

```
;  $OWN$                    108  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                   744  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                  1649  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                              Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619        59         0      1000          00:01.7









;                                     COMMAND QUALIFIERS

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

; Size:         1649 code + 852 data bytes
; Run Time:        00:34.0
; Elapsed Time:    01:43.1
; Lines/CPU Min:    2328
; Lexemes/CPU-Min: 16604
; Memory Used:  197 pages
; Compilation Complete
```

INSMAIN
LIS

INSDELETE
LIS

INSOLDCMD
LIS

INSPRSOLD
LIS

INSGLOBAL
LIS

INSKFSCAN
LIS

INSLIST
LIS

INSMSG
LIS