```
DDDDDDDDDDD      UUU          UUU  MMM          MMM  PPPPPPPPPPPP
DDDDDDDDDDDDD    UUU          UUU  MMM          MMM  PPPPPPPPPPPPP
DDDDDDDDDDDDD    UUU          UUU  MMM          MMM  PPPPPPPPPPPPP
DDD          DDD UUU          UUU  MMMMMMM    MMMMMMM  PPP          PPP
DDD          DDD UUU          UUU  MMMMMMM    MMMMMMM  PPP          PPP
DDD          DDD UUU          UUU  MMMMMMM    MMMMMMM  PPP          PPP
DDD          DDD UUU          UUU  MMM  MMM    MMM  MMM  PPP          PPP
DDD          DDD UUU          UUU  MMM  MMM    MMM  MMM  PPP          PPP
DDD          DDD UUU          UUU  MMM  MMM    MMM  MMM  PPP          PPP
DDD          DDD UUU          UUU  MMM          MMM  MMM  PPPPPPPPPPPP
DDD          DDD UUU          UUU  MMM          MMM  MMM  PPPPPPPPPPPP
DDD          DDD UUU          UUU  MMM          MMM  MMM  PPPPPPPPPPPP
DDD          DDD UUU          UUU  MMM          MMM  MMM  PPP
DDD          DDD UUU          UUU  MMM          MMM  MMM  PPP
DDD          DDD UUU          UUU  MMM          MMM  MMM  PPP
DDD          DDD UUU          UUU  MMM          MMM  MMM  PPP
DDD          DDD UUU          UUU  MMM          MMM  MMM  PPP
DDD          DDD UUU          UUU  MMM          MMM  MMM  PPP
DDDDDDDDDDDD     UUUUUUUUUUUUUUUU  MMM          MMM  MMM  PPP
DDDDDDDDDDDD     UUUUUUUUUUUUUUUU  MMM          MMM  MMM  PPP
DDDDDDDDDDDD     UUUUUUUUUUUUUUUU  MMM          MMM  MMM  PPP
```

FILEID**DUMPFILE

```
DDDDDDDD    UU      UU  MM      MM  PPPPPPPP    FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE
DDDDDDDD    UU      UU  MM      MM  PPPPPPPP    FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE
DD      DD  UU      UU  MMMM  MMMM  PP      PP  FF              II      LL          EE
DD      DD  UU      UU  MMMM  MMMM  PP      PP  FF              II      LL          EE
DD      DD  UU      UU  MM  MM  MM  PP      PP  FF              II      LL          EE
DD      DD  UU      UU  MM  MM  MM  PP      PP  FF              II      LL          EE
DD      DD  UU      UU  MM      MM  PPPPPPPP    FFFFFFFF        II      LL          EEEEEEEE
DD      DD  UU      UU  MM      MM  PPPPPPPP    FFFFFFFF        II      LL          EEEEEEEE
DD      DD  UU      UU  MM      MM  PP          FF              II      LL          EE
DD      DD  UU      UU  MM      MM  PP          FF              II      LL          EE
DD      DD  UU      UU  MM      MM  PP          FF              II      LL          EE          ....
DD      DD  UU      UU  MM      MM  PP          FF              II      LL          EE          ....
DDDDDDDD    UUUUUUUUUU  MM      MM  PP          FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  ....
DDDDDDDD    UUUUUUUUUU  MM      MM  PP          FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  ....

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

```
;     1          0001  0 MODULE DUMP$FILE (
;     2          0002  0         IDENT='V04-000',
;     3          0003  0         ADDRESSING_MODE(EXTERNAL=GENERAL,
;     4          0004  0                 NONEXTERNAL=LONG_RELATIVE)
;     5          0005  0         ) =
;     6          0006  1 BEGIN
;     7          0007  1 !
;     8          0008  1 !
;     9          0009  1 !*****************************************************************************
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    12          0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    13          0013  1 !*  ALL RIGHTS RESERVED.                                                      *
;    14          0014  1 !*                                                                           *
;    15          0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    16          0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE     *
;    17          0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER      *
;    18          0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    19          0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    20          0020  1 !*  TRANSFERRED.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    23          0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    24          0024  1 !*  CORPORATION.                                                              *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    27          0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    28          0028  1 !*                                                                           *
;    29          0029  1 !*                                                                           *
;    30          0030  1 !*****************************************************************************
;    31          0031  1 !
;    32          0032  1 !
;    33          0033  1 !++
;    34          0034  1 !
;    35          0035  1 ! FACILITY:  File dump utility
;    36          0036  1 !
;    37          0037  1 ! ABSTRACT:
;    38          0038  1 !       This module contains the routines to do the work of dumping files.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !       VAX native, user mode.
;    42          0042  1 !
;    43          0043  1 ! AUTHOR:  Benn Schreiber, Stephen Zalewski      CREATION DATE:  22-Jun-1981
;    44          0044  1 !
;    45          0045  1 ! MODIFIED BY:
;    46          0046  1 !
;    47          0047  1 !       V03-001 MLJ0033         Martin L. Jack, 23-Aug-1981  9:48
;    48          0048  1 !                               Extensive rewriting to finish implementation.
;    49          0049  1 !
;    50          0050  1 !--
```

```
;    52          0051  1 LIBRARY 'SYS$LIBRARY:STARLET';
;    53          0052  1 REQUIRE 'SRC$:DUMPRE';
;    54          0168  1
;    55          0169  1
;    56          0170  1 FORWARD ROUTINE
;    57          0171  1     dump$fao_setup:     NOVALUE,      ! Set up FAO control strings
;    58          0172  1     dump$new_page:      NOVALUE,      ! Output new page
;    59          0173  1     dump$put_header:    NOVALUE,      ! Output heading lines
;    60          0174  1     dump$output_getmsg: NOVALUE,      ! Get message and output
;    61          0175  1     dump$blank_line:    NOVALUE,      ! Write blank line to listing file
;    62          0176  1     dump$put_line:      NOVALUE,      ! Output record, watching lines
;    63          0177  1     dump$dump_buffer:   NOVALUE;      ! Dump one record/block
;    64          0178  1
;    65          0179  1 EXTERNAL ROUTINE
;    66          0180  1     dump$fao_line:      NOVALUE,      ! Format one line
;    67          0181  1     dump$header:        NOVALUE,      ! Dump file header(s)
;    68          0182  1     dump$one_header,                  ! Dump block as a file header
;    69          0183  1     dump$read,                        ! Read from file
;    70          0184  1     dump$write:         NOVALUE,      ! Write to output file
;    71          0185  1     SYS$FAO;                          ! Formatted ASCII output routine
;    72          0186  1
;    73          0187  1 EXTERNAL
;    74          0188  1     dump$gl_flags : BBLOCK,           ! General flags
;    75          0189  1     dump$gl_outdesc : BBLOCK,         ! Output buffer descriptor
;    76          0190  1     dump$gl_idesc : BBLOCK,           ! Descriptor for input filename
;    77          0191  1     dump$gl_file_efblk,               ! End of file block
;    78          0192  1     dump$gl_file_hiblk,               ! Highest allocated block
;    79          0193  1     dump$gl_lpp,                      ! Number of lines per page
;    80          0194  1     dump$gl_ifab : REF BBLOCK,        ! Input FAB
;    81          0195  1     dump$gl_inam : REF BBLOCK,        ! Input NAM block
;    82          0196  1     dump$gl_cur_block,                ! Current record/block
;    83          0197  1     dump$gl_max_block,                ! Maximum record/block to dump
;    84          0198  1     dump$gl_width,                    ! Width of listing line
;    85          0199  1     dump$gl_number,                   ! Starting dump index number
;    86          0200  1     dump$gl_record,                   ! Record or block number
;    87          0201  1     dump$gq_time;                     ! Time at beginning of dump
;    88          0202  1
;    89          0203  1 EXTERNAL LITERAL
;    90          0204  1     dump$_facility,
;    91          0205  1     dump$_dumpofil,
;    92          0206  1     dump$_dumpodev,
;    93          0207  1     dump$_bn,
;    94          0208  1     dump$_lbn,
;    95          0209  1     dump$_vbn,
;    96          0210  1     dump$_fildnt,
;    97          0211  1     dump$_recno,
;    98          0212  1     dump$_header;
;    99          0213  1
;   100          0214  1 LITERAL
;   101          0215  1     max_fao_size = 40;                ! Size of largest of faotables' expanded fao strings
;   102          0216  1
;   103          0217  1 OWN
;   104          0218  1     modeindex,                        ! Index into faotable
;   105          0219  1     dumpmode,                         ! mode for dump$fao_line
;   106          0220  1     entrysize,                        ! Size of one entry
;   107          0221  1     entsperline,                      ! Number of entries on one line
;   108          0222  1     linesthispage,                    ! Number of lines on this page
```

```
;  109          0223  1     dumpwidth,                          ! Width of one full dump listing line
;  110          0224  1     plinfaostring : BBLOCK[max_fao_size], ! FAO string for partial lines
;  111          0225  1     plinfaodesc : BBLOCK[dsc$c_s_bln]   ! Descriptor for partial line fao control string
;  112          0226  1                 INITIAL(max_fao_size,
;  113          0227  1                         plinfaostring),
;  114          0228  1     faoctrstring : BBLOCK[max_fao_size],! FAO control string
;  115          0229  1     faoctrdesc : BBLOCK[dsc$c_s_bln]    ! FAO control string descriptor
;  116          0230  1                 INITIAL(max_fao_size,
;  117          0231  1                         faoctrstring);
;  118          0232  1
;  119          0233  1 BIND
;  120          0234  1     sizetbl = UPLIT(BYTE(9,5,3,11,6,4,12,7,4)) : VECTOR[,BYTE],
;  121          0235  1     charsperbyte = UPLIT(BYTE(2,3,3)) : VECTOR[,BYTE], ! Number of ascii chars /byte based on radix
;  122          0236  1
;  123          0237  1     offtable = UPLIT(cstring('6XL'),    ! FAO control to print buffer offsets
;  124          0238  1                      cstring('6SL'),
;  125          0239  1                      cstring('6OL')) : VECTOR[,LONG],
;  126          0240  1
;  127          0241  1     faotable = UPLIT(cstring('9XL'),
;  128          0242  1                      cstring('5XW'),
;  129          0243  1                      cstring('3XB'),
;  130          0244  1                      cstring('11SL'),
;  131          0245  1                      cstring('6SW'),
;  132          0246  1                      cstring('4SB'),
;  133          0247  1                      cstring('12OL'),
;  134          0248  1                      cstring('7OW'),
;  135          0249  1                      cstring('4OB')) : VECTOR[,LONG],
;  136          0250  1
;  137          0251  1     sizetable = UPLIT(BYTE(                 ! Table to round entry's per
;  138          0252  1                       1,                    ! line to nearest lower power
;  139          0253  1                       2,                    ! of two.  Max length is 32.
;  140          0254  1                       4,
;  141          0255  1                       8,
;  142          0256  1                       16,
;  143          0257  1                       32,
;  144          0258  1                       64,
;  145          0259  1                       128)):VECTOR[,BYTE];
```

```
;  147     0260  1 GLOBAL ROUTINE dump$dump_file: NOVALUE=
;  148     0261  2 BEGIN
;  149     0262  2 !
;  150     0263  2 ! This routine is the driver for file dumping.
;  151     0264  2 !
;  152     0265  2 LOCAL
;  153     0266  2     status : BLOCK[4,BYTE],
;  154     0267  2     bufdesc : BBLOCK[dsc$c_s_bln],
;  155     0268  2     subdesc : BBLOCK[dsc$c_s_bln];              ! This desc is used to point to
;  156     0269  2                                                 ! each individual BLOCK/RECORD in bufdesc
;  157     0270  2
;  158     0271  2
;  159     0272  2 dump$fao_setup();                               ! Set up fao control string
;  160     0273  2 linesthispage = .dump$gl_lpp;                   ! Force new page
;  161     0274  2
;  162     0275  2
;  163     0276  2 IF .dump$gl_flags[dump$v_header]                ! If /HEADER specified
;  164     0277  2 THEN
;  165     0278  3     BEGIN
;  166     0279  3     dump$header();                              ! Dump the header
;  167     0280  3     linesthispage = .dump$gl_lpp;               ! Force new page
;  168     0281  2     END;
;  169     0282  2
;  170     0283  2
;  171     0284  2 ! Read and dump the file.
;  172     0285  2 !
;  173     0286  2 WHILE true DO
;  174     0287  3     BEGIN
;  175     0288  3     dump$gl_record = .dump$gl_record + 1;       ! Increment unit counter
;  176     0289  4     IF (NOT .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_rnd] ! If not disk
;  177     0290  4     OR .dump$gl_flags[dump$v_records])          ! or record mode
;  178     0291  3     AND .dump$gl_record GTRU .dump$gl_max_block ! Stop if already dumped last
;  179     0292  3         THEN RETURN;                            ! needed block
;  180     0293  3     status = dump$read(bufdesc);
;  181     0294  3     IF .status EQL ss$_endoffile THEN EXITLOOP;
;  182     0295  3     IF NOT .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_rnd]  ! If not disk
;  183     0296  3     OR .dump$gl_flags[dump$v_records]                   ! or record mode
;  184     0297  3     THEN
;  185     0298  4         BEGIN
;  186     0299  4         IF .dump$gl_record GEQU .dump$gl_cur_block      ! In range to be dumped
;  187     0300  4         THEN
;  188     0301  5             BEGIN
;  189     0302  5             IF NOT .dump$gl_flags[dump$v_records]
;  190     0303  5                 THEN linesthispage = .dump$gl_lpp;      ! Force new page
;  191     0304  5             dump$dump_buffer(bufdesc, .status, false);  ! Dump entire block
;  192     0305  5             END
;  193     0306  4         END
;  194     0307  3     ELSE
;  195     0308  3         WHILE .bufdesc[dsc$w_length] GTRU 0 DO          ! Dump all blocks
;  196     0309  4             BEGIN
;  197     0310  4             subdesc[dsc$w_length] = MINU(512, .bufdesc[dsc$w_length]);
;  198     0311  4             subdesc[dsc$a_pointer] = .bufdesc[dsc$a_pointer];
;  199     0312  4             linesthispage = .dump$gl_lpp;
;  200     0313  4             dump$dump_buffer(subdesc, .status, false);          ! Dump a block
;  201     0314  4             status = ss$_normal;
;  202     0315  4             bufdesc[dsc$w_length] = .bufdesc[dsc$w_length] -
;  203     0316  4                 .subdesc[dsc$w_length];
```

```
;  204          0317  4              bufdesc[dsc$a_pointer] = .bufdesc[dsc$a_pointer] +
;  205          0318  4                  .subdesc[dsc$w_length];
;  206          0319  3              END;
;  207          0320  2      END;
;  208          0321  1  END;


                                                                .TITLE  DUMP$FILE
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                    04  07  0C  04  06  0B  03  05  09  00000 P.AAA:  .BYTE   9, 5, 3, 11, 6, 4, 12, 7, 4          ;
                                                    00009           .BLKB   3
                                        03  03  02  0000C P.AAB:  .BYTE   2, 3, 3                                 ;
                                    4C  58  36  03  0000F P.AAD:  .ASCII  <3>\6XL\                                ;
                                    4C  53  36  03  00013 P.AAE:  .ASCII  <3>\6SL\                                ;
                                    4C  4F  36  03  00017 P.AAF:  .ASCII  <3>\6OL\                                ;
                                                    0001B           .BLKB   1
                          00000000' 00000000' 00000000' 0001C P.AAC:  .ADDRESS P.AAD, P.AAE, P.AAF                 ;
                                    4C  58  39  03  00028 P.AAH:  .ASCII  <3>\9XL\                                ;
                                    57  58  35  03  0002C P.AAI:  .ASCII  <3>\5XW\                                ;
                                    42  58  33  03  00030 P.AAJ:  .ASCII  <3>\3XB\                                ;
                                4C  53  31  31  04  00034 P.AAK:  .ASCII  <4>\11SL\                               ;
                                    57  53  36  03  00039 P.AAL:  .ASCII  <3>\6SW\                                ;
                                    42  53  34  03  0003D P.AAM:  .ASCII  <3>\4SB\                                ;
                                4C  4F  32  31  04  00041 P.AAN:  .ASCII  <4>\12OL\                               ;
                                    57  4F  37  03  00046 P.AAO:  .ASCII  <3>\7OW\                                ;
                                    42  4F  34  03  0004A P.AAP:  .ASCII  <3>\4OB\                                ;
                                                    0004E           .BLKB   2
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00050 P.AAG:  .ADDRESS P.AAH, P.AAI, P.AAJ, P.AAK, P.AAL, -  ;
                          00000000' 00000000' 00000000' 00068                   P.AAM, P.AAN, P.AAO, P.AAP              ;
                      80  40  20  10  08  04  02  01  00074 P.AAQ:  .BYTE   1, 2, 4, 8, 16, 32, 64, -128             ;

                                                                .PSECT  $OWN$,NOEXE,2

                                                    00000 MODEINDEX:
                                                                .BLKB   4
                                                    00004 DUMPMODE:
                                                                .BLKB   4
                                                    00008 ENTRYSIZE:
                                                                .BLKB   4
                                                    0000C ENTSPERLINE:
                                                                .BLKB   4
                                                    00010 LINESTHISPAGE:
                                                                .BLKB   4
                                                    00014 DUMPWIDTH:
                                                                .BLKB   4
                                                    00018 PLINFAOSTRING:
                                                                .BLKB   40
                                          00000028  00040 PLINFAODESC:
                                                                .LONG   40
                                          00000000' 00044       .ADDRESS PLINFAOSTRING                            ;
                                                    00048 FAOCTRSTRING:
                                                                .BLKB   40
                                          00000028  00070 FAOCTRDESC:
                                                                .LONG   40                                        ;
```

```
                                00000000' 00074          .ADDRESS FAOCTRSTRING                                          ;

                                                SIZETBL=            P.AAA
                                                CHARSPERBYTE=       P.AAB
                                                OFFTABLE=           P.AAC
                                                FAOTABLE=           P.AAG
                                                SIZETABLE=          P.AAQ
                                                        .EXTRN  DUMP$FAO_LINE, DUMP$HEADER
                                                        .EXTRN  DUMP$ONE_HEADER
                                                        .EXTRN  DUMP$READ, DUMP$WRITE
                                                        .EXTRN  SYS$FAO, DUMP$GL_FLAGS
                                                        .EXTRN  DUMP$GL_OUTDESC
                                                        .EXTRN  DUMP$GL_IDESC, DUMP$GL_FILE_EFBLK
                                                        .EXTRN  DUMP$GL_FILE_HIBLK
                                                        .EXTRN  DUMP$GL_LPP, DUMP$GL_IFAB
                                                        .EXTRN  DUMP$GL_INAM, DUMP$GL_CUR_BLOCK
                                                        .EXTRN  DUMP$GL_MAX_BLOCK
                                                        .EXTRN  DUMP$GL_WIDTH, DUMP$GL_NUMBER
                                                        .EXTRN  DUMP$GL_RECORD, DUMP$GL_TIME
                                                        .EXTRN  DUMP$_FACILITY, DUMP$_DUMPOFIL
                                                        .EXTRN  DUMP$_DUMPODEV, DUMP$_BN
                                                        .EXTRN  DUMP$_LBN, DUMP$_VBN
                                                        .EXTRN  DUMP$_FILDNT, DUMP$_RECNO
                                                        .EXTRN  DUMP$_HEADER

                                                        .PSECT  $CODE$,NOWRT,2

                                      01FC 00000        .ENTRY  DUMP$DUMP_FILE, Save R2,R3,R4,R5,R6,R7,R8      ; 0260
                                58 00000000V  EF  9E 00002        MOVAB   DUMP$DUMP_BUFFER, R8
                                57 00000000G  00  9E 00009        MOVAB   DUMP$GL_IFAB, R7
                                56 00000000G  00  9E 00010        MOVAB   DUMP$GL_RECORD, R6
                                55 00000000G  00  9E 00017        MOVAB   DUMP$GL_FLAGS, R5
                                54 00000000'  EF  9E 0001E        MOVAB   LINESTHISPAGE, R4
                                53 00000000G  00  9E 00025        MOVAB   DUMP$GL_LPP, R3
                                5E            10  C2 0002C        SUBL2   #16, SP
                    00000000V   EF            00  FB 0002F        CALLS   #0, DUMP$FAO_SETUP                     ; 0272
                                64            63  D0 00036        MOVL    DUMP$GL_LPP, LINESTHISPAGE             ; 0273
                 0A             65            06  E1 00039        BBC     #6, DUMP$GL_FLAGS, 1$                  ; 0276
                    00000000G   00            00  FB 0003D        CALLS   #0, DUMP$HEADER                        ; 0279
                                64            63  D0 00044        MOVL    DUMP$GL_LPP, LINESTHISPAGE             ; 0280
                                              66  D6 00047  1$:    INCL    DUMP$GL_RECORD                         ; 0288
                                50            67  D0 00049        MOVL    DUMP$GL_IFAB, R0                       ; 0289
                 05         43  A0            04  E1 0004C        BBC     #4, 67(R0), 2$
                 09         01  A5            05  E1 00051        BBC     #5, DUMP$GL_FLAGS+1, 3$                ; 0290
                    00000000G   00            66  D1 00056  2$:    CMPL    DUMP$GL_RECORD, DUMP$GL_MAX_BLOCK      ; 0291
                                              7A  1A 0005D        BGTRU   8$
                                      08      AE  9F 0005F  3$:    PUSHAB  BUFDESC                                ; 0293
                    00000000G   00            01  FB 00062        CALLS   #1, DUMP$READ
                                52            50  D0 00069        MOVL    R0, STATUS
                    00000870    8F            52  D1 0006C        CMPL    STATUS, #2160                          ; 0294
                                              64  13 00073        BEQL    8$
                                50            67  D0 00075        MOVL    DUMP$GL_IFAB, R0                       ; 0295
                 05         43  A0            04  E1 00078        BBC     #4, 67(R0), 4$
                 1D         01  A5            05  E1 0007D        BBC     #5, DUMP$GL_FLAGS+1, 6$                ; 0296
                    00000000G   00            66  D1 00082  4$:    CMPL    DUMP$GL_RECORD, DUMP$GL_CUR_BLOCK      ; 0299
                                              BC  1F 00089        BLSSU   1$
                 03         01  A5            05  E0 0008B        BBS     #5, DUMP$GL_FLAGS+1, 5$                ; 0302
```

```
                                64              63  D0 00090          MOVL    DUMP$GL_LPP, LINESTHISPAGE          ; 0303
                                                7E  D4 00093 5$:      CLRL    -(SP)                               ; 0304
                                                52  DD 00095          PUSHL   STATUS                              ;
                                    10          AE  9F 00097          PUSHAB  BUFDESC                             ;
                                68              03  FB 0009A          CALLS   #3, DUMP$DUMP_BUFFER                ;
                                                A8  11 0009D          BRB     1$                                  ; 0298
                                    08          AE  B5 0009F 6$:      TSTW    BUFDESC                             ; 0308
                                                A3  13 000A2          BEQL    1$                                  ;
                                50      08      AE  3C 000A4          MOVZWL  BUFDESC, R0                         ; 0310
                          0200  8F              50  B1 000A8          CMPW    R0, #512                            ;
                                                05  1B 000AD          BLEQU   7$                                  ;
                                50      0200    8F  3C 000AF          MOVZWL  #512, R0                            ;
                                6E              50  B0 000B4 7$:      MOVW    R0, SUBDESC                         ;
                            04  AE      0C      AE  D0 000B7          MOVL    BUFDESC+4, SUBDESC+4                ; 0311
                                64              63  D0 000BC          MOVL    DUMP$GL_LPP, LINESTHISPAGE          ; 0312
                                                7E  D4 000BF          CLRL    -(SP)                               ; 0313
                                                52  DD 000C1          PUSHL   STATUS                              ;
                                    08          AE  9F 000C3          PUSHAB  SUBDESC                             ;
                                68              03  FB 000C6          CALLS   #3, DUMP$DUMP_BUFFER                ;
                                52              01  D0 000C9          MOVL    #1, STATUS                          ; 0314
                            08  AE              6E  A2 000CC          SUBW2   SUBDESC, BUFDESC                    ; 0316
                                50              6E  3C 000D0          MOVZWL  SUBDESC, R0                         ; 0318
                            0C  AE              50  C0 000D3          ADDL2   R0, BUFDESC+4                       ;
                                                C6  11 000D7          BRB     6$                                  ; 0308
                                                    04 000D9 8$:      RET                                         ; 0321

; Routine Size:  218 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  210     0322  1 ROUTINE dump$fao_setup: NOVALUE=
;  211     0323  2 BEGIN
;  212     0324  2 !
;  213     0325  2 ! This routine sets up the FAO control string.  It also
;  214     0326  2 ! calculates the mode and entry widths.
;  215     0327  2 !
;  216     0328  2 LOCAL
;  217     0329  2     entry;
;  218     0330  2
;  219     0331  2 entry = 0;                                          ! used for entry size calc.
;  220     0332  2 dumpmode = 0;
;  221     0333  2 entrysize = 4;                                      ! Assume longword...
;  222     0334  2
;  223     0335  2 IF .dump$gl_flags[dump$v_decimal]
;  224     0336  2 THEN
;  225     0337  2     modeindex = 3
;  226     0338  2 ELSE IF .dump$gl_flags[dump$v_octal]
;  227     0339  2 THEN
;  228     0340  2     modeindex = 6
;  229     0341  2 ELSE
;  230     0342  2     modeindex = 0;                                  ! Default to hex dump
;  231     0343  2
;  232     0344  2
;  233     0345  2 IF .dump$gl_flags[dump$v_word]
;  234     0346  2 THEN
;  235     0347  3     BEGIN
;  236     0348  3     entrysize = 2;
;  237     0349  3     dumpmode = 1;
;  238     0350  3     modeindex = .modeindex + 1;
;  239     0351  3     END
;  240     0352  2 ELSE IF .dump$gl_flags[dump$v_byte]
;  241     0353  2 THEN
;  242     0354  3     BEGIN
;  243     0355  3     entrysize = 1;
;  244     0356  3     dumpmode = 2;
;  245     0357  3     modeindex = .modeindex + 2;
;  246     0358  2     END;
;  247     0359  2
;  248     0360  2
;  249     0361  2 ! Find entries per line and make it the nearest lower power of 2.
;  250     0362  2 !
;  251     0363  2 entsperline = ((.dump$gl_width - 5)/(.sizetbl[.modeindex]+.entrysize)) AND NOT 1;
;  252     0364  2
;  253     0365  2 IF .entsperline GTR 64                              ! Make sure entsperline is reasonable
;  254     0366  2 THEN SIGNAL_STOP(dump$_facility^16 + shr$_badlogic + sts$k_severe);
;  255     0367  2
;  256     0368  2 WHILE .entsperline GEQ .sizetable[.entry]           ! Find nearest larger power of 2
;  257     0369  2     DO entry = .entry + 1;                          ! from entsperline.
;  258     0370  2
;  259     0371  2 entsperline = .sizetable[.entry-1];                 ! Make entsperline nearest lower
;  260     0372  2                                                     ! power of two.
;  261     0373  2 dumpwidth = .entsperline*(.sizetbl[.modeindex] + .entrysize) + 8;
;  262     0374  2
;  263     0375  2 faoctrdesc[dsc$w_length] = max_fao_size;
;  264     0376  2 SYS$FAO(
;  265     0377  2     $descriptor('!!!ZL(!AC) !!!ZLAF !!!AC'),
;  266     0378  2     faoctrdesc,
```

```
;  267          0379  2      faoctrdesc,
;  268          0380  2      .entsperline,
;  269          0381  2      .faotable[.modeindex],
;  270          0382  2      .entsperline * .entrysize,
;  271          0383  2      .offtable[.modeindex/3]);
;  272          0384  2
;  273          0385  2
;  274          0386  2 ! Set up FAO control string to be used for partial lines.
;  275          0387  2 !
;  276          0388  2 SYS$FAO(
;  277          0389  2      $descriptor('!!!!!!ZL(!AC) !!!!!ZLAF !!!!!AC'),
;  278          0390  2      plinfaodesc,
;  279          0391  2      plinfaodesc,
;  280          0392  2      .faotable[.modeindex],
;  281          0393  2      .entsperline * .entrysize,
;  282          0394  2      .offtable[.modeindex/3]);
;  283          0395  1 END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

5A 21 21 21 20 29 43 41 21 28 4C 5A 21 21 21  0007C P.AAS:  .ASCII  \!!!ZL(!AC) !!!ZLAF !!!AC\                ;
                  43 41 21 21 21 20 46 41 4C  0008B                                                          ;
                                              00000018  00094 P.AAR:  .LONG   24                             ;
                                              00000000' 00098         .ADDRESS P.AAS                         ;
21 20 29 43 41 21 28 4C 5A 21 21 21 21 21 21  0009C P.AAU:  .ASCII  \!!!!!!ZL(!AC) !!!!!ZLAF !!!!!AC\        ;
41 21 21 21 21 21 20 46 41 4C 5A 21 21 21 21  000AB                                                        ;
                                           43 000BA                                                        ;
                                              000BB         .BLKB   1                                      ;
                                              0000001F  000BC P.AAT:  .LONG   31                             ;
                                              00000000' 000C0         .ADDRESS P.AAU                         ;


                                                        .PSECT  $CODE$,NOWRT,2

                                                 007C 00000 DUMP$FAO_SETUP:
                                                            .WORD   Save R2,R3,R4,R5,R6                     ; 0322
                     56 00000000G   00   9E 00002           MOVAB   SYS$FAO, R6                             ;
                     55 00000000G   00   9E 00009           MOVAB   DUMP$GL_FLAGS, R5                       ;
                     54 00000000'   EF   9E 00010           MOVAB   SIZETBL, R4                             ;
                     53 00000000'   EF   9E 00017           MOVAB   MODEINDEX, R3                           ;
                                    52   D4 0001E           CLRL    ENTRY                                   ; 0331
                           04       A3   D4 00020           CLRL    DUMPMODE                                ; 0332
                08   A3               04   D0 00023           MOVL    #4, ENTRYSIZE                           ; 0333
05                  65               03   E1 00027           BBC     #3, DUMP$GL_FLAGS, 1$                   ; 0335
                    63               03   D0 0002B           MOVL    #3, MODEINDEX                           ; 0337
                                     0C   11 0002E           BRB     3$                                      ;
05          01      A5               02   E1 00030 1$:       BBC     #2, DUMP$GL_FLAGS+1, 2$                 ; 0338
                    63               06   D0 00035           MOVL    #6, MODEINDEX                           ; 0340
                                     02   11 00038           BRB     3$                                      ;
                                     63   D4 0003A 2$:       CLRL    MODEINDEX                               ; 0342
0C          01      A5               06   E1 0003C 3$:       BBC     #6, DUMP$GL_FLAGS+1, 4$                 ; 0345
            08      A3               02   D0 00041           MOVL    #2, ENTRYSIZE                           ; 0348
            04      A3               01   D0 00045           MOVL    #1, DUMPMODE                            ; 0349
                                     63   D6 00049           INCL    MODEINDEX                               ; 0350
```

```
                                        0F   11 0004B        BRB     5$                                              ; 0345
                  0B          65        02   E1 0004D 4$:    BBC     #2, DUMP$GL_FLAGS, 5$                           ; 0352
                        08    A3        01   D0 00051        MOVL    #1, ENTRYSIZE                                   ; 0355
                        04    A3        02   D0 00055        MOVL    #2, DUMPMODE                                    ; 0356
                              63        02   C0 00059        ADDL2   #2, MODEINDEX                                   ; 0357
                  51 00000000G  00      05   C3 0005C 5$:    SUBL3   #5, DUMP$GL_WIDTH, R1                           ; 0363
                              50        64   9E 00064        MOVAB   SIZETBL, R0                                     ;
                              50   00 B340   9A 00067        MOVZBL  @MODEINDEX[R0], R0                              ;
                              50   08   A3   C0 0006C        ADDL2   ENTRYSIZE, R0                                   ;
                              51        50   C6 00070        DIVL2   R0, R1                                          ;
               0C    A3       51        01   CB 00073        BICL3   #1, R1, ENTSPERLINE                             ;
                  00000040    8F   0C   A3   D1 00078        CMPL    ENTSPERLINE, #64                                ; 0365
                                        0D   15 00080        BLEQ    6$                                              ;
                             00000000*  8F   DD 00082        PUSHL   #<<<DUMP$_FACILITY@16>+4384>+4>                 ; 0366
                  00000000G   00        01   FB 00088        CALLS   #1, LIB$STOP                                    ;
       0C    A3       74 A442 08        00   ED 0008F 6$:    CMPZV   #0, #8, SIZETABLE[ENTRY], ENTSPERLINE           ; 0368
                                        04   14 00097        BGTR    7$                                              ;
                                        52   D6 00099        INCL    ENTRY                                           ; 0369
                                        F2   11 0009B        BRB     6$                                              ;
                        0C    A3   73 A442   9A 0009D 7$:    MOVZBL  SIZETABLE-1[ENTRY], ENTSPERLINE             ; 0371
                              52   0C   A3   D0 000A3        MOVL    ENTSPERLINE, R2                                 ; 0373
                              51        63   D0 000A7        MOVL    MODEINDEX, R1                                   ;
                              50      6441   9A 000AA        MOVZBL  SIZETBL[R1], R0                                 ;
                              50   08   A3   C0 000AE        ADDL2   ENTRYSIZE, R0                                   ;
                              50        52   C4 000B2        MULL2   R2, R0                                          ;
                        14    A3   08   A0   9E 000B5        MOVAB   8(R0), DUMPWIDTH                                ;
                        70    A3        28   B0 000BA        MOVW    #40, FAOCTRDESC                                 ; 0375
                  50          51        03   C7 000BE        DIVL3   #3, R1, R0                                      ; 0383
                                   1C A440   DD 000C2        PUSHL   OFFTABLE[R0]                                    ;
                  7E          52   08   A3   C5 000C6        MULL3   ENTRYSIZE, R2, -(SP)                            ; 0382
                                   50 A441   DD 000CB        PUSHL   FAOTABLE[R1]                                    ; 0381
                                        52   DD 000CF        PUSHL   R2                                              ; 0380
                                   70   A3   9F 000D1        PUSHAB  FAOCTRDESC                                      ; 0376
                                   70   A3   9F 000D4        PUSHAB  FAOCTRDESC                                      ;
                                 0094   C4   9F 000D7        PUSHAB  P.AAR                                           ; 0377
                              66        07   FB 000DB        CALLS   #7, SYS$FAO                                     ;
                              51        63   D0 000DE        MOVL    MODEINDEX, R1                                   ; 0394
                  50          51        03   C7 000E1        DIVL3   #3, R1, R0                                      ;
                                   1C A440   DD 000E5        PUSHL   OFFTABLE[R0]                                    ;
                  7E    0C    A3   08   A3   C5 000E9        MULL3   ENTRYSIZE, ENTSPERLINE, -(SP)                   ; 0393
                                   50 A441   DD 000EF        PUSHL   FAOTABLE[R1]                                    ; 0392
                                   40   A3   9F 000F3        PUSHAB  PLINFAODESC                                     ; 0388
                                   40   A3   9F 000F6        PUSHAB  PLINFAODESC                                     ;
                                 00BC   C4   9F 000F9        PUSHAB  P.AAT                                           ; 0389
                              66        06   FB 000FD        CALLS   #6, SYS$FAO                                     ;
                                        04 00100             RET                                                     ; 0395

; Routine Size:  257 bytes,    Routine Base:  $CODE$ + 00DA
```

```
;  285          0396  1 ROUTINE dump$put_header(bufdesc,header): NOVALUE=
;  286          0397  2 BEGIN
;  287          0398  2 MAP
;  288          0399  2     bufdesc : REF BBLOCK;
;  289          0400  2
;  290          0401  2
;  291          0402  2 IF
;  292          0403  3     BEGIN
;  293          0404  3     IF .dump$gl_flags[dump$v_records]
;  294          0405  3     THEN
;  295          0406  3         .linesthispage + 4 GEQ .dump$gl_lpp
;  296          0407  3     ELSE
;  297          0408  3         true
;  298          0409  3     END
;  299          0410  2 THEN
;  300          0411  2     dump$new_page()
;  301          0412  2 ELSE
;  302          0413  2     dump$blank_line();
;  303          0414  2
;  304          0415  2
;  305          0416  2 IF NOT .header                                  ! Not dumping header
;  306          0417  2 THEN
;  307          0418  2     dump$output_getmsg(
;  308          0419  3         (IF .dump$gl_flags[dump$v_records]
;  309          0420  3             THEN dump$_recno
;  310          0421  3         ELSE IF NOT .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_rnd]
;  311          0422  3             THEN dump$_bn
;  312          0423  3         ELSE IF .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_for]
;  313          0424  3             THEN dump$_lbn
;  314          0425  2             ELSE dump$_vbn),
;  315          0426  2         %B'0001',
;  316          0427  2         .dump$gl_record,
;  317          0428  2         .bufdesc[dsc$w_length])
;  318          0429  2 ELSE
;  319          0430  2     dump$output_getmsg(dump$_header, %B'0001');
;  320          0431  2
;  321          0432  2 dump$blank_line();
;  322          0433  1 END;


                                000C 00000 DUMP$PUT_HEADER:
                                           .WORD   Save R2,R3                                  ; 0396
                  53 00000000V  EF   9E 00002          MOVAB   DUMP$OUTPUT_GETMSG, R3        ;
                  52 00000000V  EF   9E 00009          MOVAB   DUMP$BLANK_LINE, R2           ;
           11 00000000G   00    05   E1 00010          BBC     #5, DUMP$GL_FLAGS+1, 1$       ; 0404
           50 00000000'   EF    04   C1 00018          ADDL3   #4, LINESTHISPAGE, R0         ; 0406
              00000000G   00    50   D1 00020          CMPL    R0, DUMP$GL_LPP               ;
                                09   19 00027          BLSS    2$                            ;
              00000000V   EF    00   FB 00029 1$:      CALLS   #0, DUMP$NEW_PAGE             ; 0411
                                03   11 00030          BRB     3$                            ;
                  62            00   FB 00032 2$:      CALLS   #0, DUMP$BLANK_LINE           ; 0413
                  4C        08  AC   E8 00035 3$:      BLBS    HEADER, 9$                    ; 0416
                  7E        04  BC   3C 00039          MOVZWL  @BUFDESC, -(SP)               ; 0428
                     00000000G  00   DD 0003D          PUSHL   DUMP$GL_RECORD                ; 0427
```

```
                                                 01  DD 00043          PUSHL   #1                                      ; 0418
                        08 00000000G   00        05  E1 00045          BBC     #5, DUMP$GL_FLAGS+1, 4$                 ; 0419
                                     00000000G   8F  DD 0004D          PUSHL   #DUMP$_RECNO                            ;
                                                 2B  11 00053          BRB     8$                                      ;
                                   50 00000000G  00  D0 00055 4$:      MOVL    DUMP$GL_IFAB, R0                        ; 0421
                        09      43   A0          04  E0 0005C          BBS     #4, 67(R0), 5$                          ;
                                   50 00000000G  8F  D0 00061          MOVL    #DUMP$_BN, R0                           ;
                                                 14  11 00068          BRB     7$                                      ;
                                   09      43    A0  E9 0006A 5$:      BLBC    67(R0), 6$                              ; 0423
                                   50 00000000G  8F  D0 0006E          MOVL    #DUMP$_LBN, R0                          ;
                                                 07  11 00075          BRB     7$                                      ;
                                   50 00000000G  8F  D0 00077 6$:      MOVL    #DUMP$_VBN, R0                          ;
                                                 50  DD 0007E 7$:      PUSHL   R0                                      ; 0421
                                   63            04  FB 00080 8$:      CALLS   #4, DUMP$OUTPUT_GETMSG                  ; 0419
                                                 0B  11 00083          BRB     10$                                     ; 0418
                                                 01  DD 00085 9$:      PUSHL   #1                                      ; 0430
                                     00000000G   8F  DD 00087          PUSHL   #DUMP$_HEADER                           ;
                                   63            02  FB 0008D          CALLS   #2, DUMP$OUTPUT_GETMSG                  ;
                                   62            00  FB 00090 10$:     CALLS   #0, DUMP$BLANK_LINE                     ; 0432
                                                     04 00093          RET                                             ; 0433

; Routine Size:  148 bytes,    Routine Base:  $CODE$ + 01DB
```

```
;  324          0434  1 GLOBAL ROUTINE dump$new_page: NOVALUE=
;  325          0435  2 BEGIN
;  326          0436  2 !
;  327          0437  2 ! Output a new page
;  328          0438  2 !
;  329          0439  2 linesthispage = 0;                              ! Reset count of lines/page
;  330          0440  2 dump$write($descriptor(%CHAR(%O'014')));         ! Output a form feed
;  331          0441  2 IF .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_for]
;  332          0442  2 OR NOT .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_fod]
;  333          0443  2 THEN
;  334          0444  3     BEGIN                                        ! Output "dump of device"
;  335          0445  3     dump$output_getmsg(
;  336          0446  3         dump$_dumpodev,
;  337          0447  3         %B'0001',
;  338          0448  3         dump$gl_idesc,
;  339          0449  3         dump$gq_time);
;  340          0450  3     END
;  341          0451  2 ELSE
;  342          0452  3     BEGIN                                        ! Output "dump of file"
;  343          0453  3     dump$output_getmsg(
;  344          0454  3         dump$_dumpofil,
;  345          0455  3         %B'0001',
;  346          0456  3         dump$gl_idesc,
;  347          0457  3         dump$gq_time);
;  348          0458  3     IF .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_rnd]
;  349          0459  3     THEN
;  350          0460  4         dump$output_getmsg(                      ! File ID and size
;  351          0461  4             dump$_fildnt,
;  352          0462  4             %B'0001',
;  353          0463  4             .dump$gl_inam[nam$w_fid_num] + .dump$gl_inam[nam$b_fid_nmx]^16,
;  354          0464  4             .dump$gl_inam[nam$w_fid_seq],
;  355          0465  4             .dump$gl_inam[nam$b_fid_rvn],
;  356          0466  4             .dump$gl_file_efblk,
;  357          0467  4             .dump$gl_file_hiblk);
;  358          0468  3     END;
;  359          0469  2
;  360          0470  2 dump$blank_line();
;  361          0471  1 END;


                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                              0C  0000C4 P.AAW:  .ASCII  <12>                                    ;
                                  0000C5         .BLKB   3
                        00000001  0000C8 P.AAV:  .LONG   1                                       ;
                        00000000' 0000CC         .ADDRESS P.AAW                                  ;


                                        .PSECT  $CODE$,NOWRT,2

                                  003C 00000     .ENTRY  DUMP$NEW_PAGE, Save R2,R3,R4,R5         ; 0434
                 55 00000000G   00  9E 00002     MOVAB   DUMP$GL_IDESC, R5                       ;
                 54 00000000G   00  9E 00009     MOVAB   DUMP$GQ_TIME, R4                        ;
                 53 00000000G   00  9E 00010     MOVAB   DUMP$GL_IFAB, R3                        ;
                 52 00000000V   EF  9E 00017     MOVAB   DUMP$OUTPUT_GETMSG, R2                  ;
```

```
                    00000000'   EF   D4 0001E          CLRL    LINESTHISPAGE                          ; 0439
                    00000000'   EF   9F 00024          PUSHAB  P.AAV                                  ; 0440
          00000000G 00          01   FB 0002A          CALLS   #1, DUMP$WRITE
                    50          63   D0 00031          MOVL    DUMP$GL_IFAB, R0                       ; 0441
                    05    43    A0   E8 00034          BLBS    67(R0), 1$
                11  41    A0         06   E0 00038     BBS     #6, 65(R0), 2$                         ; 0442
                                     54   DD 0003D 1$:  PUSHL   R4                                     ; 0445
                                     55   DD 0003F      PUSHL   R5
                                     01   DD 00041      PUSHL   #1
                    00000000G        8F   DD 00043      PUSHL   #DUMP$_DUMPODEV
                              62           04   FB 00049 CALLS   #4, DUMP$OUTPUT_GETMSG
                                           4C   11 0004C BRB     3$                                    ; 0441
                                     54   DD 0004E 2$:  PUSHL   R4                                     ; 0453
                                     55   DD 00050      PUSHL   R5
                                     01   DD 00052      PUSHL   #1
                    00000000G        8F   DD 00054      PUSHL   #DUMP$_DUMPOFIL
                              62           04   FB 0005A CALLS   #4, DUMP$OUTPUT_GETMSG
                              50           63   D0 0005D MOVL    DUMP$GL_IFAB, R0                      ; 0458
                35    43    A0         04   E1 00060     BBC     #4, 67(R0), 3$
                    00000000G 00       DD   DD 00065     PUSHL   DUMP$GL_FILE_HIBLK                    ; 0467
                    00000000G 00       DD   DD 0006B     PUSHL   DUMP$GL_FILE_EFBLK                    ; 0466
                              50 00000000G 00   D0 00071 MOVL    DUMP$GL_INAM, R0                      ; 0465
                              7E    28   A0   9A 00078 MOVZBL  40(R0), -(SP)
                              7E    26   A0   3C 0007C MOVZWL  38(R0), -(SP)                          ; 0464
                              51    24   A0   3C 00080 MOVZWL  36(R0), R1                            ; 0463
                              50    29   A0   9A 00084 MOVZBL  41(R0), R0
                50            50         10   78 00088 ASHL    #16, R0, R0
                                       6041   9F 0008C PUSHAB  (R0)[R1]
                                           01   DD 0008F PUSHL   #1                                    ; 0460
                    00000000G              8F   DD 00091 PUSHL   #DUMP$_FILDNT
                              62           07   FB 00097 CALLS   #7, DUMP$OUTPUT_GETMSG
                00000000V     EF           00   FB 0009A 3$:  CALLS   #0, DUMP$BLANK_LINE               ; 0470
                                           04 000A1      RET                                           ; 0471

; Routine Size:  162 bytes,    Routine Base:  $CODE$ + 026F
```

```
;  363          0472  1 GLOBAL ROUTINE dump$output_getmsg(messageid,messageflags,args): NOVALUE=
;  364          0473  2 BEGIN
;  365          0474  2 !
;  366          0475  2 ! Routine to do a $GETMSG and then FAO and output it.
;  367          0476  2 !
;  368          0477  2 ! Inputs:
;  369          0478  2 !
;  370          0479  2 !       messageid       id of message
;  371          0480  2 !       messageflags    flags for GETMSG
;  372          0481  2 !       args            first of n args
;  373          0482  2 !
;  374          0483  2 LOCAL
;  375          0484  2     status,
;  376          0485  2     outbuf : BBLOCK[dump$c_maxlisiz],
;  377          0486  2     outbufdesc : BBLOCK[dsc$c_s_bln],
;  378          0487  2     faoctrbuf : BBLOCK[dump$c_maxlisiz],
;  379          0488  2     faoctrdesc : BBLOCK[dsc$c_s_bln];
;  380          0489  2
;  381          0490  2
;  382          0491  2 CH$FILL(0,dsc$c_s_bln,faoctrdesc);
;  383          0492  2 CH$FILL(0,dsc$c_s_bln,outbufdesc);
;  384          0493  2 faoctrdesc[dsc$w_length] = dump$c_maxlisiz;
;  385          0494  2 faoctrdesc[dsc$a_pointer] = faoctrbuf;
;  386          0495  2 outbufdesc[dsc$w_length] = dump$c_maxlisiz;
;  387          0496  2 outbufdesc[dsc$a_pointer] = outbuf;
;  388          0497  2
;  389          0498  2
;  390        P 0499  2 status = $GETMSG(
;  391        P 0500  2     msgid=.messageid,
;  392        P 0501  2     msglen=faoctrdesc,
;  393        P 0502  2     bufadr=faoctrdesc,
;  394          0503  2     flags=.messageflags);
;  395          0504  2 IF NOT .status
;  396          0505  2 THEN
;  397          0506  2     SIGNAL_STOP(dump$_facility^16 + shr$_badlogic + sts$k_severe);
;  398          0507  2
;  399          0508  2
;  400        P 0509  2 status = $FAOL(
;  401        P 0510  2     ctrstr=faoctrdesc,
;  402        P 0511  2     outbuf=outbufdesc,
;  403        P 0512  2     outlen=outbufdesc,
;  404          0513  2     prmlst=args);
;  405          0514  2 IF NOT .status
;  406          0515  2 THEN
;  407          0516  2     SIGNAL_STOP(dump$_facility^16 + shr$_badlogic + sts$k_severe);
;  408          0517  2
;  409          0518  2
;  410          0519  2 dump$put_line(outbufdesc);
;  411          0520  1 END;


                                                .EXTRN  SYS$GETMSG, SYS$FAOL

                                  007C 00000    .ENTRY  DUMP$OUTPUT_GETMSG, Save R2,R3,R4,R5,R6     ; 0472
                    56 00000000G   00  9E 00002 MOVAB   LIB$STOP, R6                                ;
                    5E      FEE8   CE  9E 00009 MOVAB   -280(SP), SP                                ;
```

```
                08              00              6E                      00  2C 0000E        MOVC5   #0, (SP), #0, #8, FAOCTRDESC    ; 0491
                                                6E                          00013
                08              00              6E                      00  2C 00014        MOVC5   #0, (SP), #0, #8, OUTBUFDESC    ; 0492
                                                    FF74                CD      00019
                                                6E      84              8F  9B 0001C        MOVZBW  #132, FAOCTRDESC                ; 0493
                                          04    AE      08              AE  9E 00020        MOVAB   FAOCTRBUF, FAOCTRDESC+4         ; 0494
                                        FF74    CD      84              8F  9B 00025        MOVZBW  #132, OUTBUFDESC                ; 0495
                                        FF78    CD      FF7C            CD  9E 0002B        MOVAB   OUTBUF, OUTBUFDESC+4            ; 0496
                                                                        7E  D4 00032        CLRL    -(SP)                           ; 0503
                                                        08              AC  DD 00034        PUSHL   MESSAGEFLAGS
                                                        08              AE  9F 00037        PUSHAB  FAOCTRDESC
                                                        0C              AE  9F 0003A        PUSHAB  FAOCTRDESC
                                                        04              AC  DD 0003D        PUSHL   MESSAGEID
                                            00000000G   00                  05  FB 00040        CALLS   #5, SYS$GETMSG
                                                        52                  50  D0 00047        MOVL    R0, STATUS
                                                        09                  52  E8 0004A        BLBS    STATUS, 1$              ; 0504
                                                            00000000*       8F  DD 0004D        PUSHL   #<<<DUMP$_FACILITY@16>+4384>+4> ; 0506
                                                        66                  01  FB 00053        CALLS   #1, LIB$STOP
                                                            0C              AC  9F 00056 1$:    PUSHAB  ARGS                    ; 0513
                                                        FF74                CD  9F 00059        PUSHAB  OUTBUFDESC
                                                        FF74                CD  9F 0005D        PUSHAB  OUTBUFDESC
                                                            0C              AE  9F 00061        PUSHAB  FAOCTRDESC
                                            00000000G   00                  04  FB 00064        CALLS   #4, SYS$FAOL
                                                        52                  50  D0 0006B        MOVL    R0, STATUS
                                                        09                  52  E8 0006E        BLBS    STATUS, 2$              ; 0514
                                                            00000000*       8F  DD 00071        PUSHL   #<<<DUMP$_FACILITY@16>+4384>+4> ; 0516
                                                        66                  01  FB 00077        CALLS   #1, LIB$STOP
                                                        FF74                CD  9F 0007A 2$:    PUSHAB  OUTBUFDESC              ; 0519
                                            00000000V   EF                      01  FB 0007E        CALLS   #1, DUMP$PUT_LINE
                                                                                04  00085        RET                             ; 0520

; Routine Size:  134 bytes,    Routine Base:  $CODE$ + 0311
```

```
;  413          0521  1 GLOBAL ROUTINE dump$blank_line: NOVALUE=
;  414          0522  2 BEGIN
;  415          0523  2 !
;  416          0524  2 ! Write blank line to listing file.
;  417          0525  2 !
;  418          0526  2 dump$put_line($descriptor(''));
;  419          0527  1 END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                000D0 P.AAY:    .BLKB   0                                        ;
                                    00000000    000D0 P.AAX:    .LONG   0                                        ;
                                    00000000'   000D4           .ADDRESS P.AAY                                   ;


                                                        .PSECT  $CODE$,NOWRT,2

                                                  0000 00000            .ENTRY  DUMP$BLANK_LINE, Save nothing    ; 0521
                                        00000000'   EF  9F 00002        PUSHAB  P.AAX                            ; 0526
                            00000000V   EF              01  FB 00008    CALLS   #1, DUMP$PUT_LINE                ;
                                                        04  0000F      RET                                      ; 0527

; Routine Size:  16 bytes,    Routine Base:  $CODE$ + 0397
```

```
;  421          0528  1 GLOBAL ROUTINE dump$put_line(desc): NOVALUE=
;  422          0529  2 BEGIN
;  423          0530  2 !
;  424          0531  2 ! This routine forces a page break if the lines per page is
;  425          0532  2 ! about to be exceeded, provided that the device is a disk.
;  426          0533  2 !
;  427          0534  2 IF .linesthispage GEQ .dump$gl_lpp                ! if lines per page exceeded
;  428          0535  2 AND .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_rnd] ! device is disk
;  429          0536  2 THEN
;  430          0537  2     dump$new_page();                              ! then new page
;  431          0538  2
;  432          0539  2
;  433          0540  2 dump$write(.desc);
;  434          0541  2 linesthispage = .linesthispage + 1;
;  435          0542  1 END;


                                                0004 00000          .ENTRY  DUMP$PUT_LINE, Save R2                  ; 0528
                          52 00000000'  EF   9E 00002          MOVAB   LINESTHISPAGE, R2                       ;
             00000000G    00            62   D1 00009          CMPL    LINESTHISPAGE, DUMP$GL_LPP              ; 0534
                                        11   19 00010          BLSS    1$                                      ;
                          50 00000000G  00   D0 00012          MOVL    DUMP$GL_IFAB, R0                        ; 0535
                  05        43        A0   04 E1 00019          BBC     #4, 67(R0), 1$                          ;
                          FEA5          CF   00 FB 0001E          CALLS   #0, DUMP$NEW_PAGE                       ; 0537
                                             04 AC DD 00023 1$:   PUSHL   DESC                                    ; 0540
             00000000G    00                  01 FB 00026          CALLS   #1, DUMP$WRITE                          ;
                                              62 D6 0002D          INCL    LINESTHISPAGE                           ; 0541
                                                 04 0002F          RET                                             ; 0542

; Routine Size:  48 bytes,    Routine Base:  $CODE$ + 03A7
```

```
;  437     0543  1 GLOBAL ROUTINE dump$dump_buffer(bufdesc,status,header): NOVALUE=
;  438     0544  2 BEGIN
;  439     0545  2 !
;  440     0546  2 ! This routine does all the work of dumping the buffer.
;  441     0547  2 !
;  442     0548  2 MAP
;  443     0549  2     bufdesc : REF BBLOCK[dsc$c_s_bln];
;  444     0550  2 BIND
;  445     0551  2     buffer = .bufdesc[dsc$a_pointer] : VECTOR[,BYTE];
;  446     0552  2 LOCAL
;  447     0553  2     tempbuffer : BBLOCK[512],
;  448     0554  2     tempdesc : BBLOCK[dsc$c_s_bln],
;  449     0555  2     tempfaobuf : BBLOCK[max_fao_size],
;  450     0556  2     additional,
;  451     0557  2     padbytes,
;  452     0558  2     bufferpointer,
;  453     0559  2     faopointer,
;  454     0560  2     number,
;  455     0561  2     bytesperline,
;  456     0562  2     bytesleft,
;  457     0563  2     entsinbuf;
;  458     0564  2
;  459     0565  2 faopointer = faoctrdesc;                                ! Assume full line
;  460     0566  2 dump$put_header(.bufdesc, .header);
;  461     0567  2 IF NOT .status
;  462     0568  2 THEN
;  463     0569  3     BEGIN
;  464     0570  3     dump$output_getmsg(.status, %B'1111');              ! Put out error status
;  465     0571  3     dump$blank_line();
;  466     0572  2     END;
;  467     0573  2
;  468     0574  2 IF NOT .header
;  469     0575  2 AND .dump$gl_flags[dump$v_file_header]
;  470     0576  2 AND .bufdesc[dsc$w_length] EQL 512
;  471     0577  2 THEN
;  472     0578  2     IF dump$one_header(.bufdesc[dsc$a_pointer])
;  473     0579  2     THEN
;  474     0580  2         RETURN;
;  475     0581  2
;  476     0582  2
;  477     0583  2 number = 0;                                             ! Local index number
;  478     0584  2 bytesperline = .entsperline*.entrysize;
;  479     0585  4 entsinbuf = ((.bufdesc[dsc$w_length]+.entrysize-1)
;  480     0586  2                 AND NOT (.entrysize-1))/.entrysize;
;  481     0587  2 bytesleft = .bufdesc[dsc$w_length];
;  482     0588  2 IF NOT .dump$gl_flags[dump$v_number]                    ! If /NUMBER not used,
;  483     0589  2 THEN dump$gl_number = 0;                                ! start each at zero
;  484     0590  2
;  485     0591  2 WHILE .entsinbuf GTR 0 DO
;  486     0592  3     BEGIN
;  487     0593  3     IF .bytesleft LSSU .bytesperline
;  488     0594  3     THEN
;  489     0595  4         BEGIN
;  490     0596  4         CH$COPY(.bytesleft,buffer[.number],0,.bytesperline,tempbuffer); ! Copy partial line, zero fill to en
;  491     0597  4         tempdesc[dsc$w_length] = max_fao_size;              ! Set up work area for parti
;  492     0598  4         tempdesc[dsc$a_pointer] = tempfaobuf;
;  493     0599  4         SYS$FAO(plinfaodesc,tempdesc,tempdesc,,entsinbuf);  ! Set up fao with # of entri
```

```
;  494          0600  4          faopointer = tempdesc;                                          ! Use this fao control strin
;  495          0601  4          bufferpointer = tempbuffer;
;  496          0602  4          dump$gl_outdesc[dsc$w_length] = .dump$gl_width;                 ! Set output length to defau
;  497          0603  4          dump$fao_line(.bufferpointer,.entsperline,.entrysize,           ! Format the output line
;  498          0604  4                  .dump$gl_number,.entsinbuf,.dumpmode,.faopointer,dump$gl_outdesc);
;  499          0605  4
;  500          0606  4
;  501          0607  4          ! Now that the partial line is ready to be written, suppress any
;  502          0608  4          ! leading zeros.
;  503          0609  4          !
;  504          0610  4          ! NOTE: Due to  the fact that RMS does reads on WORD offsets, the first
;  505          0611  4          !       leading byte of zeros will not be replaced if the dump is in
;  506          0612  4          !       block mode and ends up on a byte offset.
;  507          0613  4          !
;  508          0614  4          additional = 0;
;  509          0615  4          padbytes = .dumpwidth - .dump$gl_outdesc[dsc$w_length];         ! Calculate padding (word offset)
;  510          0616  4          IF NOT .dump$gl_flags[dump$v_decimal]                           ! If HEX or OCTAL dump
;  511          0617  4          THEN
;  512          0618  5              BEGIN                                                       ! calculate further padding if neces
;  513          0619  5              IF (additional = .bytesleft MOD .entrysize) GTR 0           ! Find additional offset
;  514          0620  5              THEN additional = (.entrysize - .additional) *              ! Customize it to type of dump
;  515          0621  5                                .charsperbyte[.modeindex/3] + 1;
;  516          0622  5              padbytes = .padbytes + .additional;
;  517          0623  4              END;
;  518          0624  4
;  519          0625  4          CH$MOVE(.dump$gl_outdesc[dsc$w_length] - .additional,
;  520          0626  4                  .dump$gl_outdesc[dsc$a_pointer] + .additional,
;  521          0627  4                  .dump$gl_outdesc[dsc$a_pointer] + .padbytes);
;  522          0628  4          CH$FILL(%C' ',.padbytes,.dump$gl_outdesc[dsc$a_pointer]);       ! Move blanks to pad areas
;  523          0629  4          dump$gl_outdesc[dsc$w_length] = .dumpwidth;                     ! Set output length to
;  524          0630  4          END                                                             ! device type.
;  525          0631  3      ELSE
;  526          0632  4          BEGIN
;  527          0633  4          bufferpointer = buffer[.number];                                ! Dump full line
;  528          0634  4          dump$gl_outdesc[dsc$w_length] = .dump$gl_width;                 ! Set output length to default value
;  529          0635  4          dump$fao_line(.bufferpointer,.entsperline,.entrysize,           ! Format the output line
;  530          0636  4                  .dump$gl_number,.entsinbuf,.dumpmode,.faopointer,dump$gl_outdesc);
;  531          0637  3          END;
;  532          0638  3
;  533          0639  3      dump$put_line(dump$gl_outdesc);                                     ! Put line out to device
;  534          0640  3      number = .number + .bytesperline;                           ! Calculate next index
;  535          0641  3      IF .dump$gl_flags[dump$v_number]                                    ! If /NUMBER qualifier used
;  536          0642  3      THEN
;  537          0643  3          dump$gl_number = .dump$gl_number + .bytesperline                ! then keep cumulative index,
;  538          0644  3      ELSE
;  539          0645  3          dump$gl_number = .number;                                       ! else make index local
;  540          0646  3      entsinbuf = .entsinbuf - .entsperline;                              ! Update # of entry's in buffer
;  541          0647  3      bytesleft = .bytesleft - (.entsperline*.entrysize);                 ! Calculate how many bytes left in b
;  542          0648  2      END;
;  543          0649  1 END;


                                0FFC 00000          .ENTRY  DUMP$DUMP_BUFFER, Save R2,R3,R4,R5,R6,R7,-  ; 0543
                                                            R8,R9,R10,R11                               ;
```

```
                    5E      FDC8    CE   9E 00002                 MOVAB   -568(SP), SP
                    52        04    AC   D0 00007                 MOVL    BUFDESC, R2                                     ;  0551
                              04    A2   DD 0000B                 PUSHL   4(R2)
                        00000000'   EF   9F 0000E                 PUSHAB  FAOCTRDESC                                      ;  0565
                              0C    AC   DD 00014                 PUSHL   HEADER                                          ;  0566
                                    52   DD 00017                 PUSHL   R2
                          FDE6      CF   02   FB 00019            CALLS   #2, DUMP$PUT_HEADER
                          0E        08   AC   E8 0001E            BLBS    STATUS, 1$                                      ;  0567
                                    0F   DD 00022                 PUSHL   #15                                             ;  0570
                                    08   AC   DD 00024            PUSHL   STATUS
                          FF0E      CF   02   FB 00027            CALLS   #2, DUMP$OUTPUT_GETMSG
                            90      AF   00   FB 0002C            CALLS   #0, DUMP$BLANK_LINE                             ;  0571
                            1D        0C   AC   E8 00030 1$:      BLBS    HEADER, 2$                                      ;  0574
                15 00000000G        00   04   E1 00034           BBC     #4, DUMP$GL_FLAGS, 2$                           ;  0575
                          0200      8F   62   B1 0003C            CMPW    (R2), #512                                      ;  0576
                                    0E   12   00041               BNEQ    2$
                                    04   A2   DD 00043            PUSHL   4(R2)                                           ;  0578
                       00000000G    00   01   FB 00046           CALLS   #1, DUMP$ONE_HEADER
                                    01   50   E9 0004D            BLBC    R0, 2$
                                         04   00050               RET
                                    5B   D4   00051 2$:          CLRL    NUMBER                                          ;  0583
                                51  00000000'   EF   D0 00053     MOVL    ENTRYSIZE, R1                                   ;  0584
                5A 00000000'        EF   51   C5 0005A           MULL3   R1, ENTSPERLINE, BYTESPERLINE
                                    50   62   3C 00062           MOVZWL  (R2), R0                                        ;  0585
                                    52     FF A140   9E 00065     MOVAB   -1(R1)[R0], R2
                                    53     FF   A1   9E 0006A     MOVAB   -1(R1), R3                                      ;  0586
                                    52   53   CA 0006E            BICL2   R3, R2
                58                  52   51   C7 00071           DIVL3   R1, R2, ENTSINBUF
                                    59   50   D0 00075           MOVL    R0, BYTESLEFT                                   ;  0587
                06 00000000G        00   01   E0 00078           BBS     #1, DUMP$GL_FLAGS+1, 3$                         ;  0588
                                 000000000G   00   D4 00080      CLRL    DUMP$GL_NUMBER                                  ;  0589
                                         58   D5 00086 3$:       TSTL    ENTSINBUF                                       ;  0591
                                         01   14 00088           BGTR    4$
                                              04 0008A           RET
                                      5A         59   D1 0008B 4$:       CMPL    BYTESLEFT, BYTESPERLINE                 ;  0593
                                         03   1F 0008E           BLSSU   5$
                                       00DF   31 00090           BRW     8$
                                      57        04   AE   D0 00093 5$:       MOVL    4(SP), R7                       ;  0596
                5A          00          6B47        59   2C 00097           MOVC5   BYTESLEFT, (NUMBER)[R7], #0, BYTESPERLINE, -
                                              40   AE        0009D                 TEMPBUFFER
                                        38    AE        28   B0 0009F           MOVW    #40, TEMPDESC                   ;  0597
                                        3C    AE          10   AE 9E 000A3     MOVAB   TEMPFAOBUF, TEMPDESC+4          ;  0598
                                                    58   DD 000A8           PUSHL   ENTSINBUF                       ;  0599
                                                  3C   AE 9F 000AA           PUSHAB  TEMPDESC
                                                  40   AE 9F 000AD           PUSHAB  TEMPDESC
                                           00000000'   EF 9F 000B0           PUSHAB  PLINFAODESC
                                 00000000G    00        04   FB 000B6           CALLS   #4, SYS$FAO
                                              6E          38   AE 9E 000BD     MOVAB   TEMPDESC, FAOPOINTER            ;  0600
                                        08    AE          40   AE 9E 000C1     MOVAB   TEMPBUFFER, BUFFERPOINTER       ;  0601
                                 00000000G    00 000000000G   00 B0 000C6     MOVW    DUMP$GL_WIDTH, DUMP$GL_OUTDESC  ;  0602
                                         000000000G   00 9F 000D1           PUSHAB  DUMP$GL_OUTDESC                 ;  0603
                                                  04   AE   DD 000D7           PUSHL   FAOPOINTER                      ;  0604
                                           00000000'   EF   DD 000DA           PUSHL   DUMPMODE
                                                  58   DD 000E0           PUSHL   ENTSINBUF
                                         000000000G   00   DD 000E2           PUSHL   DUMP$GL_NUMBER
                                           00000000'   EF   DD 000E8           PUSHL   ENTRYSIZE                       ;  0603
                                           00000000'   EF   DD 000EE           PUSHL   ENTSPERLINE
```

```
                                  24    AE  DD 000F4        PUSHL   BUFFERPOINTER                                    ;
                  00000000G  00         08  FB 000F7        CALLS   #8, DUMP$FAO_LINE                                ;
                                        56  D4 000FE        CLRL    ADDITIONAL                                       ; 0614
                         50 00000000G   00  3C 00100        MOVZWL  DUMP$GL_OUTDESC, R0                              ; 0615
             0C   AE 00000000'   EF     50  C3 00107        SUBL3   R0, DUMPWIDTH, PADBYTES                          ;
                  35 00000000G   00     03  E0 00110        BBS     #3, DUMP$GL_FLAGS, 7$                            ; 0616
       7E         00             59     01  7A 00118        EMUL    #1, BYTESLEFT, #0, -(SP)                         ; 0619
       56         56             8E 00000000'  EF  7B 0011D EDIV    ENTRYSIZE, (SP)+, ADDITIONAL, ADDITIONAL         ;
                                        56  D5 00126        TSTL    ADDITIONAL                                       ;
                                        1F  15 00128        BLEQ    6$                                               ;
                  51 00000000'   EF     56  C3 0012A        SUBL3   ADDITIONAL, ENTRYSIZE, R1                        ; 0620
                  52 00000000'   EF     03  C7 00132        DIVL3   #3, MODEINDEX, R2                                ; 0621
                         53 00000000'EF42   9A 0013A        MOVZBL  CHARSPERBYTE[R2], R3                             ;
                                   51   53  C4 00142        MULL2   R3, R1                                           ;
                         56         01  A1  9E 00145        MOVAB   1(R1), ADDITIONAL                                ;
                      0C AE             56  C0 00149 6$:    ADDL2   ADDITIONAL, PADBYTES                             ; 0622
                         50             56  C2 0014D 7$:    SUBL2   ADDITIONAL, R0                                   ; 0625
                         57 00000000G   00  D0 00150        MOVL    DUMP$GL_OUTDESC+4, R7                            ; 0626
             0C BE47           6647     50  28 00157        MOVC3   R0, (ADDITIONAL)[R7], @PADBYTES[R7]              ; 0627
    0C    AE      20             6E     00  2C 0015E        MOVC5   #0, (SP), #32, PADBYTES, (R7)                    ; 0628
                                        67     00164
                  00000000G  00 00000000'  EF  B0 00165     MOVW    DUMPWIDTH, DUMP$GL_OUTDESC                       ; 0629
                                        3E  11 00170        BRB     9$                                               ; 0593
                      08  AE        04 BE4B 9E 00172 8$:    MOVAB   @4(SP)[NUMBER], BUFFERPOINTER                    ; 0633
                  00000000G  00 00000000G  00  B0 00178     MOVW    DUMP$GL_WIDTH, DUMP$GL_OUTDESC                   ; 0634
                             00000000G  00  9F 00183        PUSHAB  DUMP$GL_OUTDESC                                  ; 0635
                                    04  AE  DD 00189        PUSHL   FAOPOINTER                                       ; 0636
                             00000000'  EF  DD 0018C        PUSHL   DUMPMODE                                         ;
                                        58  DD 00192        PUSHL   ENTSINBUF                                        ;
                             00000000G  00  DD 00194        PUSHL   DUMP$GL_NUMBER                                   ;
                             00000000'  EF  DD 0019A        PUSHL   ENTRYSIZE                                        ; 0635
                             00000000'  EF  DD 001A0        PUSHL   ENTSPERLINE                                      ;
                                    24  AE  DD 001A6        PUSHL   BUFFERPOINTER                                    ;
                  00000000G  00         08  FB 001A9        CALLS   #8, DUMP$FAO_LINE                                ;
                             00000000G  00  9F 001B0 9$:    PUSHAB  DUMP$GL_OUTDESC                                  ; 0639
                       FE15  CF         01  FB 001B6        CALLS   #1, DUMP$PUT_LINE                                ;
                             5B         5A  C0 001BB        ADDL2   BYTESPERLINE, NUMBER                             ; 0640
                  09 00000000G  00      01  E1 001BE        BBC     #1, DUMP$GL_FLAGS+1, 10$                         ; 0641
                  00000000G  00         5A  C0 001C6        ADDL2   BYTESPERLINE, DUMP$GL_NUMBER                     ; 0643
                                        07  11 001CD        BRB     11$                                              ;
                  00000000G  00         5B  D0 001CF 10$:   MOVL    NUMBER, DUMP$GL_NUMBER                           ; 0645
                             58 00000000'  EF  C2 001D6 11$: SUBL2  ENTSPERLINE, ENTSINBUF                           ; 0646
                  50 00000000'  EF 00000000'  EF  C5 001DD  MULL3   ENTRYSIZE, ENTSPERLINE, R0                       ; 0647
                             59         50  C2 001E9        SUBL2   R0, BYTESLEFT                                    ;
                                  FE97  31 001EC            BRW     3$                                               ; 0591
                                        04 001EF            RET                                                      ; 0649

; Routine Size:  496 bytes,    Routine Base:  $CODE$ + 03D7
```

```
;  545          0650  1 END
;  546          0651  0 ELUDOM



                                                        .EXTRN  LIB$STOP

;                                PSECT SUMMARY
;
;      Name                      Bytes                         Attributes
;
;  $OWN$                           120  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                          216  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         1479  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]STARLET.L32;1     9776        20         0       581         00:00.7



;                                    COMMAND QUALIFIERS

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

; Size:          1479 code + 336 data bytes
; Run Time:         00:14.7
; Elapsed Time:     00:58.9
; Lines/CPU Min:     2658
; Lexemes/CPU-Min: 23897
; Memory Used:  152 pages
; Compilation Complete
```

DTSMAIN
LIS

DUMP
LIS

DUMPFILE
LIS

DTSMSG
LIS

DTSPARSE
LIS

DTRTEST
LIS

DUMPFAOLN
LIS

DUMPREQ

DUMP

DUMP
MAP

DUMPHEADE
LIS

DTSAST
LIS

DTSTEST
LIS