```
DDDDDDDDDDDD     UUU         UUU  MMM         MMM  PPPPPPPPPPPP
DDDDDDDDDDDDD    UUU         UUU  MMM         MMM  PPPPPPPPPPPPP
DDDDDDDDDDDDD    UUU         UUU  MMM         MMM  PPPPPPPPPPPPP
DDD          DDD UUU         UUU  MMMMMMM   MMMMMM  PPP         PPP
DDD          DDD UUU         UUU  MMMMMMM   MMMMMM  PPP         PPP
DDD          DDD UUU         UUU  MMMMMMM   MMMMMM  PPP         PPP
DDD          DDD UUU         UUU  MMM  MMM  MMM  MM  PPP         PPP
DDD          DDD UUU         UUU  MMM  MMM  MMM  MM  PPP         PPP
DDD          DDD UUU         UUU  MMM  MMM  MMM  MM  PPP         PPP
DDD          DDD UUU         UUU  MMM       MMM  MM  PPPPPPPPPPPPP
DDD          DDD UUU         UUU  MMM       MMM  MM  PPPPPPPPPPPPP
DDD          DDD UUU         UUU  MMM       MMM  MM  PPPPPPPPPPPPP
DDD          DDD UUU         UUU  MMM       MMM  MM  PPP
DDD          DDD UUU         UUU  MMM       MMM  MM  PPP
DDD          DDD UUU         UUU  MMM       MMM  MM  PPP
DDD          DDD UUU         UUU  MMM       MMM  MM  PPP
DDD          DDD UUU         UUU  MMM       MMM  MM  PPP
DDD          DDD UUU         UUU  MMM       MMM  MM  PPP
DDD          DDD UUU         UUU  MMM       MMM  MM  PPP
DDDDDDDDDDDD     UUUUUUUUUUUUUUU  MMM       MMM  MM  PPP
DDDDDDDDDDDDD    UUUUUUUUUUUUUUU  MMM       MMM  MM  PPP
DDDDDDDDDDDDD    UUUUUUUUUUUUUUU  MMM       MMM  MM  PPP
```

FILEID**DUMP

```
DDDDDDDD    UU      UU  MM      MM  PPPPPPPP
DDDDDDDD    UU      UU  MM      MM  PPPPPPPP
DD      DD  UU      UU  MMMM  MMMM  PP      PP
DD      DD  UU      UU  MMMM  MMMM  PP      PP
DD      DD  UU      UU  MM  MM  MM  PP      PP
DD      DD  UU      UU  MM  MM  MM  PP      PP
DD      DD  UU      UU  MM      MM  PPPPPPPP
DD      DD  UU      UU  MM      MM  PPPPPPPP
DD      DD  UU      UU  MM      MM  PP
DD      DD  UU      UU  MM      MM  PP
DD      DD  UU      UU  MM      MM  PP          ....
DD      DD  UU      UU  MM      MM  PP          ....
DDDDDDDD    UUUUUUUUUU  MM      MM  PP          ....
DDDDDDDD    UUUUUUUUUU  MM      MM  PP          ....

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$MAIN (                                  ! File dump program
;      2     0002  0         IDENT='V04-000',
;      3     0003  0         MAIN=dump$start,
;      4     0004  0         ADDRESSING_MODE(EXTERNAL=GENERAL,
;      5     0005  0                 NONEXTERNAL=LONG_RELATIVE)
;      6     0006  0         ) =
;      7     0007  1 BEGIN
;      8     0008  1 !
;      9     0009  1 !
;     10     0010  1 !*****************************************************************************
;     11     0011  1 !*                                                                           *
;     12     0012  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;     13     0013  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;     14     0014  1 !*  ALL RIGHTS RESERVED.                                                      *
;     15     0015  1 !*                                                                           *
;     16     0016  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;     17     0017  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;     18     0018  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;     19     0019  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;     20     0020  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;     21     0021  1 !*  TRANSFERRED.                                                              *
;     22     0022  1 !*                                                                           *
;     23     0023  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;     24     0024  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;     25     0025  1 !*  CORPORATION.                                                              *
;     26     0026  1 !*                                                                           *
;     27     0027  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;     28     0028  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;     29     0029  1 !*                                                                           *
;     30     0030  1 !*                                                                           *
;     31     0031  1 !*****************************************************************************
;     32     0032  1 !
;     33     0033  1 !
;     34     0034  1 !++
;     35     0035  1 !
;     36     0036  1 ! FACILITY:  File dump utility
;     37     0037  1 !
;     38     0038  1 ! ABSTRACT:
;     39     0039  1 !       This module contains the command processing and driver routines.
;     40     0040  1 !
;     41     0041  1 ! ENVIRONMENT:
;     42     0042  1 !       VAX native, user mode.
;     43     0043  1 !
;     44     0044  1 ! AUTHOR:  Benn Schreiber, Stephen Zalewski      CREATION DATE:  22-Jun-1981
;     45     0045  1 !
;     46     0046  1 ! MODIFIED BY:
;     47     0047  1 !
;     48     0048  1 !       V03-013 SHZ0002         Stephen H. Zalewski     12-Apr-1984
;     49     0049  1 !               Move all conflicting qualifier checking to DUMP.CLD.
;     50     0050  1 !
;     51     0051  1 !       V03-012 BLS0258         Benn Schreiber           5-Jan-1984
;     52     0052  1 !               Clear fab$l_xab after opening the file.
;     53     0053  1 !
;     54     0054  1 !       V03-011 SHZ0001         Stephen H. Zalewski     28-Jun-1983
;     55     0055  1 !               Have DUMP open files shared.
;     56     0056  1 !
;     57     0057  1 !       V03-010 MLJ0095         Martin L. Jack, 17-Aug-1982  18:26
```

```
;    58          0058  1 !                 Remove references to CLI$END_PARSE.
;    59          0059  1 !
;    60          0060  1 !       V03-009 LMP0038         L. Mark Pilant,         30-Jun-1982  13:55
;    61          0061  1 !               Correct a problem that generated the BADSTART error message
;    62          0062  1 !               if the EOF block on a file was zero.
;    63          0063  1 !
;    64          0064  1 !       V03-008 LMP0034         L. Mark Pilant,         28-Jun-1982  9:36
;    65          0065  1 !               Fix a bug introduced by LMP0030 that resulted in an access
;    66          0066  1 !               violation when doing wildcard dumps.
;    67          0067  1 !
;    68          0068  1 !       V03-007 LMP0030         L. Mark Pilant,         15-Jun-1982  10:35
;    69          0069  1 !               Allow dumping of logical blocks on a Files-11 mounted disk.
;    70          0070  1 !
;    71          0071  1 !       V03-006 MLJ0081         Martin L. Jack, 24-Feb-1982  17:35
;    72          0072  1 !               Lengthen DUMP$GQ_TIME to avoid overwriting EXIT_STATUS.
;    73          0073  1 !
;    74          0074  1 !       V03-005 MLJ0059         Martin L. Jack, 6-Nov-1981  14:13
;    75          0075  1 !               Properly handle EFBLK of 0.
;    76          0076  1 !
;    77          0077  1 !       V03-004 MLJ0056         Martin L. Jack, 18-Oct-1981  23:31
;    78          0078  1 !               Special case reading from terminals to allow ^Z to function.
;    79          0079  1 !
;    80          0080  1 !       V03-003 MLJ0046         Martin L. Jack, 21-Sep-1981  18:27
;    81          0081  1 !               Allow for device name change between $PARSE and $OPEN.
;    82          0082  1 !
;    83          0083  1 !       V03-002 MLJ0045         Martin L. Jack, 10-Sep-1981  15:27
;    84          0084  1 !               Set SQO bit where appropriate.  Allow record mode dump of
;    85          0085  1 !               network device.  Allow DUMP/HEADER on tape.
;    86          0086  1 !
;    87          0087  1 !       V03-001 MLJ0033         Martin L. Jack, 23-Aug-1981  9:48
;    88          0088  1 !               Extensive rewriting to finish implementation.
;    89          0089  1 !
;    90          0090  1 !--
```

```
;     92          0091  1 LIBRARY 'SYS$LIBRARY:STARLET';
;     93          0092  1 LIBRARY 'SYS$LIBRARY:TPAMAC';
;     94          0093  1 REQUIRE 'SRC$:DUMPRE';
;     95          0209  1
;     96          0210  1
;     97          0211  1 FORWARD ROUTINE
;     98          0212  1     dump$handler,                       ! Top-level condition handler
;     99          0213  1     dump$start,                         ! Main routine
;    100          0214  1     dump$tparse,                        ! Call TPARSE
;    101          0215  1     dump$store_num,                     ! Store numeric qualifier value
;    102          0216  1     dump$open_input,                    ! Open input file
;    103          0217  1     dump$open_output,                   ! Open output file
;    104          0218  1     dump$read,                          ! Read from input file
;    105          0219  1     dump$write:          NOVALUE,       ! Write to output file
;    106          0220  1     dump$close_input:    NOVALUE,       ! Close input file
;    107          0221  1     dump$close_output:   NOVALUE,       ! Close output file
;    108          0222  1     dump$list_width:     NOVALUE,       ! Get listing device width
;    109          0223  1     dump$file_error:     NOVALUE;       ! Signal file-related error
;    110          0224  1
;    111          0225  1
;    112          0226  1 EXTERNAL ROUTINE
;    113          0227  1     cli$get_value,                      ! Get qualifier value
;    114          0228  1     cli$present,                        ! Test if qualifier present
;    115          0229  1     dump$blank_line,                    ! Write blank line
;    116          0230  1     dump$dump_file,                     ! Dump the file
;    117          0231  1     dump$output_getmsg,                 ! Output a message
;    118          0232  1     lib$free_vm,                        ! Free virtual memory
;    119          0233  1     lib$get_vm,                         ! Allocate virtual memory
;    120          0234  1     lib$find_file,                      ! Search for wild card files
;    121          0235  1     lib$lp_lines,                       ! Number of lines on printer
;    122          0236  1     lib$tparse,                         ! Table-driven parser
;    123          0237  1     str$copy_dx;                        ! Copy a string
;    124          0238  1
;    125          0239  1
;    126          0240  1 EXTERNAL LITERAL
;    127          0241  1     dump$_facility,
;    128          0242  1     dump$_badrange,
;    129          0243  1     dump$_confqual,
;    130          0244  1     dump$_devquals,
;    131          0245  1     dump$_devspec,
;    132          0246  1     dump$_getchn,
;    133          0247  1     dump$_endoffile,
;    134          0248  1     dump$_novirmem,
;    135          0249  1     dump$_badstart;
;    136          0250  1
;    137          0251  1
;    138          0252  1 GLOBAL
;    139          0253  1     dump$gl_ifab : REF BBLOCK,                  ! Pointer to input FAB
;    140          0254  1     dump$gl_inam : REF BBLOCK,                  ! Pointer to input NAM block
;    141          0255  1     dump$gl_irab : $RAB_DECL,                   ! Input RAB
;    142          0256  1     dump$gl_orab : $RAB_DECL,                   ! Output RAB
;    143          0257  1     dump$gl_ofab : $FAB_DECL,                   ! Output FAB
;    144          0258  1     dump$gl_onam : $NAM_DECL,                   ! Output NAM block
;    145          0259  1     dump$gl_orss : BBLOCK[nam$c_maxrss],        ! Output resultant string
;    146          0260  1     dump$gl_idesc : BBLOCK[dsc$c_s_bln],        ! Descriptor for input RSA
;    147          0261  1     dump$gl_odesc : BBLOCK[dsc$c_s_bln],        ! Descriptor for output RSA
;    148          0262  1     dump$ab_outbuf : BBLOCK[dump$c_maxlisiz],   ! Output buffer
```

```
;  149          0263  1      dump$gl_outdesc : BBLOCK[dsc$c_s_bln],      ! Descriptor for output buffer
;  150          0264  1      dump$gl_channel,                            ! Input channel
;  151          0265  1      dump$gl_width,                              ! Width of listing
;  152          0266  1      dump$gl_lpp,                                ! Lines per page
;  153          0267  1      dump$gl_buffer : BBLOCK[dsc$c_s_bln],       ! Descriptor for input buffer
;  154          0268  1      dump$gl_flags : BBLOCK[4],                  ! General flags
;  155          0269  1      dump$gl_start_qual,                         ! Value of START qualifier
;  156          0270  1      dump$gl_end_qual,                           ! Value of END qualifier
;  157          0271  1      dump$gl_count_qual,                         ! Value of COUNT qualifier
;  158          0272  1      dump$gl_number_qual,                        ! Value of NUMBER qualifier
;  159          0273  1      dump$gl_number,                             ! Local byte offset for NUMBER
;  160          0274  1      dump$gl_cur_block,                          ! Current block number
;  161          0275  1      dump$gl_max_block,                          ! Highest block to be dumped
;  162          0276  1      dump$gl_file_efblk,                         ! End of file block
;  163          0277  1      dump$gl_file_hiblk,                         ! Highest allocated block
;  164          0278  1      dump$gl_record,                             ! Current block/record number
;  165          0279  1      dump$gq_time : VECTOR[2];                   ! Time at beginning of dump
;  166          0280  1
;  167          0281  1
;  168          0282  1  OWN
;  169          0283  1      exit_status : BBLOCK[4] INITIAL(ss$_normal),! Most severe error status
;  170          0284  1      tpa_block : BBLOCK[tpa$k_length0];          ! TPARSE block
```

```
;  172          0285  1 LITERAL
;  173          0286  1          dump$m_tpa_start=       $fieldmask(dump$v_tpa_start),
;  174          0287  1          dump$m_tpa_count=       $fieldmask(dump$v_tpa_count),
;  175          0288  1          dump$m_tpa_end=         $fieldmask(dump$v_tpa_end);
;  176          0289  1
;  177          0290  1
;  178          0291  1 ! TPARSE tables to parse /BLOCK and /RECORD qualifier values.
;  179          0292  1 !
;  180          0293  1 $INIT_STATE(blkrec_states, blkrec_keys);
;  181        P 0294  1 $STATE(,
;  182        P 0295  1          ('START',,,dump$m_tpa_start,dump$gl_flags),
;  183        P 0296  1          ('END'  ,,,dump$m_tpa_end,  dump$gl_flags),
;  184          0297  1          ('COUNT',,,dump$m_tpa_count,dump$gl_flags));
;  185        P 0298  1 $STATE(,
;  186        P 0299  1          ('='),
;  187          0300  1          (':'));
;  188        P 0301  1 $STATE(parse_number,
;  189        P 0302  1          (tpa$_decimal,eos,dump$store_num),      ! Decimal number
;  190          0303  1          ('%'));                                 ! Base prefix
;  191        P 0304  1 $STATE(,
;  192        P 0305  1          ('X'),                                  ! Hex base designator
;  193        P 0306  1          ('O',octnum),                           ! Octal base designator
;  194          0307  1          ('D',decnum));                          ! Decimal base designator
;  195        P 0308  1 $STATE(,
;  196          0309  1          (tpa$_hex,eos,dump$store_num));         ! Introduced hex number
;  197        P 0310  1 $STATE(octnum,
;  198          0311  1          (tpa$_octal,eos,dump$store_num));       ! Introduced octal number
;  199        P 0312  1 $STATE(decnum,
;  200          0313  1          (tpa$_decimal,,dump$store_num));        ! Introduced decimal number
;  201        P 0314  1 $STATE(eos,
;  202          0315  1          (tpa$_eos,tpa$_exit));                  ! End of string
;  203          0316  1
;  204          0317  1
;  205          0318  1 ! TPARSE table to parse /NUMBER qualifier.
;  206          0319  1 !
;  207          0320  1 $INIT_STATE(number_states, number_keys);
;  208        P 0321  1 $STATE(,
;  209          0322  1          ((parse_number),tpa$_exit));            ! /NUMBER=
```

```
;  211          0323  1 ROUTINE dump$handler(sigargs, mechargs)=
;  212          0324  2 BEGIN
;  213          0325  2 !
;  214          0326  2 ! This routine is a condition handler established by the main
;  215          0327  2 ! routine.  It saves the most severe condition for the exit status.
;  216          0328  2 !
;  217          0329  2 MAP
;  218          0330  2     sigargs : REF BBLOCK,
;  219          0331  2     mechargs : REF BBLOCK;
;  220          0332  2 BIND
;  221          0333  2     signame = sigargs[chf$l_sig_name] : BBLOCK; ! Name of signal
;  222          0334  2
;  223          0335  2
;  224          0336  2 IF NOT .signame                                ! If an error signal
;  225          0337  4     AND ((.signame[sts$v_severity]             ! and severity is worse
;  226          0338  4         GTRU .exit_status[sts$v_severity])
;  227          0339  3         OR .exit_status[sts$v_severity])       ! or no errors yet
;  228          0340  2 THEN
;  229          0341  2     exit_status = .signame;                    ! then save it for exit
;  230          0342  2
;  231          0343  2
;  232          0344  2 RETURN ss$_resignal;                           ! Resignal to get message
;  233          0345  1 END;                                          ! Of dump$handler


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

                                .PSECT  _LIB$KEY1$,NOWRT,  SHR,  PIC,1

                         00000 ;TPA$KEYSTO
                               U.2:     .BLKB   0
 54   52   41   54   53  00000 ;TPA$KEYST
                               U.4:     .ASCII  \START\                                 ;
                      FF 00005          .BYTE   -1                                      ;
                         00006 ;TPA$KEYSTO
                               U.8:     .BLKB   0
           44   4E   45  00006 ;TPA$KEYST
                               U.10:    .ASCII  \END\                                   ;
                      FF 00009          .BYTE   -1                                      ;
                         0000A ;TPA$KEYSTO
                               U.14:    .BLKB   0
 54   4E   55   4F   43  0000A ;TPA$KEYST
                               U.16:    .ASCII  \COUNT\                                 ;
                      FF 0000F          .BYTE   -1                                      ;
                      FF 00010 ;TPA$KEYFILL
                               U.20:    .BYTE   -1                                      ;

                                .PSECT  _LIB$STATE$,NOWRT,  SHR,  PIC,1

                         00000 BLKREC_STATES::
                                         .BLKB   0
                    6100 00000 ;TPA$TYPE
                               U.5:     .WORD   24832                                   ;
              00000000* 00002 ;TPA$ADDR
                               U.6:     .LONG   <<DUMP$GL_FLAGS-U.6>-4>                 ;
              10000000  00006 ;TPA$MASK
```

```
                        U.7:    .LONG   268435456                       ;
    6101  0000A ;TPA$TYPE
                        U.11:   .WORD   24833                           ;
00000000* 0000C ;TPA$ADDR
                        U.12:   .LONG   <<DUMP$GL_FLAGS-U.12>-4>        ;
20000000  00010 ;TPA$MASK
                        U.13:   .LONG   536870912                       ;
    6502  00014 ;TPA$TYPE
                        U.17:   .WORD   25858                           ;
00000000* 00016 ;TPA$ADDR
                        U.18:   .LONG   <<DUMP$GL_FLAGS-U.18>-4>        ;
40000000  0001A ;TPA$MASK
                        U.19:   .LONG   1073741824                      ;
    003D  0001E ;TPA$TYPE
                        U.21:   .WORD   61                              ;
    043A  00020 ;TPA$TYPE
                        U.22:   .WORD   1082                            ;
          00022 PARSE_NUMBER:
                        .BLKB   0
    91F3  00022 ;TPA$TYPE
                        U.23:   .WORD   -28173                          ;
00000000V 00024 ;TPA$ACTION
                        U.24:   .LONG   <<DUMP$STORE_NUM-U.24>-4>       ;
    0000* 00028 ;TPA$TARGET
                        U.26:   .WORD   <<U.25-U.26>-2>                 ;
    0425  0002A ;TPA$TYPE
                        U.27:   .WORD   1061                            ;
    0058  0002C ;TPA$TYPE
                        U.28:   .WORD   88                              ;
    104F  0002E ;TPA$TYPE
                        U.29:   .WORD   4175                            ;
    0000* 00030 ;TPA$TARGET
                        U.31:   .WORD   <<U.30-U.31>-2>                 ;
    1444  00032 ;TPA$TYPE
                        U.32:   .WORD   5188                            ;
    0000* 00034 ;TPA$TARGET
                        U.34:   .WORD   <<U.33-U.34>-2>                 ;
    95F5  00036 ;TPA$TYPE
                        U.35:   .WORD   -27147                          ;
00000000V 00038 ;TPA$ACTION
                        U.36:   .LONG   <<DUMP$STORE_NUM-U.36>-4>       ;
    0000* 0003C ;TPA$TARGET
                        U.37:   .WORD   <<U.25-U.37>-2>                 ;
          0003E ;OCTNUM
                        U.30:   .BLKB   0
    95F4  0003E ;TPA$TYPE
                        U.38:   .WORD   -27148                          ;
00000000V 00040 ;TPA$ACTION
                        U.39:   .LONG   <<DUMP$STORE_NUM-U.39>-4>       ;
    0000* 00044 ;TPA$TARGET
                        U.40:   .WORD   <<U.25-U.40>-2>                 ;
          00046 ;DECNUM
                        U.33:   .BLKB   0
    85F3  00046 ;TPA$TYPE
                        U.41:   .WORD   -31245                          ;
00000000V 00048 ;TPA$ACTION
                        U.42:   .LONG   <<DUMP$STORE_NUM-U.42>-4>       ;
```

```
                        0004C ;EOS
                              U.25:     .BLKB   0
                15F7    0004C ;TPA$TYPE
                              U.43:     .WORD   5623                                    ;
                FFFF    0004E ;TPA$TARGET
                              U.44:     .WORD   -1                                      ;
                        00050 NUMBER_STATES::
                                        .BLKB   0
                1DF8    00050 ;TPA$TYPE
                              U.46:     .WORD   7672                                    ;
                0000*   00052 ;TPA$SUBEXP
                              U.47:     .WORD   <<PARSE_NUMBER-U.47>-2>                 ;
                FFFF    00054 ;TPA$TARGET
                              U.48:     .WORD   -1                                      ;

                                        .PSECT  _LIB$KEY0$,NOWRT,  SHR,  PIC,1

                        00000 BLKREC_KEYS::
                                        .BLKB   0
                        00000 ;TPA$KEY0
                              U.1:      .BLKB   0
                0000*   00000 ;TPA$KEY
                              U.3:      .WORD   <U.2-U.1>                               ;
                0000*   00002 ;TPA$KEY
                              U.9:      .WORD   <U.8-U.1>                               ;
                0000*   00004 ;TPA$KEY
                              U.15:     .WORD   <U.14-U.1>                              ;
                        00006           .BLKB   2
                        00008 NUMBER_KEYS::
                                        .BLKB   0
                        00008 ;TPA$KEY0
                              U.45:     .BLKB   0

                                        .PSECT  $OWN$,NOEXE,2

            00000001    00000 EXIT_STATUS:
                                        .LONG   1                                       ;
                        00004 TPA_BLOCK:
                                        .BLKB   36

                                        .PSECT  $GLOBAL$,NOEXE,2

                        00000 DUMP$GL_IFAB::
                                        .BLKB   4
                        00004 DUMP$GL_INAM::
                                        .BLKB   4
                        00008 DUMP$GL_IRAB::
                                        .BLKB   68
                        0004C DUMP$GL_ORAB::
                                        .BLKB   68
                        00090 DUMP$GL_OFAB::
                                        .BLKB   80
                        000E0 DUMP$GL_ONAM::
                                        .BLKB   96
                        00140 DUMP$GL_ORSS::
                                        .BLKB   255
                        0023F           .BLKB   1
```

```
00240 DUMP$GL_IDESC::
                .BLKB   8
00248 DUMP$GL_ODESC::
                .BLKB   8
00250 DUMP$AB_OUTBUF::
                .BLKB   132
002D4 DUMP$GL_OUTDESC::
                .BLKB   8
002DC DUMP$GL_CHANNEL::
                .BLKB   4
002E0 DUMP$GL_WIDTH::
                .BLKB   4
002E4 DUMP$GL_LPP::
                .BLKB   4
002E8 DUMP$GL_BUFFER::
                .BLKB   8
002F0 DUMP$GL_FLAGS::
                .BLKB   4
002F4 DUMP$GL_START_QUAL::
                .BLKB   4
002F8 DUMP$GL_END_QUAL::
                .BLKB   4
002FC DUMP$GL_COUNT_QUAL::
                .BLKB   4
00300 DUMP$GL_NUMBER_QUAL::
                .BLKB   4
00304 DUMP$GL_NUMBER::
                .BLKB   4
00308 DUMP$GL_CUR_BLOCK::
                .BLKB   4
0030C DUMP$GL_MAX_BLOCK::
                .BLKB   4
00310 DUMP$GL_FILE_EFBLK::
                .BLKB   4
00314 DUMP$GL_FILE_HIBLK::
                .BLKB   4
00318 DUMP$GL_RECORD::
                .BLKB   4
0031C DUMP$GQ_TIME::
                .BLKB   8

                .EXTRN  CLI$GET_VALUE, CLI$PRESENT
                .EXTRN  DUMP$BLANK_LINE
                .EXTRN  DUMP$DUMP_FILE, DUMP$OUTPUT_GETMSG
                .EXTRN  LIB$FREE_VM, LIB$GET_VM
                .EXTRN  LIB$FIND_FILE, LIB$LP_LINES
                .EXTRN  LIB$TPARSE, STR$COPY_DX
                .EXTRN  DUMP$_FACILITY, DUMP$_BADRANGE
                .EXTRN  DUMP$_CONFQUAL, DUMP$_DEVQUALS
                .EXTRN  DUMP$_DEVSPEC, DUMP$_GETCHN
                .EXTRN  DUMP$_ENDOFFILE
                .EXTRN  DUMP$_NOVIRMEM, DUMP$_BADSTART

                .PSECT  $CODE$,NOWRT,2

0004 00000 DUMP$HANDLER:
                .WORD   Save R2                                         ; 0323
```

```
                         52 00000000'  EF  9E 00002          MOVAB   EXIT_STATUS, R2                                ;
                   50    04  AC        04  C1 00009          ADDL3   #4, SIGARGS, R0                               ; 0333
                               12        60  E8 0000E          BLBS    (R0), 2$                                      ; 0336
              51   62         03        00  EF 00011          EXTZV   #0, #3, EXIT_STATUS, R1                       ; 0338
              51   60         03        00  ED 00016          CMPZV   #0, #3, (R0), R1                              ;
                               03        1A  0001B            BGTRU   1$                                            ;
                               03        62  E9 0001D          BLBC    EXIT_STATUS, 2$                               ; 0339
                               62        60  D0 00020 1$:      MOVL    (R0), EXIT_STATUS                             ; 0341
                               50  0918  8F  3C 00023 2$:      MOVZWL  #2328, R0                                     ; 0344
                                         04  00028            RET                                                   ; 0345

; Routine Size:  41 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  235     0346  1 ROUTINE dump$start=
;  236     0347  2 BEGIN
;  237     0348  2 !
;  238     0349  2 ! This is the main program.  It gathers all the command inputs, and then
;  239     0350  2 ! dumps the requested files.
;  240     0351  2 !
;  241     0352  2 LOCAL
;  242     0353  2     find_context : REF BBLOCK,                  ! Context for lib$find_file
;  243     0354  2     find_result : BBLOCK[dsc$c_s_bln],          ! Result of lib$find_file
;  244     0355  2     find_related : BBLOCK[dsc$c_s_bln],         ! Related file for find_file
;  245     0356  2     find_default : BBLOCK[dsc$c_s_bln],         ! Default file
;  246     0357  2     value_desc : BBLOCK[dsc$c_s_bln],          ! Qualifier value descriptor
;  247     0358  2     status,                                    ! Status variable
;  248     0359  2     output_desc : BBLOCK[dsc$c_s_bln],         ! Output file specification
;  249     0360  2     input_desc : BBLOCK[dsc$c_s_bln],          ! Input file specification
;  250     0361  2     input_devchar : BLOCK[16,BYTE];            ! $GETDVI arg block
;  251     0362  2 BUILTIN
;  252     0363  2     fp;
;  253     0364  2
;  254     0365  2
;  255     0366  2 .fp = dump$handler;                            ! Enable the condition handler
;  256     0367  2
;  257     0368  2
;  258     0369  2 CH$FILL(0, dsc$c_s_bln, input_desc);
;  259     0370  2 input_desc[dsc$b_class] = dsc$k_class_d;       ! Make descriptors dynamic
;  260     0371  2 CH$MOVE(dsc$c_s_bln, input_desc, output_desc);
;  261     0372  2 CH$MOVE(dsc$c_s_bln, input_desc, find_result);
;  262     0373  2 CH$MOVE(dsc$c_s_bln, input_desc, find_related);
;  263     0374  2 CH$MOVE(dsc$c_s_bln, input_desc, find_default);
;  264     0375  2 CH$MOVE(dsc$c_s_bln, input_desc, value_desc);
;  265     0376  2
;  266     0377  2
;  267     0378  2 ! Get parameters and qualifiers.
;  268     0379  2 !
;  269     0380  2 cli$get_value($descriptor('INPUT'), input_desc);        ! Get input file spec
;  270     0381  2 cli$get_value($descriptor('OUTPUT'), output_desc);      ! Get output file spec
;  271     0382  2 dump$gl_flags[dump$v_allocated] =       cli$present($descriptor('ALLOCATED'));
;  272     0383  2                                         cli$present($descriptor('ASCII'));
;  273     0384  2 dump$gl_flags[dump$v_blocks] =          cli$present($descriptor('BLOCKS'));
;  274     0385  2 dump$gl_flags[dump$v_byte] =            cli$present($descriptor('BYTE'));
;  275     0386  2 dump$gl_flags[dump$v_decimal] =         cli$present($descriptor('DECIMAL'));
;  276     0387  2 dump$gl_flags[dump$v_file_header] =     cli$present($descriptor('FILE_HEADER'));
;  277     0388  2 dump$gl_flags[dump$v_formatted] =       cli$present($descriptor('FORMATTED'));
;  278     0389  2 dump$gl_flags[dump$v_header] =          cli$present($descriptor('HEADER'));
;  279     0390  2 dump$gl_flags[dump$v_hex] =             cli$present($descriptor('HEXADECIMAL'));
;  280     0391  2 dump$gl_flags[dump$v_longword] =       cli$present($descriptor('LONGWORD'));
;  281     0392  2 dump$gl_flags[dump$v_number] =         cli$present($descriptor('NUMBER'));
;  282     0393  2 dump$gl_flags[dump$v_octal] =          cli$present($descriptor('OCTAL'));
;  283     0394  2 dump$gl_flags[dump$v_output] =         cli$present($descriptor('OUTPUT'));
;  284     0395  2 dump$gl_flags[dump$v_printer] =        cli$present($descriptor('PRINTER'));
;  285     0396  2 dump$gl_flags[dump$v_records] =        cli$present($descriptor('RECORDS'));
;  286     0397  2 dump$gl_flags[dump$v_word] =           cli$present($descriptor('WORD'));
;  287     0398  2
;  288     0399  2
;  289     0400  2 ! If /NUMBER qualifier is present, get the value.
;  290     0401  2 !
;  291     0402  2 IF .dump$gl_flags[dump$v_number]                   ! /NUMBER present
```

```
;  292     0403  2 THEN
;  293     0404  3     BEGIN
;  294     0405  3     IF cli$get_value($descriptor('NUMBER'), value_desc)
;  295     0406  3     THEN
;  296     0407  4         BEGIN
;  297     0408  4         dump$gl_flags[dump$v_tpa_number] = true;        ! Note parsing /NUMBER
;  298     0409  4         IF NOT dump$tparse(value_desc, number_states, number_keys)
;  299     0410  4         THEN
;  300     0411  4             SIGNAL_STOP(
;  301     0412  4                 dump$_facility^16 + shr$_syntax + sts$k_severe,
;  302     0413  4                 1, value_desc);
;  303     0414  3         END;
;  304     0415  2     END;
;  305     0416  2
;  306     0417  2
;  307     0418  2 ! If /BLOCK qualifier is present, get the value(s).
;  308     0419  2 !
;  309     0420  2 IF .dump$gl_flags[dump$v_blocks]                        ! /BLOCKS present
;  310     0421  2 THEN
;  311     0422  3     BEGIN
;  312     0423  3     WHILE cli$get_value($descriptor('BLOCKS'), value_desc) DO
;  313     0424  4         BEGIN
;  314     0425  4         IF NOT dump$tparse(value_desc, blkrec_states, blkrec_keys)
;  315     0426  4         THEN
;  316     0427  4             SIGNAL_STOP(
;  317     0428  4                 dump$_facility^16 + shr$_syntax + sts$k_severe,
;  318     0429  4                 1, value_desc);
;  319     0430  3         END;
;  320     0431  2     END;
;  321     0432  2
;  322     0433  2
;  323     0434  2 ! If /RECORD qualifier is present, get the value(s).
;  324     0435  2 !
;  325     0436  2 IF .dump$gl_flags[dump$v_records]                       ! /RECORDS present
;  326     0437  2 THEN
;  327     0438  3     BEGIN
;  328     0439  3     WHILE cli$get_value($descriptor('RECORDS'), value_desc) DO
;  329     0440  4         BEGIN
;  330     0441  4         IF NOT dump$tparse(value_desc, blkrec_states, blkrec_keys)
;  331     0442  4         THEN
;  332     0443  4             SIGNAL_STOP(
;  333     0444  4                 dump$_facility^16 + shr$_syntax + sts$k_severe,
;  334     0445  4                 1, value_desc);
;  335     0446  3         END;
;  336     0447  2     END;
;  337     0448  2
;  338     0449  2
;  339     0450  2 ! Check range of START and END if both were specified, to ensure that START
;  340     0451  2 ! is less than END.
;  341     0452  2 !
;  342     0453  2 IF .dump$gl_flags[dump$v_start] AND .dump$gl_flags[dump$v_end]
;  343     0454  2 AND .dump$gl_start_qual GTRU .dump$gl_end_qual
;  344     0455  2 THEN
;  345     0456  2     SIGNAL_STOP(dump$_badrange);
;  346     0457  2
;  347     0458  2
;  348     0459  2 ! Get number of lines on output page.
```

```
;  349          0460  2 !
;  350          0461  2 dump$gl_lpp = lib$lp_lines() - 6;
;  351          0462  2
;  352          0463  2
;  353          0464  2 ! Loop, calling LIB$FIND_FILE to get files matching the input spec.
;  354          0465  2 !
;  355          0466  2 find_context = 0;                                       ! Initialize context
;  356          0467  2 UNTIL
;  357          0468  3     BEGIN
;  358          0469  3     status = lib$find_file(
;  359          0470  3         input_desc,
;  360          0471  3         find_result,
;  361          0472  3         find_context,
;  362          0473  3         find_default,
;  363          0474  3         find_related);
;  364          0475  3     IF .find_context NEQ 0 THEN dump$gl_inam = .find_context[fab$l_nam];
;  365          0476  3     IF .status EQL rms$_dnf OR .status EQL rms$_fnf
;  366          0477  3     THEN
;  367          0478  4         BEGIN
;  368          0479  5         IF (.dump$gl_inam[nam$l_fnb] AND                ! Check for only device
;  369          0480  6             (nam$m_exp_dir OR
;  370          0481  6              nam$m_exp_name OR
;  371          0482  6              nam$m_exp_type OR
;  372          0483  6              nam$m_exp_ver OR
;  373          0484  4              nam$m_wildcard)) EQL 0
;  374          0485  4         THEN
;  375          0486  5             BEGIN
;  376          0487  5             input_devchar[dumpdvi_w_size] = 4;          ! Build $GETDVI item
;  377          0488  5             input_devchar[dumpdvi_w_type] = dvi$_devchar;       ! list for the
;  378          0489  5             input_devchar[dumpdvi_l_addr] = find_context[fab$l_dev];    ! device
;  379          0490  5             input_devchar[dumpdvi_l_len] = 0;           ! characteristics
;  380          0491  5             input_devchar[dumpdvi_l_end] = 0;           ! Terminate the list
;  381        P 0492  5             $GETDVI(EFN = dumpdvi_c_efn,                ! Get characteristics
;  382        P 0493  5                     DEVNAM = input_desc,
;  383          0494  5                     ITMLST = input_devchar);
;  384          0495  5             $WAITFR(EFN = dumpdvi_c_efn);               ! Wait until complete
;  385          0496  5             status = 1;                                 ! Don't take an error
;  386          0497  5             BBLOCK[find_context[fab$l_dev], dev$v_for] = 1;     ! Mark foreign
;  387          0498  4             END;
;  388          0499  3         END;
;  389          0500  3     .status EQL rms$_nmf
;  390          0501  3     END
;  391          0502  2 DO
;  392          0503  3     BEGIN
;  393          0504  3     IF NOT .status                                      ! Report error
;  394          0505  3     THEN
;  395          0506  4         BEGIN
;  396          0507  4         SIGNAL(
;  397          0508  4             dump$_facility^16 + shr$_openin + sts$k_error,
;  398          0509  4             1, find_result,
;  399          0510  4             .find_context[fab$l_sts], .find_context[fab$l_stv]);
;  400          0511  4         END
;  401          0512  3     ELSE
;  402          0513  4         BEGIN
;  403          0514  4         IF dump$open_input(.find_context, find_result)
;  404          0515  4         AND dump$open_output(output_desc, .find_context)
;  405          0516  4         THEN
```

```
;  406     0517  5              BEGIN
;  407     0518  5              dump$list_width(dump$gl_ofab);              ! Get width of listing
;  408     0519  5              $GETTIM(timadr=dump$gq_time);               ! Get current time
;  409     0520  5              dump$gl_number = .dump$gl_number_qual;      ! Set initial /NUMBER
;  410     0521  5              dump$dump_file();                           ! Dump the file
;  411     0522  5              dump$close_input(.find_context);
;  412     0523  5              dump$close_output();
;  413     0524  4              END;
;  414     0525  4          str$copy_dx(find_related, find_result);         ! Propagate related
;  415     0526  3          END;
;  416     0527  3      IF NOT .dump$gl_inam[nam$v_wildcard]
;  417     0528  3      THEN RETURN .exit_status OR sts$m_inhib_msg;
;  418     0529  2      END;                                                ! Of LIB$FIND_FILE loop
;  419     0530  2
;  420     0531  2
;  421     0532  2  RETURN .exit_status OR sts$m_inhib_msg;                 ! Exit with no message
;  422     0533  1  END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                  54  55  50  4E  49  00000 P.AAB:  .ASCII  \INPUT\                          ;
                                                      00005         .BLKB   3                                 ;
                                    00000005          00008 P.AAA:  .LONG   5                                 ;
                                    00000000'         0000C         .ADDRESS P.AAB                            ;
                                  54  55  50  54  55  4F  00010 P.AAD:  .ASCII  \OUTPUT\                       ;
                                                      00016         .BLKB   2                                 ;
                                    00000006          00018 P.AAC:  .LONG   6                                 ;
                                    00000000'         0001C         .ADDRESS P.AAD                            ;
                              44  45  54  41  43  4F  4C  4C  41  00020 P.AAF:  .ASCII  \ALLOCATED\           ;
                                                      00029         .BLKB   3                                 ;
                                    00000009          0002C P.AAE:  .LONG   9                                 ;
                                    00000000'         00030         .ADDRESS P.AAF                            ;
                                          49  49  43  53  41  00034 P.AAH:  .ASCII  \ASCII\                   ;
                                                      00039         .BLKB   3                                 ;
                                    00000005          0003C P.AAG:  .LONG   5                                 ;
                                    00000000'         00040         .ADDRESS P.AAH                            ;
                                      53  4B  43  4F  4C  42  00044 P.AAJ:  .ASCII  \BLOCKS\                  ;
                                                      0004A         .BLKB   2                                 ;
                                    00000006          0004C P.AAI:  .LONG   6                                 ;
                                    00000000'         00050         .ADDRESS P.AAJ                            ;
                                          45  54  59  42  00054 P.AAL:  .ASCII  \BYTE\                        ;
                                    00000004          00058 P.AAK:  .LONG   4                                 ;
                                    00000000'         0005C         .ADDRESS P.AAL                            ;
                                4C  41  4D  49  43  45  44  00060 P.AAN:  .ASCII  \DECIMAL\                   ;
                                                      00067         .BLKB   1                                 ;
                                    00000007          00068 P.AAM:  .LONG   7                                 ;
                                    00000000'         0006C         .ADDRESS P.AAN                            ;
52  45  44  41  45  48  5F  45  4C  49  46  00070 P.AAP:  .ASCII  \FILE_HEADER\                               ;
                                                      0007B         .BLKB   1                                 ;
                                    0000000B          0007C P.AAO:  .LONG   11                                ;
                                    00000000'         00080         .ADDRESS P.AAP                            ;
                              44  45  54  54  41  4D  52  4F  46  00084 P.AAR:  .ASCII  \FORMATTED\           ;
                                                      0008D         .BLKB   3                                 ;
                                    00000009          00090 P.AAQ:  .LONG   9                                 ;
                                    00000000'         00094         .ADDRESS P.AAR                            ;
```

```
                                52 45 44 41 45 48  00098 P.AAT:  .ASCII  \HEADER\                                ;
                                                   0009E        .BLKB   2
                                   00000006        000A0 P.AAS: .LONG   6                                       ;
                                   00000000'       000A4        .ADDRESS P.AAT                                  ;
                4C 41 4D 49 43 45 44 41 58 45 48   000A8 P.AAV: .ASCII  \HEXADECIMAL\                           ;
                                                   000B3        .BLKB   1
                                   0000000B        000B4 P.AAU: .LONG   11                                      ;
                                   00000000'       000B8        .ADDRESS P.AAV                                  ;
                      44 52 4F 57 47 4E 4F 4C      000BC P.AAX: .ASCII  \LONGWORD\                              ;
                                   00000008        000C4 P.AAW: .LONG   8                                       ;
                                   00000000'       000C8        .ADDRESS P.AAX                                  ;
                                52 45 42 4D 55 4E  000CC P.AAZ: .ASCII  \NUMBER\                              ;
                                                   000D2        .BLKB   2
                                   00000006        000D4 P.AAY: .LONG   6                                       ;
                                   00000000'       000D8        .ADDRESS P.AAZ                                  ;
                                      4C 41 54 43 4F  000DC P.ABB: .ASCII  \OCTAL\                              ;
                                                   000E1        .BLKB   3
                                   00000005        000E4 P.ABA: .LONG   5                                       ;
                                   00000000'       000E8        .ADDRESS P.ABB                                  ;
                                54 55 50 54 55 4F  000EC P.ABD: .ASCII  \OUTPUT\                              ;
                                                   000F2        .BLKB   2
                                   00000006        000F4 P.ABC: .LONG   6                                       ;
                                   00000000'       000F8        .ADDRESS P.ABD                                  ;
                             52 45 54 4E 49 52 50  000FC P.ABF: .ASCII  \PRINTER\                             ;
                                                   00103        .BLKB   1
                                   00000007        00104 P.ABE: .LONG   7                                       ;
                                   00000000'       00108        .ADDRESS P.ABF                                  ;
                             53 44 52 4F 43 45 52  0010C P.ABH: .ASCII  \RECORDS\                             ;
                                                   00113        .BLKB   1
                                   00000007        00114 P.ABG: .LONG   7                                       ;
                                   00000000'       00118        .ADDRESS P.ABH                                  ;
                                      44 52 4F 57  0011C P.ABJ: .ASCII  \WORD\                                 ;
                                   00000004        00120 P.ABI: .LONG   4                                       ;
                                   00000000'       00124        .ADDRESS P.ABJ                                  ;
                                52 45 42 4D 55 4E  00128 P.ABL: .ASCII  \NUMBER\                              ;
                                                   0012E        .BLKB   2
                                   00000006        00130 P.ABK: .LONG   6                                       ;
                                   00000000'       00134        .ADDRESS P.ABL                                  ;
                                53 4B 43 4F 4C 42  00138 P.ABN: .ASCII  \BLOCKS\                              ;
                                                   0013E        .BLKB   2
                                   00000006        00140 P.ABM: .LONG   6                                       ;
                                   00000000'       00144        .ADDRESS P.ABN                                  ;
                             53 44 52 4F 43 45 52  00148 P.ABP: .ASCII  \RECORDS\                             ;
                                                   0014F        .BLKB   1
                                   00000007        00150 P.ABO: .LONG   7                                       ;
                                   00000000'       00154        .ADDRESS P.ABP                                  ;

                                                                .EXTRN  SYS$GETDVI, SYS$WAITFR
                                                                .EXTRN  SYS$GETTIM

                                                                .PSECT  $CODE$,NOWRT,2

                                                   0FFC 00000 DUMP$START:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 0346
                      5B 00000000'  EF  9E 00002                MOVAB   BLKREC_KEYS, R11                        ;
                      5A 00000000G  00  9E 00009                MOVAB   LIB$STOP, R10                           ;
                      59 00000000G  00  9E 00010                MOVAB   CLI$GET_VALUE, R9                       ;
```

```
                                58 00000000G   00   9E 00017         MOVAB   CLI$PRESENT, R8                                    ;
                                57 00000000'   EF   9E 0001E         MOVAB   P.AAA, R7                                        ;
                                56 00000000'   EF   9E 00025         MOVAB   DUMP$GL_FLAGS, R6                                ;
                                5E        BC   AE   9E 0002C         MOVAB   -68(SP), SP                                      ;
                                6D        A4   AF   9E 00030         MOVAB   DUMP$HANDLER, (FP)                               ; 0366
        08              00      6E             00   2C 00034         MOVC5   #0, (SP), #0, #8, INPUT_DESC                     ; 0369
                                           14   AE      00039                                                                 ;
                          17    AE             02   90 0003B         MOVB    #2, INPUT_DESC+3                                 ; 0370
                  1C    AE    14    AE         08   28 0003F         MOVC3   #8, INPUT_DESC, OUTPUT_DESC                      ; 0371
                  3C    AE    14    AE         08   28 00045         MOVC3   #8, INPUT_DESC, FIND_RESULT                      ; 0372
                  34    AE    14    AE         08   28 0004B         MOVC3   #8, INPUT_DESC, FIND_RELATED                     ; 0373
                  2C    AE    14    AE         08   28 00051         MOVC3   #8, INPUT_DESC, FIND_DEFAULT                     ; 0374
                  24    AE    14    AE         08   28 00057         MOVC3   #8, INPUT_DESC, VALUE_DESC                       ; 0375
                                           14   AE   9F 0005D         PUSHAB  INPUT_DESC                                       ; 0380
                                              57   DD 00060         PUSHL   R7                                               ;
                                69            02   FB 00062         CALLS   #2, CLI$GET_VALUE                                ;
                                           1C   AE   9F 00065         PUSHAB  OUTPUT_DESC                                      ; 0381
                                           10   A7   9F 00068         PUSHAB  P.AAC                                            ;
                                69            02   FB 0006B         CALLS   #2, CLI$GET_VALUE                                ;
                                           24   A7   9F 0006E         PUSHAB  P.AAE                                            ; 0382
                                68            01   FB 00071         CALLS   #1, CLI$PRESENT                                  ;
        66              01      00            50   F0 00074         INSV    R0, #0, #1, DUMP$GL_FLAGS                        ;
                                           34   A7   9F 00079         PUSHAB  P.AAG                                            ; 0383
                                68            01   FB 0007C         CALLS   #1, CLI$PRESENT                                  ;
                                           44   A7   9F 0007F         PUSHAB  P.AAI                                            ; 0384
                                68            01   FB 00082         CALLS   #1, CLI$PRESENT                                  ;
        66              01      01            50   F0 00085         INSV    R0, #1, #1, DUMP$GL_FLAGS                        ;
                                           50   A7   9F 0008A         PUSHAB  P.AAK                                            ; 0385
                                68            01   FB 0008D         CALLS   #1, CLI$PRESENT                                  ;
        66              01      02            50   F0 00090         INSV    R0, #2, #1, DUMP$GL_FLAGS                        ;
                                           60   A7   9F 00095         PUSHAB  P.AAM                                            ; 0386
                                68            01   FB 00098         CALLS   #1, CLI$PRESENT                                  ;
        66              01      03            50   F0 0009B         INSV    R0, #3, #1, DUMP$GL_FLAGS                        ;
                                           74   A7   9F 000A0         PUSHAB  P.AAO                                            ; 0387
                                68            01   FB 000A3         CALLS   #1, CLI$PRESENT                                  ;
        66              01      04            50   F0 000A6         INSV    R0, #4, #1, DUMP$GL_FLAGS                        ;
                                      0088    C7   9F 000AB         PUSHAB  P.AAQ                                            ; 0388
                                68            01   FB 000AF         CALLS   #1, CLI$PRESENT                                  ;
        66              01      05            50   F0 000B2         INSV    R0, #5, #1, DUMP$GL_FLAGS                        ;
                                      0098    C7   9F 000B7         PUSHAB  P.AAS                                            ; 0389
                                68            01   FB 000BB         CALLS   #1, CLI$PRESENT                                  ;
        66              01      06            50   F0 000BE         INSV    R0, #6, #1, DUMP$GL_FLAGS                        ;
                                      00AC    C7   9F 000C3         PUSHAB  P.AAU                                            ; 0390
                                68            01   FB 000C7         CALLS   #1, CLI$PRESENT                                  ;
        66              01      07            50   F0 000CA         INSV    R0, #7, #1, DUMP$GL_FLAGS                        ;
                                      00BC    C7   9F 000CF         PUSHAB  P.AAW                                            ; 0391
                                68            01   FB 000D3         CALLS   #1, CLI$PRESENT                                  ;
    01  A6              01      00            50   F0 000D6         INSV    R0, #0, #1, DUMP$GL_FLAGS+1                      ;
                                      00CC    C7   9F 000DC         PUSHAB  P.AAY                                            ; 0392
                                68            01   FB 000E0         CALLS   #1, CLI$PRESENT                                  ;
    01  A6              01      01            50   F0 000E3         INSV    R0, #1, #1, DUMP$GL_FLAGS+1                      ;
                                      00DC    C7   9F 000E9         PUSHAB  P.ABA                                            ; 0393
                                68            01   FB 000ED         CALLS   #1, CLI$PRESENT                                  ;
    01  A6              01      02            50   F0 000F0         INSV    R0, #2, #1, DUMP$GL_FLAGS+1                      ;
                                      00EC    C7   9F 000F6         PUSHAB  P.ABC                                            ; 0394
                                68            01   FB 000FA         CALLS   #1, CLI$PRESENT                                  ;
    01  A6              01      03            50   F0 000FD         INSV    R0, #3, #1, DUMP$GL_FLAGS+1                      ;
```

```
                                00FC   C7  9F 00103          PUSHAB  P.ABE                                    ; 0395
                            68         01  FB 00107          CALLS   #1, CLI$PRESENT                         ;
      01    A6          01      04     50  F0 0010A          INSV    R0, #4, #1, DUMP$GL_FLAGS+1             ;
                                010C   C7  9F 00110          PUSHAB  P.ABG                                   ; 0396
                            68         01  FB 00114          CALLS   #1, CLI$PRESENT                         ;
      01    A6          01      05     50  F0 00117          INSV    R0, #5, #1, DUMP$GL_FLAGS+1             ;
                                0118   C7  9F 0011D          PUSHAB  P.ABI                                   ; 0397
                            68         01  FB 00121          CALLS   #1, CLI$PRESENT                         ;
      01    A6          01      06     50  F0 00124          INSV    R0, #6, #1, DUMP$GL_FLAGS+1             ;
                36      01    A6       01  E1 0012A          BBC     #1, DUMP$GL_FLAGS+1, 1$                 ; 0402
                                  24     AE  9F 0012F          PUSHAB  VALUE_DESC                              ; 0405
                                0128   C7  9F 00132          PUSHAB  P.ABK                                   ;
                            69         02  FB 00136          CALLS   #2, CLI$GET_VALUE                       ;
                            29         50  E9 00139          BLBC    R0, 1$                                  ;
                        03    A6         80  8F 88 0013C      BISB2   #128, DUMP$GL_FLAGS+3                   ; 0408
                                    08     AB  9F 00141      PUSHAB  NUMBER_KEYS                             ; 0409
                                00000000'  EF  9F 00144      PUSHAB  NUMBER_STATES                           ;
                                    2C     AE  9F 0014A      PUSHAB  VALUE_DESC                              ;
                    00000000V   EF         03  FB 0014D      CALLS   #3, DUMP$TPARSE                         ;
                                0E         50  E8 00154      BLBS    R0, 1$                                  ;
                                  24     AE  9F 00157      PUSHAB  VALUE_DESC                              ; 0411
                                01     DD  9F 0015A      PUSHL   #1                                      ;
                                00000000*  8F  DD 0015C      PUSHL   #<<<DUMP$_FACILITY@16>+4344>+4>         ; 0412
                            6A         03  FB 00162          CALLS   #3, LIB$STOP                            ;
                32          66         01  E1 00165  1$:    BBC     #1, DUMP$GL_FLAGS, 3$                   ; 0420
                                  24     AE  9F 00169  2$:    PUSHAB  VALUE_DESC                              ; 0423
                                0138   C7  9F 0016C          PUSHAB  P.ABM                                   ;
                            69         02  FB 00170          CALLS   #2, CLI$GET_VALUE                       ;
                            25         50  E9 00173          BLBC    R0, 3$                                  ;
                                       5B  DD 00176          PUSHL   R11                                     ; 0425
                                00000000'  EF  9F 00178      PUSHAB  BLKREC_STATES                           ;
                                    2C     AE  9F 0017E      PUSHAB  VALUE_DESC                              ;
                    00000000V   EF         03  FB 00181      CALLS   #3, DUMP$TPARSE                         ;
                                DE         50  E8 00188      BLBS    R0, 2$                                  ;
                                  24     AE  9F 0018B      PUSHAB  VALUE_DESC                              ; 0427
                                01     DD  9F 0018E      PUSHL   #1                                      ;
                                00000000*  8F  DD 00190      PUSHL   #<<<DUMP$_FACILITY@16>+4344>+4>         ; 0428
                            6A         03  FB 00196          CALLS   #3, LIB$STOP                            ;
                                       CE  11 00199          BRB     2$                                      ; 0423
                32      01    A6       05  E1 0019B  3$:    BBC     #5, DUMP$GL_FLAGS+1, 5$                 ; 0436
                                  24     AE  9F 001A0  4$:    PUSHAB  VALUE_DESC                              ; 0439
                                0148   C7  9F 001A3          PUSHAB  P.ABO                                   ;
                            69         02  FB 001A7          CALLS   #2, CLI$GET_VALUE                       ;
                            25         50  E9 001AA          BLBC    R0, 5$                                  ;
                                       5B  DD 001AD          PUSHL   R11                                     ; 0441
                                00000000'  EF  9F 001AF      PUSHAB  BLKREC_STATES                           ;
                                    2C     AE  9F 001B5      PUSHAB  VALUE_DESC                              ;
                    00000000V   EF         03  FB 001B8      CALLS   #3, DUMP$TPARSE                         ;
                                DE         50  E8 001BF      BLBS    R0, 4$                                  ;
                                  24     AE  9F 001C2      PUSHAB  VALUE_DESC                              ; 0443
                                01     DD  9F 001C5      PUSHL   #1                                      ;
                                00000000*  8F  DD 001C7      PUSHL   #<<<DUMP$_FACILITY@16>+4344>+4>         ; 0444
                            6A         03  FB 001CD          CALLS   #3, LIB$STOP                            ;
                                       CE  11 001D0          BRB     4$                                      ; 0439
                                    01     A6  95 001D2  5$:    TSTB    DUMP$GL_FLAGS+1                         ; 0453
                                       14  18 001D5          BGEQ    6$                                      ;
                            10         02  A6  E9 001D7      BLBC    DUMP$GL_FLAGS+2, 6$                     ;
```

```
                08   A6        04   A6  D1 001DB          CMPL    DUMP$GL_START_QUAL, DUMP$GL_END_QUAL          ;0454
                                    09  1B 001E0          BLEQU   6$                                            ;
                          00000000G 8F  DD 001E2          PUSHL   #DUMP$_BADRANGE                               ;0456
                             6A     01  FB 001E8          CALLS   #1, LIB$STOP                                  ;
00000000G       00                  00  FB 001EB 6$:      CALLS   #0, LIB$LP_LINES                              ;0461
              F4   A6        FA     A0  9E 001F2          MOVAB   -6(R0), DUMP$GL_LPP                           ;
                                    6E  D4 001F7          CLRL    FIND_CONTEXT                                  ;0466
                               34   AE  9F 001F9 7$:      PUSHAB  FIND_RELATED                                  ;0469
                               30   AE  9F 001FC          PUSHAB  FIND_DEFAULT                                  ;
                               08   AE  9F 001FF          PUSHAB  FIND_CONTEXT                                  ;
                               48   AE  9F 00202          PUSHAB  FIND_RESULT                                   ;
                               24   AE  9F 00205          PUSHAB  INPUT_DESC                                    ;
00000000G       00                  05  FB 00208          CALLS   #5, LIB$FIND_FILE                             ;
                53                  50  D0 0020F          MOVL    R0, STATUS                                    ;
                52                  6E  D0 00212          MOVL    FIND_CONTEXT, R2                              ;0475
                                    06  13 00215          BEQL    8$                                            ;
              FD14   C6        28   A2  D0 00217          MOVL    40(R2), DUMP$GL_INAM                          ;
0001C04A        8F                  53  D1 0021D 8$:      CMPL    STATUS, #114762                               ;0476
                                    09  13 00224          BEQL    9$                                            ;
00018292        8F                  53  D1 00226          CMPL    STATUS, #98962                                ;
                                    41  12 0022D          BNEQ    10$                                           ;
                50         FD14     C6  D0 0022F 9$:      MOVL    DUMP$GL_INAM, R0                              ;0479
            0147   8F        34     A0  B3 00234          BITW    52(R0), #327                                  ;0484
                                    34  12 0023A          BNEQ    10$                                           ;
                04   AE 00020004     8F  D0 0023C          MOVL    #131076, INPUT_DEVCHAR                        ;0487
                08   AE        40   A2  9E 00244          MOVAB   64(R2), INPUT_DEVCHAR+4                       ;0489
                                0C  AE  7C 00249          CLRQ    INPUT_DEVCHAR+8                               ;0490
                                    7E  7C 0024C          CLRQ    -(SP)                                         ;0494
                                    7E  7C 0024E          CLRQ    -(SP)                                         ;
                               14   AE  9F 00250          PUSHAB  INPUT_DEVCHAR                                 ;
                               28   AE  9F 00253          PUSHAB  INPUT_DESC                                    ;
                             7E         03  7D 00256          MOVQ    #3, -(SP)                                     ;
00000000G       00                  08  FB 00259          CALLS   #8, SYS$GETDVI                                ;
                                    03  DD 00260          PUSHL   #3                                            ;0495
00000000G       00                  01  FB 00262          CALLS   #1, SYS$WAITFR                                ;
                53                  01  D0 00269          MOVL    #1, STATUS                                    ;0496
              43   A2                01  88 0026C          BISB2   #1, 67(R2)                                    ;0497
000182CA        8F                  53  D1 00270 10$:      CMPL    STATUS, #99018                                ;0500
                                    03  12 00277          BNEQ    11$                                           ;
                                0083    31 00279          BRW     15$                                           ;
                18                  53  E8 0027C 11$:      BLBS    STATUS, 12$                                   ;0504
                7E         08       A2  7D 0027F          MOVQ    8(R2), -(SP)                                  ;0510
                           44   AE  9F 00283          PUSHAB  FIND_RESULT                                   ;0507
                                    01  DD 00286          PUSHL   #1                                            ;
                      00000000*     8F  DD 00288          PUSHL   #<<<DUMP$_FACILITY@16>+4248>+2>               ;0508
00000000G       00                  05  FB 0028E          CALLS   #5, LIB$SIGNAL                                ;
                                    5C  11 00295          BRB     14$                                           ;0504
                           3C   AE  9F 00297 12$:      PUSHAB  FIND_RESULT                                   ;0514
                                    52  DD 0029A          PUSHL   R2                                            ;
00000000V       EF                  02  FB 0029C          CALLS   #2, DUMP$OPEN_INPUT                           ;
                40                  50  E9 002A3          BLBC    R0, 13$                                       ;
                                    52  DD 002A6          PUSHL   R2                                            ;0515
                           20   AE  9F 002A8          PUSHAB  OUTPUT_DESC                                   ;
00000000V       EF                  02  FB 002AB          CALLS   #2, DUMP$OPEN_OUTPUT                          ;
                31                  50  E9 002B2          BLBC    R0, 13$                                       ;
                      FDA0     C6  9F 002B5          PUSHAB  DUMP$GL_OFAB                                  ;0518
00000000V       EF                  01  FB 002B9          CALLS   #1, DUMP$LIST_WIDTH                           ;
```

```
                                 2C    A6  9F 002C0          PUSHAB  DUMP$GQ_TIME                                          ; 0519
                00000000G  00          01  FB 002C3          CALLS   #1, SYS$GETTIM                                        ;
                      14   A6        10  A6  D0 002CA          MOVL    DUMP$GL_NUMBER_QUAL, DUMP$GL_NUMBER                   ; 0520
                00000000G  00          00  FB 002CF          CALLS   #0, DUMP$DUMP_FILE                                    ; 0521
                                       52  DD 002D6          PUSHL   R2                                                    ; 0522
                00000000V  EF          01  FB 002D8          CALLS   #1, DUMP$CLOSE_INPUT                                  ;
                00000000V  EF          00  FB 002DF          CALLS   #0, DUMP$CLOSE_OUTPUT                                 ; 0523
                                 3C    AE  9F 002E6 13$:     PUSHAB  FIND_RESULT                                           ; 0525
                                 38    AE  9F 002E9          PUSHAB  FIND_RELATED                                          ;
                00000000G  00          02  FB 002EC          CALLS   #2, STR$COPY_DX                                       ;
                           50    FD14  C6  D0 002F3 14$:     MOVL    DUMP$GL_INAM, R0                                      ; 0527
                           03    35    A0  E9 002F8          BLBC    53(R0), 15$                                           ;
                                       FEFA 31 002FC          BRW     7$                                                    ;
              50 00000000'  EF 10000000  8F  C9 002FF 15$:     BISL3   #268435456, EXIT_STATUS, R0                           ; 0532
                                          04 0030B          RET                                                           ; 0533

; Routine Size:  780 bytes,    Routine Base:  $CODE$ + 0029
```

```
;  424          0534  1 ROUTINE dump$tparse(string, states, keys)=
;  425          0535  2 BEGIN
;  426          0536  2 !
;  427          0537  2 ! This routine calls TPARSE given the string, states and keys.
;  428          0538  2 !
;  429          0539  2 ! Inputs:
;  430          0540  2 !
;  431          0541  2 !       string          address of descriptor for string
;  432          0542  2 !       states          address of TPARSE states table
;  433          0543  2 !       keys            address of TPARSE keys table
;  434          0544  2 !
;  435          0545  2 MAP
;  436          0546  2     string : REF BBLOCK;
;  437          0547  2
;  438          0548  2
;  439          0549  2 CH$FILL(0, tpa$k_length0, tpa_block);                   ! Initialize block
;  440          0550  2 tpa_block[tpa$l_count] = tpa$k_count0;
;  441          0551  2 tpa_block[tpa$l_options] = tpa$m_abbrev;
;  442          0552  2 tpa_block[tpa$l_stringcnt] = .string[dsc$w_length];
;  443          0553  2 tpa_block[tpa$l_stringptr] = .string[dsc$a_pointer];
;  444          0554  2 RETURN lib$tparse(tpa_block, .states, .keys);           ! Call TPARSE
;  445          0555  1 END;


                                007C 00000 DUMP$TPARSE:
                                           .WORD   Save R2,R3,R4,R5,R6                      ; 0534
                     56 00000000'  EF  9E 00002    MOVAB   TPA_BLOCK, R6                    ;
           24        00            6E  00 2C 00009  MOVC5   #0, (SP), #0, #36, TPA_BLOCK     ; 0549
                                       66 0000E    ;
                     66            08  D0 0000F    MOVL    #8, TPA_BLOCK                    ; 0550
                  04 A6            02  D0 00012    MOVL    #2, TPA_BLOCK+4                  ; 0551
                     50        04  AC  D0 00016    MOVL    STRING, R0                       ; 0552
                  08 A6            60  3C 0001A    MOVZWL  (R0), TPA_BLOCK+8                ;
                  0C A6        04  A0  D0 0001E    MOVL    4(R0), TPA_BLOCK+12              ; 0553
                     7E        08  AC  7D 00023    MOVQ    STATES, -(SP)                    ; 0554
                               56  DD 00027    PUSHL   R6                               ;
           00000000G 00            03  FB 00029  CALLS   #3, LIB$TPARSE                   ;
                                       04 00030    RET                                      ; 0555

; Routine Size:  49 bytes,    Routine Base:  $CODE$ + 0335
```

```
;  447     0556  1 ROUTINE dump$store_num=
;  448     0557  2 BEGIN
;  449     0558  2 !
;  450     0559  2 ! This routine is called when the /BLOCKS, /RECORDS, or /NUMBER qualifier
;  451     0560  2 ! is used.  It interrogates flags set by TPARSE to determine which numeric
;  452     0561  2 ! value has been parsed and stores it in the appropriate result cell.
;  453     0562  2 !
;  454     0563  2 IF .dump$gl_flags[dump$v_tpa_start]              ! Parsing START
;  455     0564  2 THEN
;  456     0565  3     BEGIN
;  457     0566  3     dump$gl_start_qual = .tpa_block[tpa$l_number];
;  458     0567  3     dump$gl_flags[dump$v_start] = true;          ! Note START was present
;  459     0568  3     END
;  460     0569  2 ELSE IF .dump$gl_flags[dump$v_tpa_end]           ! Parsing END
;  461     0570  2 THEN
;  462     0571  3     BEGIN
;  463     0572  3     dump$gl_end_qual = .tpa_block[tpa$l_number];
;  464     0573  3     dump$gl_flags[dump$v_end] = true;            ! Note END was present
;  465     0574  3     END
;  466     0575  2 ELSE IF .dump$gl_flags[dump$v_tpa_count]         ! Parsing COUNT
;  467     0576  2 THEN
;  468     0577  3     BEGIN
;  469     0578  3     dump$gl_count_qual = .tpa_block[tpa$l_number];
;  470     0579  3     dump$gl_flags[dump$v_count] = true;          ! Note COUNT was present
;  471     0580  3     END
;  472     0581  2 ELSE IF .dump$gl_flags[dump$v_tpa_number]        ! Parsing NUMBER
;  473     0582  2 THEN
;  474     0583  2     dump$gl_number_qual = .tpa_block[tpa$l_number]
;  475     0584  2 ELSE
;  476     0585  2     SIGNAL_STOP(dump$_facility^16 + shr$_badlogic + sts$k_severe);
;  477     0586  2
;  478     0587  2
;  479     0588  2 dump$gl_flags[dump$v_tpa_start] = false;         ! Clear flags for next call
;  480     0589  2 dump$gl_flags[dump$v_tpa_end] = false;
;  481     0590  2 dump$gl_flags[dump$v_tpa_count] = false;
;  482     0591  2 dump$gl_flags[dump$v_tpa_number] = false;
;  483     0592  2
;  484     0593  2
;  485     0594  2 RETURN true                                      ! Return success to TPARSE
;  486     0595  1 END;


                                000C 00000 DUMP$STORE_NUM:
                                           .WORD   Save R2,R3                                          ; 0556
                     53 00000000'  EF  9E 00002    MOVAB   TPA_BLOCK+28, R3                         ;
                     52 00000000'  EF  9E 00009    MOVAB   DUMP$GL_FLAGS, R2                        ;
          0B   03    A2            04  E1 00010    BBC     #4, DUMP$GL_FLAGS+3, 1$                  ; 0563
               04    A2            63  D0 00015    MOVL    TPA_BLOCK+28, DUMP$GL_START_QUAL         ; 0566
               01    A2      80    8F  88 00019    BISB2   #128, DUMP$GL_FLAGS+1                    ; 0567
                                   36  11 0001E    BRB     5$                                       ; 0563
          0A   03    A2            05  E1 00020 1$: BBC     #5, DUMP$GL_FLAGS+3, 2$                  ; 0569
               08    A2            63  D0 00025    MOVL    TPA_BLOCK+28, DUMP$GL_END_QUAL           ; 0572
               02    A2            01  88 00029    BISB2   #1, DUMP$GL_FLAGS+2                      ; 0573
                                   27  11 0002D    BRB     5$                                       ; 0569
```

```
                     0A       03   A2              06  E1 0002F 2$:    BBC     #6, DUMP$GL_FLAGS+3, 3$                          ; 0575
                              0C   A2              63  D0 00034        MOVL    TPA_BLOCK+28, DUMP$GL_COUNT_QUAL                 ; 0578
                              02   A2              02  88 00038        BISB2   #2, DUMP$GL_FLAGS+2                              ; 0579
                                                   18  11 0003C        BRB     5$                                               ; 0575
                                             03    A2  95 0003E 3$:     TSTB    DUMP$GL_FLAGS+3                                  ; 0581
                                                   06  18 00041        BGEQ    4$                                               ;
                              10   A2              63  D0 00043        MOVL    TPA_BLOCK+28, DUMP$GL_NUMBER_QUAL                ; 0583
                                                   0D  11 00047        BRB     5$                                               ;
                                   00000000*       8F  DD 00049 4$:     PUSHL   #<<<DUMP$_FACILITY@16>+4384>+4>                  ; 0585
                           00000000G   00          01  FB 0004F        CALLS   #1, LIB$STOP                                     ;
                              03   A2        F0    8F  8A 00056 5$:     BICB2   #240, DUMP$GL_FLAGS+3                            ; 0591
                                   50              01  D0 0005B        MOVL    #1, R0                                           ; 0594
                                                   04  0005E        RET                                                      ; 0595

; Routine Size:  95 bytes,    Routine Base:  $CODE$ + 0366
```

```
;  488          0596  1 ROUTINE dump$open_input(fab, filedesc)=
;  489          0597  2 BEGIN
;  490          0598  2 !
;  491          0599  2 ! This routine opens the input file
;  492          0600  2 !
;  493          0601  2 ! Inputs:
;  494          0602  2 !
;  495          0603  2 !       fab     pointer to an already initialized fab complete with NAM block
;  496          0604  2 !       filedesc pointer to string descriptor of resultant string from $parse
;  497          0605  2 !
;  498          0606  2 ! Outputs:
;  499          0607  2 !
;  500          0608  2 !       File is opened
;  501          0609  2 !
;  502          0610  2 ! Routine value:
;  503          0611  2 !
;  504          0612  2 !       true    successful
;  505          0613  2 !       false   error, signal already done
;  506          0614  2 !
;  507          0615  2 MAP
;  508          0616  2     fab : REF BBLOCK,
;  509          0617  2     filedesc : REF BBLOCK;
;  510          0618  2 LOCAL
;  511          0619  2     ixab : $XABFHC_DECL,
;  512          0620  2     dib : BBLOCK[dib$c_length],
;  513          0621  2     dibdesc : BBLOCK[dsc$c_s_bln],
;  514          0622  2     status;
;  515          0623  2
;  516          0624  2
;  517          0625  2 dump$gl_ifab = .fab;                                  ! Set pointer to FAB
;  518          0626  2 dump$gl_inam = .fab[fab$l_nam];                       ! and NAM block
;  519          0627  2
;  520          0628  2 fab[fab$b_shr]=fab$m_get OR fab$m_put OR fab$m_upi; ! Open file shared.
;  521          0629  2
;  522          0630  2 IF .BBLOCK[fab[fab$l_dev], dev$v_net]                ! If network device
;  523          0631  2 THEN
;  524          0632  2     BEGIN
;  525          0633  2     IF .dump$gl_flags[dump$v_allocated]               ! Ensure no conflicting
;  526          0634  2     OR .dump$gl_flags[dump$v_blocks]                  ! qualifiers
;  527          0635  2     OR .dump$gl_flags[dump$v_header]
;  528          0636  2     THEN
;  529          0637  2         SIGNAL_STOP(dump$_devquals);
;  530          0638  2
;  531          0639  2
;  532          0640  2     dump$gl_flags[dump$v_records] = true;             ! Force record mode
;  533          0641  2     END;
;  534          0642  2
;  535          0643  2
;  536          0644  2 IF .BBLOCK[fab[fab$l_dev], dev$v_for]                ! If foreign device
;  537          0645  2 OR (NOT .BBLOCK[fab[fab$l_dev], dev$v_fod]           ! or not disk, tape, or network
;  538          0646  2 AND NOT .BBLOCK[fab[fab$l_dev], dev$v_net])
;  539          0647  2 THEN
;  540          0648  2     BEGIN
;  541          0649  2     IF .dump$gl_flags[dump$v_allocated]               ! Ensure no file-oriented
;  542          0650  2     OR .dump$gl_flags[dump$v_records]                 ! qualifiers
;  543          0651  2     OR .dump$gl_flags[dump$v_header]
;  544          0652  3     THEN
```

```
;  545          0653  3          SIGNAL_STOP(dump$_devquals);
;  546          0654  3
;  547          0655  3
;  548          0656  3      IF (.dump$gl_inam[nam$l_fnb] AND            ! Ensure nothing except device
;  549          0657  3          (nam$m_exp_dir OR
;  550          0658  3           nam$m_exp_name OR
;  551          0659  3           nam$m_exp_type OR
;  552          0660  3           nam$m_exp_ver OR
;  553          0661  3           nam$m_wildcard)) NEQ 0
;  554          0662  3      THEN
;  555          0663  3          SIGNAL_STOP(dump$_devspec);
;  556          0664  3      END
;  557          0665  2  ELSE
;  558          0666  3      BEGIN
;  559          0667  3      IF NOT .BBLOCK[fab[fab$l_dev], dev$v_rnd]   ! Ensure no disk-oriented
;  560          0668  3      AND .dump$gl_flags[dump$v_allocated]        ! qualifiers on tape
;  561          0669  3      THEN
;  562          0670  3          SIGNAL_STOP(dump$_devquals);
;  563          0671  3
;  564          0672  3
;  565        P 0673  3      $XABFHC_INIT(xab=ixab,                      ! Initialize XAB
;  566          0674  3          nxt=0);
;  567          0675  3      fab[fab$l_xab] = ixab;                      ! Set pointer to XAB
;  568          0676  2      END;
;  569          0677  2
;  570          0678  2
;  571          0679  2  IF NOT .dump$gl_flags[dump$v_records]
;  572          0680  2  THEN
;  573          0681  2      fab[fab$v_ufo] = true                       ! Open file only
;  574          0682  2  ELSE
;  575          0683  2      fab[fab$v_get] = fab[fab$v_sqo] = true;     ! Allow GETs, sequential op
;  576          0684  2
;  577          0685  2
;  578          0686  2  IF NOT .BBLOCK[fab[fab$l_dev], dev$v_for]       ! Do OPEN if not foreign
;  579          0687  2  THEN
;  580          0688  3      BEGIN
;  581          0689  4      IF NOT $OPEN(fab=.fab)                      ! Open the input file
;  582          0690  3      THEN
;  583          0691  4          BEGIN
;  584          0692  4          dump$file_error(
;  585          0693  4              dump$_facility^16 + shr$_openin + sts$k_error,
;  586          0694  4              .fab,
;  587          0695  4              .fab[fab$l_sts], .fab[fab$l_stv]);
;  588          0696  4          fab[fab$l_xab] = 0;
;  589          0697  4          RETURN false;
;  590          0698  3          END;
;  591          0699  2      END;
;  592          0700  2
;  593          0701  2  fab[fab$l_xab] = 0;
;  594          0702  2
;  595          0703  2  IF .BBLOCK[fab[fab$l_dev], dev$v_for]           ! If foreign device
;  596          0704  3  OR (NOT .BBLOCK[fab[fab$l_dev], dev$v_fod]      ! or not disk, tape, or network
;  597          0705  3  AND NOT .BBLOCK[fab[fab$l_dev], dev$v_net])
;  598          0706  2  THEN
;  599          0707  3      BEGIN
;  600          0708  3      dump$gl_idesc[dsc$w_length] = .dump$gl_inam[nam$b_dev];     ! Prune to
;  601          0709  3      dump$gl_idesc[dsc$a_pointer] = .dump$gl_inam[nam$l_dev];    ! device only
```

```
;   602        P 0710  3     status = $ASSIGN(DEVNAM = dump$gl_idesc,     ! Do ASSIGN if foreign
;   603          0711  3                     CHAN = fab[fab$l_stv]);
;   604          0712  3     IF NOT .status THEN SIGNAL_STOP(.status);
;   605          0713  3     END
;   606          0714  2 ELSE
;   607          0715  3     BEGIN
;   608          0716  3     dump$gl_idesc[dsc$w_length] = .dump$gl_inam[nam$b_rsl];
;   609          0717  3     dump$gl_idesc[dsc$a_pointer] = .dump$gl_inam[nam$[_rsa];
;   610          0718  2     END;
;   611          0719  2
;   612          0720  2
;   613          0721  2 IF NOT .dump$gl_flags[dump$v_records]
;   614          0722  2 THEN
;   615          0723  2     dump$gl_channel = .fab[fab$l_stv]           ! Save the channel
;   616          0724  2 ELSE
;   617          0725  3     BEGIN
;   618        P 0726  3     $RAB_INIT(rab=dump$gl_irab,                 ! Initialize input RAB
;   619          0727  3         fab=.fab);
;   620          0728  4     IF NOT $CONNECT(rab=dump$gl_irab)           ! Connect RAB
;   621          0729  3     THEN
;   622          0730  4         BEGIN
;   623          0731  4         dump$file_error(
;   624          0732  4             dump$_facility^16 + shr$_openin + sts$k_error,
;   625          0733  4             .fab,
;   626          0734  4             .dump$gl_irab[rab$l_sts], .dump$gl_irab[rab$l_stv]);
;   627          0735  4         RETURN false;
;   628          0736  3         END;
;   629          0737  2     END;
;   630          0738  2
;   631          0739  2
;   632          0740  2 dump$gl_cur_block = 1;
;   633          0741  2 dump$gl_max_block = -1;
;   634          0742  2
;   635          0743  2 IF .BBLOCK[fab[fab$l_dev], dev$v_rnd]             ! Disk device
;   636          0744  2 THEN
;   637          0745  2     IF .BBLOCK[fab[fab$l_dev], dev$v_for]          ! If foreign disk
;   638          0746  2     THEN
;   639          0747  3         BEGIN
;   640          0748  3         dibdesc[dsc$w_length] = dib$c_length;      ! Set up to get device
;   641          0749  3         dibdesc[dsc$a_pointer] = dib;              ! characteristics
;   642          0750  3         status = $GETCHN(CHAN=.dump$gl_channel, PRIBUF=dibdesc);
;   643          0751  3         IF NOT .status
;   644          0752  3         THEN
;   645          0753  3             SIGNAL_STOP(dump$_getchn, 0, .status);
;   646          0754  3         dump$gl_cur_block = 0;
;   647          0755  3         dump$gl_max_block = .dib[dib$l_maxblock] - 1;
;   648          0756  3         END
;   649          0757  2     ELSE
;   650          0758  3         BEGIN                                      ! Files-11 disk
;   651          0759  3
;   652          0760  3         ! Save FHC information for page heading.
;   653          0761  3         !
;   654          0762  3         dump$gl_file_efblk = .ixab[xab$l_ebk];
;   655          0763  3         IF .dump$gl_file_efblk NEQ 0 AND .ixab[xab$w_ffb] EQL 0
;   656          0764  3         THEN
;   657          0765  3             dump$gl_file_efblk = .dump$gl_file_efblk - 1;
;   658          0766  3         dump$gl_file_hiblk = .fab[fab$l_alq];
```

```
;  659          0767  3          IF NOT .dump$gl_flags[dump$v_records]           ! Not record mode
;  660          0768  3          THEN
;  661          0769  4              BEGIN
;  662          0770  4              dump$gl_max_block = .dump$gl_file_efblk;
;  663          0771  4              IF .dump$gl_flags[dump$v_allocated]
;  664          0772  4                  THEN dump$gl_max_block = .dump$gl_file_hiblk;
;  665          0773  3              END;
;  666          0774  2          END;
;  667          0775  2
;  668          0776  2      IF .dump$gl_flags[dump$v_start]
;  669          0777  2      THEN
;  670          0778  2          dump$gl_cur_block = MAXU(.dump$gl_cur_block, .dump$gl_start_qual);
;  671          0779  2
;  672          0780  2      IF .BBLOCK[fab[fab$l_dev], dev$v_for]
;  673          0781  2      AND .dump$gl_cur_block GTRU .dump$gl_max_block
;  674          0782  2      THEN SIGNAL_STOP(dump$_badstart, 1, .dump$gl_max_block);
;  675          0783  2
;  676          0784  2      IF .dump$gl_flags[dump$v_end]
;  677          0785  2      THEN
;  678          0786  2          dump$gl_max_block = MINU(.dump$gl_max_block, .dump$gl_end_qual);
;  679          0787  2
;  680          0788  2      IF .dump$gl_flags[dump$v_count]
;  681          0789  2      THEN
;  682          0790  2          IF .dump$gl_flags[dump$v_start]
;  683          0791  2          THEN
;  684          0792  2              dump$gl_max_block = MINU(.dump$gl_max_block,
;  685          0793  2                  .dump$gl_start_qual + .dump$gl_count_qual - 1)
;  686          0794  2          ELSE
;  687          0795  2              dump$gl_max_block = MINU(.dump$gl_max_block,
;  688          0796  2                  .dump$gl_cur_block + .dump$gl_count_qual - 1);
;  689          0797  2
;  690          0798  2
;  691          0799  2      dump$gl_record = 0;
;  692          0800  2      IF NOT .dump$gl_flags[dump$v_records]
;  693          0801  2      AND .BBLOCK[fab[fab$l_dev], dev$v_rnd]
;  694          0802  2      THEN
;  695          0803  2          dump$gl_record = .dump$gl_cur_block - 1;
;  696          0804  2
;  697          0805  2
;  698          0806  2      ! Allocate input buffer.
;  699          0807  2      !
;  700          0808  2      IF .dump$gl_flags[dump$v_records]                   ! If record dump
;  701          0809  2      THEN
;  702          0810  2          dump$gl_buffer[dsc$w_length] = dump$c_rmsbufsz  ! Largest RMS record
;  703          0811  2      ELSE
;  704          0812  2          IF .BBLOCK[fab[fab$l_dev], dev$v_sqd]
;  705          0813  2          THEN
;  706          0814  2              dump$gl_buffer[dsc$w_length] = dump$c_tapbufsz  ! Largest tape QIO
;  707          0815  2          ELSE
;  708          0816  2              dump$gl_buffer[dsc$w_length] = dump$c_qiobufsz; ! Largest non-tape QIO
;  709          0817  2
;  710          0818  2
;  711          0819  2      status = lib$get_vm(                                ! Get memory
;  712          0820  2          dump$gl_buffer[dsc$w_length],
;  713          0821  2          dump$gl_buffer[dsc$a_pointer]);
;  714          0822  2      IF NOT .status THEN SIGNAL_STOP(dump$_novirmem, 0, .status);
;  715          0823  2
```

```
;  716          0824  2
;  717          0825  2 dump$gl_irab[rab$w_usz] = .dump$gl_buffer[dsc$w_length];
;  718          0826  2 dump$gl_irab[rab$l_ubf] = .dump$gl_buffer[dsc$a_pointer];
;  719          0827  2 RETURN true
;  720          0828  1 END;


                                                $RMS_PTR=           DUMP$GL_IRAB
                                                        .EXTRN  SYS$OPEN, SYS$ASSIGN
                                                        .EXTRN  SYS$CONNECT, SYS$GETCHN

                                            0FFC 00000 DUMP$OPEN_INPUT:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 0596
                                 5B 00000000G   8F  D0 00002        MOVL    #DUMP$_DEVQUALS, R11            ;
                                 5A 00000000G   00  9E 00009        MOVAB   LIB$STOP, R10                   ;
                                 59 00000000'   EF  9E 00010        MOVAB   DUMP$GL_FLAGS, R9               ;
                                 5E     FF58    CE  9E 00017        MOVAB   -168(SP), SP                    ;
                                 56       04    AC  D0 0001C        MOVL    FAB, R6                         ; 0625
                           FD10  C9             56  D0 00020        MOVL    R6, DUMP$GL_IFAB                ;
                           FD14  C9       28    A6  D0 00025        MOVL    40(R6), DUMP$GL_INAM            ; 0626
                             17  A6       43    8F  90 0002B        MOVB    #67, 23(R6)                     ; 0628
                                 57       40    A6  9E 00030        MOVAB   64(R6), R7                      ; 0630
                        14       67             0D  E1 00034        BBC     #13, (R7), 3$                   ;
                                 08             69  E8 00038        BLBS    DUMP$GL_FLAGS, 1$               ; 0633
                        04       69             01  E0 0003B        BBS     #1, DUMP$GL_FLAGS, 1$           ; 0634
                        05       69             06  E1 0003F        BBC     #6, DUMP$GL_FLAGS, 2$           ; 0635
                                                5B  DD 00043 1$:    PUSHL   R11                             ; 0637
                                 6A             01  FB 00045        CALLS   #1, LIB$STOP                    ;
                                       01  A9   20  88 00048 2$:    BISB2   #32, DUMP$GL_FLAGS+1            ; 0640
                                 08       03    A7  E8 0004C 3$:    BLBS    3(R7), 4$                       ; 0644
                        2D       67             0E  E0 00050        BBS     #14, (R7), 7$                   ; 0645
                        29       67             0D  E0 00054        BBS     #13, (R7), 7$                   ; 0646
                                 09             69  E8 00058 4$:    BLBS    DUMP$GL_FLAGS, 5$               ; 0649
                        04       01  A9         05  E0 0005B        BBS     #5, DUMP$GL_FLAGS+1, 5$         ; 0650
                        05       69             06  E1 00060        BBC     #6, DUMP$GL_FLAGS, 6$           ; 0651
                                                5B  DD 00064 5$:    PUSHL   R11                             ; 0653
                                 6A             01  FB 00066        CALLS   #1, LIB$STOP                    ;
                                 50     FD14    C9  D0 00069 6$:    MOVL    DUMP$GL_INAM, R0                ; 0656
                              0147  8F       34  A0  B3 0006E        BITW    52(R0), #327                    ; 0661
                                                29  13 00074        BEQL    9$                              ;
                                    00000000G   8F  DD 00076        PUSHL   #DUMP$_DEVSPEC                  ; 0663
                                 6A             01  FB 0007C        CALLS   #1, LIB$STOP                    ;
                                                1E  11 0007F        BRB     9$                              ; 0644
                        08       67             1C  E0 00081 7$:    BBS     #28, (R7), 8$                   ; 0667
                                 05             69  E9 00085        BLBC    DUMP$GL_FLAGS, 8$               ; 0668
                                                5B  DD 00088        PUSHL   R11                             ; 0670
                                 6A             01  FB 0008A        CALLS   #1, LIB$STOP                    ;
                    2C          00             00  2C 0008D 8$:    MOVC5   #0, (SP), #0, #44, $RMS_PTR     ; 0674
                                          7C    AE      00092                                               ;
                                 7C  AE   2C1D  8F  B0 00094        MOVW    #11293, $RMS_PTR                ;
                                 24  A6     7C  AE  9E 0009A        MOVAB   IXAB, 36(R6)                    ; 0675
                        06       01  A9         05  E0 0009F 9$:    BBS     #5, DUMP$GL_FLAGS+1, 10$        ; 0679
                                 06  A6         02  88 000A4        BISB2   #2, 6(R6)                       ; 0681
                                                09  11 000A8        BRB     11$                             ;
                                 04  A6     40  8F  88 000AA 10$:   BISB2   #64, 4(R6)                     ; 0683
                                 16  A6         02  88 000AF        BISB2   #2, 22(R6)                      ;
```

```
                                 25        03    A7  E8 000B3 11$:    BLBS    3(R7), 12$                                          ; 0686
                                                56  DD 000B7         PUSHL   R6                                                  ; 0689
                          00000000G      00      01  FB 000B9         CALLS   #1, SYS$OPEN
                                         19      50  E8 000C0         BLBS    R0, 12$
                                         7E    08    A6  7D 000C3     MOVQ    8(R6), -(SP)                                        ; 0695
                                                56  DD 000C7         PUSHL   R6                                                  ; 0694
                                 00000000*     8F  DD 000C9         PUSHL   #<<<DUMP$_FACILITY@16>+4248>+2>                     ; 0693
                          00000000V      EF      04  FB 000CF         CALLS   #4, DUMP$FILE_ERROR
                                               24    A6  D4 000D6     CLRL    36(R6)                                              ; 0696
                                           01D8      31  000D9         BRW     37$                                                 ; 0697
                                               24    A6  D4 000DC 12$: CLRL    36(R6)                                              ; 0701
                                         50  FD14    C9  D0 000DF     MOVL    DUMP$GL_INAM, R0                                    ; 0708
                                         08    03    A7  E8 000E4     BLBS    3(R7), 13$                                          ; 0703
                                 2D      67          0E  E0 000E8     BBS     #14, (R7), 14$                                      ; 0704
                                 29      67          0D  E0 000EC     BBS     #13, (R7), 14$                                      ; 0705
                                   FF50  C9    39    A0  9B 000F0 13$: MOVZBW  57(R0), DUMP$GL_IDESC                               ; 0708
                                   FF54  C9    44    A0  D0 000F6     MOVL    68(R0), DUMP$GL_IDESC+4                             ; 0709
                                                7E  7C 000FC         CLRQ    -(SP)                                               ; 0711
                                               0C    A6  9F 000FE     PUSHAB  12(R6)
                                           FF50    C9  9F 00101     PUSHAB  DUMP$GL_IDESC
                          00000000G      00      04  FB 00105         CALLS   #4, SYS$ASSIGN
                                         58      50  D0 0010C         MOVL    R0, STATUS
                                         13      58  E8 0010F         BLBS    STATUS, 15$                                         ; 0712
                                                58  DD 00112         PUSHL   STATUS
                                         6A      01  FB 00114         CALLS   #1, LIB$STOP
                                                0C  11 00117         BRB     15$                                                 ; 0703
                                   FF50  C9    03    A0  9B 00119 14$: MOVZBW  3(R0), DUMP$GL_IDESC                                ; 0716
                                   FF54  C9    04    A0  D0 0011F     MOVL    4(R0), DUMP$GL_IDESC+4                              ; 0717
                                 07      01    A9    05  E0 00125 15$: BBS     #5, DUMP$GL_FLAGS+1, 16$                            ; 0721
                                           EC  A9    0C    A6  D0 0012A MOVL    12(R6), DUMP$GL_CHANNEL                             ; 0723
                                                3B  11 0012F         BRB     17$
0044    8F                       00             6E  00  2C 00131 16$: MOVC5   #0, (SP), #0, #68, $RMS_PTR                         ; 0727
                                           FD18    C9      00138
                                   FD18  C9  4401    8F  B0 0013B     MOVW    #17409, $RMS_PTR                                    ;
                                   FD54  C9          56  D0 00142     MOVL    R6, $RMS_PTR+60                                     ;
                                           FD18    C9  9F 00147         PUSHAB  DUMP$GL_IRAB                                        ; 0728
                          00000000G      00      01  FB 0014B         CALLS   #1, SYS$CONNECT
                                         17      50  E8 00152         BLBS    R0, 17$
                                         7E    FD20    C9  7D 00155     MOVQ    DUMP$GL_IRAB+8, -(SP)                               ; 0734
                                                56  DD 0015A         PUSHL   R6                                                  ; 0733
                                 00000000*     8F  DD 0015C         PUSHL   #<<<DUMP$_FACILITY@16>+4248>+2>                     ; 0732
                          00000000V      EF      04  FB 00162         CALLS   #4, DUMP$FILE_ERROR
                                           0148      31  00169         BRW     37$                                                 ; 0735
                                           18  A9          01  D0 0016C 17$: MOVL    #1, DUMP$GL_CUR_BLOCK                       ; 0740
                                           1C  A9          01  CE 00170     MNEGL   #1, DUMP$GL_MAX_BLOCK                       ; 0741
                                 62        67          1C  E1 00174     BBC     #28, (R7), 21$                                      ; 0743
                                         38    03    A7  E9 00178     BLBC    3(R7), 19$                                          ; 0745
                                         6E    74    8F  9B 0017C     MOVZBW  #116, DIBDESC                                       ; 0748
                                   04    AE    08    AE  9E 00180     MOVAB   DIB, DIBDESC+4                                      ; 0749
                                                7E  7C 00185         CLRQ    -(SP)                                               ; 0750
                                               08    AE  9F 00187     PUSHAB  DIBDESC
                                                7E  D4 0018A         CLRL    -(SP)
                                               EC    A9  DD 0018C     PUSHL   DUMP$GL_CHANNEL
                          00000000G      00      05  FB 0018F         CALLS   #5, SYS$GETCHN
                                         58      50  D0 00196         MOVL    R0, STATUS
                                         0D      58  E8 00199         BLBS    STATUS, 18$                                         ; 0751
                                                58  DD 0019C         PUSHL   STATUS                                              ; 0753
```

```
                                      7E  D4 0019E         CLRL    -(SP)                                               ;
                           00000000G  8F  DD 001A0         PUSHL   #DUMP$_GETCHN                                       ;
                      6A              03  FB 001A6         CALLS   #3, LIB$STOP                                        ;
                                 18   A9  D4 001A9  18$:   CLRL    DUMP$GL_CUR_BLOCK                                   ; 0754
 1C    A9        78   AE              01  C3 001AC         SUBL3   #1, DIB+112, DUMP$GL_MAX_BLOCK                      ; 0755
                                      26  11 001B2         BRB     21$                                                 ; 0745
                 20   A9         E4   AD  D0 001B4  19$:   MOVL    IXAB+16, DUMP$GL_FILE_EFBLK                         ; 0762
                                      08  13 001B9         BEQL    20$                                                 ; 0763
                                 E8   AD  B5 001BB         TSTW    IXAB+20                                             ;
                                      03  12 001BE         BNEQ    20$                                                 ;
                                 20   A9  D7 001C0         DECL    DUMP$GL_FILE_EFBLK                                  ; 0765
                 24   A9         10   A6  D0 001C3  20$:   MOVL    16(R6), DUMP$GL_FILE_HIBLK                          ; 0766
       0D        01   A9              05  E0 001C8         BBS     #5, DUMP$GL_FLAGS+1, 21$                            ; 0767
                 1C   A9         20   A9  D0 001CD         MOVL    DUMP$GL_FILE_EFBLK, DUMP$GL_MAX_BLOCK               ; 0770
                 05                   69  E9 001D2         BLBC    DUMP$GL_FLAGS, 21$                                  ; 0771
                 1C   A9         24   A9  D0 001D5         MOVL    DUMP$GL_FILE_HIBLK, DUMP$GL_MAX_BLOCK               ; 0772
                                 01   A9  95 001DA  21$:   TSTB    DUMP$GL_FLAGS+1                                     ; 0776
                                      12  18 001DD         BGEQ    23$                                                 ;
                 50              18   A9  D0 001DF         MOVL    DUMP$GL_CUR_BLOCK, R0                               ; 0778
                 04   A9              50  D1 001E3         CMPL    R0, DUMP$GL_START_QUAL                              ;
                                      04  1E 001E7         BGEQU   22$                                                 ;
                 50              04   A9  D0 001E9         MOVL    DUMP$GL_START_QUAL, R0                              ;
                 18   A9              50  D0 001ED  22$:   MOVL    R0, DUMP$GL_CUR_BLOCK                               ;
                 15              03   A7  E9 001F1  23$:   BLBC    3(R7), 24$                                          ; 0780
                 1C   A9         18   A9  D1 001F5         CMPL    DUMP$GL_CUR_BLOCK, DUMP$GL_MAX_BLOCK                ; 0781
                                      0E  1B 001FA         BLEQU   24$                                                 ;
                                 1C   A9  DD 001FC         PUSHL   DUMP$GL_MAX_BLOCK                                   ; 0782
                                      01  DD 001FF         PUSHL   #1                                                  ;
                           00000000G  8F  DD 00201         PUSHL   #DUMP$_BADSTART                                     ;
                      6A              03  FB 00207         CALLS   #3, LIB$STOP                                        ;
                      12         02   A9  E9 0020A  24$:   BLBC    DUMP$GL_FLAGS+2, 26$                                ; 0784
                      50         1C   A9  D0 0020E         MOVL    DUMP$GL_MAX_BLOCK, R0                               ; 0786
                 08   A9              50  D1 00212         CMPL    R0, DUMP$GL_END_QUAL                                ;
                                      04  1B 00216         BLEQU   25$                                                 ;
                 50              08   A9  D0 00218         MOVL    DUMP$GL_END_QUAL, R0                                ;
                 1C   A9              50  D0 0021C  25$:   MOVL    R0, DUMP$GL_MAX_BLOCK                               ;
       30        02   A9              01  E1 00220  26$:   BBC     #1, DUMP$GL_FLAGS+2, 30$                            ; 0788
                                 01   A9  95 00225         TSTB    DUMP$GL_FLAGS+1                                     ; 0790
                                      13  18 00228         BGEQ    27$                                                 ;
       51        04   A9         0C   A9  C1 0022A         ADDL3   DUMP$GL_COUNT_QUAL, DUMP$GL_START_QUAL, R1          ; 0793
                                      51  D7 00230         DECL    R1                                                  ;
                 50              1C   A9  D0 00232         MOVL    DUMP$GL_MAX_BLOCK, R0                               ;
                 51                   50  D1 00236         CMPL    R0, R1                                              ;
                                      13  1A 00239         BGTRU   28$                                                 ;
                                      14  11 0023B         BRB     29$                                                 ; 0792
       51        18   A9         0C   A9  C1 0023D  27$:   ADDL3   DUMP$GL_COUNT_QUAL, DUMP$GL_CUR_BLOCK, R1           ; 0796
                                      51  D7 00243         DECL    R1                                                  ;
                 50              1C   A9  D0 00245         MOVL    DUMP$GL_MAX_BLOCK, R0                               ;
                 51                   50  D1 00249         CMPL    R0, R1                                              ;
                                      03  1B 0024C         BLEQU   29$                                                 ;
                 50                   51  D0 0024E  28$:   MOVL    R1, R0                                              ;
                 1C   A9              50  D0 00251  29$:   MOVL    R0, DUMP$GL_MAX_BLOCK                               ; 0795
                                 28   A9  D4 00255  30$:   CLRL    DUMP$GL_RECORD                                      ; 0799
       0F        01   A9              05  E0 00258         BBS     #5, DUMP$GL_FLAGS+1, 32$                            ; 0800
       06             67              1C  E1 0025D         BBC     #28, (R7), 31$                                      ; 0801
 28    A9        18   A9              01  C3 00261         SUBL3   #1, DUMP$GL_CUR_BLOCK, DUMP$GL_RECORD               ; 0803
       08        01   A9              05  E1 00267  31$:   BBC     #5, DUMP$GL_FLAGS+1, 33$                            ; 0808
```

```
                        F8   A9     7FFF    8F   B0 0026C 32$:   MOVW    #32767, DUMP$GL_BUFFER                          ; 0810
                                            10   11 00272         BRB     35$                                             ;
                    06          67          05   E1 00274 33$:   BBC     #5, (R7), 34$                                   ; 0812
                        F8   A9             01   AE 00278         MNEGW   #1, DUMP$GL_BUFFER                              ; 0814
                                            06   11 0027C         BRB     35$                                             ;
                        F8   A9     0200    8F   B0 0027E 34$:   MOVW    #512, DUMP$GL_BUFFER                            ; 0816
                                      FC    A9   9F 00284 35$:   PUSHAB  DUMP$GL_BUFFER+4                                ; 0821
                                      F8    A9   9F 00287         PUSHAB  DUMP$GL_BUFFER                                  ; 0820
                00000000G   00            02   FB 0028A         CALLS   #2, LIB$GET_VM                                  ;
                            58            50   D0 00291         MOVL    R0, STATUS                                      ;
                            0D            58   E8 00294         BLBS    STATUS, 36$                                     ; 0822
                                          58   DD 00297         PUSHL   STATUS                                          ;
                                          7E   D4 00299         CLRL    -(SP)                                           ;
                                00000000G 8F   DD 0029B         PUSHL   #DUMP$_NOVIRMEM                                 ;
                            6A            03   FB 002A1         CALLS   #3, LIB$STOP                                    ;
                    FD38    C9      F8    A9   B0 002A4 36$:   MOVW    DUMP$GL_BUFFER, DUMP$GL_IRAB+32                 ; 0825
                    FD3C    C9      FC    A9   D0 002AA         MOVL    DUMP$GL_BUFFER+4, DUMP$GL_IRAB+36               ; 0826
                            50            01   D0 002B0         MOVL    #1, R0                                          ; 0827
                                                 04 002B3         RET                                                     ;
                                          50   D4 002B4 37$:   CLRL    R0                                              ; 0828
                                               04 002B6         RET                                                     ;

; Routine Size:  695 bytes,    Routine Base:  $CODE$ + 03C5
```

```
;  722          0829  1 ROUTINE dump$open_output(output_desc, ifab)=
;  723          0830  2 BEGIN
;  724          0831  2 !
;  725          0832  2 ! Open output file
;  726          0833  2 !
;  727          0834  2 ! Inputs:
;  728          0835  2 !
;  729          0836  2 !       output_desc     pointer to string descriptor for output file
;  730          0837  2 !       ifab            pointer to input fab
;  731          0838  2 !
;  732          0839  2 MAP
;  733          0840  2     output_desc : REF BBLOCK,
;  734          0841  2     ifab : REF BBLOCK;
;  735          0842  2
;  736          0843  2
;  737        P 0844  2 $FAB_INIT(fab=dump$gl_ofab,                       ! Initialize output FAB
;  738        P 0845  2     dna=UPLIT BYTE('.DMP'),                       ! Default /OUTPUT type
;  739        P 0846  2     dns=%CHARCOUNT('.DMP'),
;  740        P 0847  2     nam=dump$gl_onam,
;  741        P 0848  2     fop=<ofp,sqo>,
;  742        P 0849  2     rat=cr,
;  743          0850  2     fac=put);
;  744          0851  2
;  745          0852  2
;  746        P 0853  2 $NAM_INIT(nam=dump$gl_onam,                       ! Initialize output NAM block
;  747        P 0854  2     rlf=.ifab[fab$l_nam],
;  748        P 0855  2     rss=nam$c_maxrss,
;  749        P 0856  2     rsa=dump$gl_orss,
;  750        P 0857  2     ess=nam$c_maxrss,
;  751          0858  2     esa=dump$gl_orss);
;  752          0859  2
;  753          0860  2
;  754        P 0861  2 $RAB_INIT(rab=dump$gl_orab,                       ! Initialize output RAB
;  755          0862  2     fab=dump$gl_ofab);
;  756          0863  2
;  757          0864  2
;  758          0865  2 ! Create the output file and connect record stream.
;  759          0866  2 !
;  760          0867  2 IF .dump$gl_flags[dump$v_printer]                 ! If /PRINTER requested,
;  761          0868  2 THEN
;  762          0869  3     BEGIN
;  763          0870  3     dump$gl_ofab[fab$v_spl] = true;               ! Spool listing
;  764          0871  3     dump$gl_ofab[fab$v_dlt] = true;               ! Delete after printing
;  765          0872  3     END
;  766          0873  2 ELSE
;  767          0874  2     IF .dump$gl_flags[dump$v_output]              ! If /OUTPUT requested
;  768          0875  2     THEN
;  769          0876  3         BEGIN
;  770          0877  3         IF .output_desc[dsc$w_length] NEQ 0       ! If /OUTPUT has a value
;  771          0878  3         THEN
;  772          0879  4             BEGIN
;  773          0880  4             dump$gl_ofab[fab$l_fna] = .output_desc[dsc$a_pointer];
;  774          0881  4             dump$gl_ofab[fab$b_fns] = .output_desc[dsc$w_length];
;  775          0882  4             END
;  776          0883  3         END
;  777          0884  2     ELSE
;  778          0885  3         BEGIN                                     ! Else, default to SYS$OUTPUT
```

```
;  779          0886  3          dump$gl_ofab[fab$l_fna] = UPLIT BYTE('SYS$OUTPUT');
;  780          0887  3          dump$gl_ofab[fab$b_fns] = %CHARCOUNT('SYS$OUTPUT');
;  781          0888  3          END;
;  782          0889
;  783          0890
;  784          0891  2      IF NOT $CREATE(fab=dump$gl_ofab)
;  785          0892  2      THEN
;  786          0893  2          BEGIN
;  787          0894  2          dump$file_error(
;  788          0895  2              dump$_facility^16 + shr$_openout + sts$k_error,
;  789          0896  2              dump$gl_ofab,
;  790          0897  2              .dump$gl_ofab[fab$l_sts], .dump$gl_ofab[fab$l_stv]);
;  791          0898  2          RETURN false;
;  792          0899  2          END;
;  793          0900
;  794          0901  2      IF NOT $CONNECT(rab=dump$gl_orab)
;  795          0902  2      THEN
;  796          0903  2          BEGIN
;  797          0904  2          dump$file_error(
;  798          0905  2              dump$_facility^16 + shr$_openout + sts$k_error,
;  799          0906  2              dump$gl_ofab,
;  800          0907  2              .dump$gl_orab[rab$l_sts], .dump$gl_orab[rab$l_stv]);
;  801          0908  2          RETURN false;
;  802          0909  2          END;
;  803          0910  2
;  804          0911  2
;  805          0912  2      dump$gl_odesc[dsc$w_length] = .dump$gl_onam[nam$b_rsl];
;  806          0913  2      dump$gl_odesc[dsc$a_pointer] = .dump$gl_onam[nam$l_rsa];
;  807          0914  2      RETURN true
;  808          0915  1      END;


                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

                                        50  4D  44  2E  00158 P.ABQ:  .ASCII  \.DMP\
                54  55  50  54  55  4F  24  53  59  53  0015C P.ABR:  .ASCII  \SYS$OUTPUT\                                    ;

                                                    $RMS_PTR=           DUMP$GL_OFAB
                                                    $RMS_PTR=           DUMP$GL_ONAM
                                                    $RMS_PTR=           DUMP$GL_ORAB
                                                    .EXTRN  SYS$CREATE

                                                    .PSECT  $CODE$,NOWRT,2

                                            007C 00000 DUMP$OPEN_OUTPUT:
                                                    .WORD   Save R2,R3,R4,R5,R6                                             ; 0829
                                    56 00000000'    EF  9E 00002        MOVAB   $RMS_PTR, R6                                ;
0050    8F          00                  6E          00  2C 00009        MOVC5   #0, (SP), #0, #80, $RMS_PTR                 ; 0850
                                                    66     00010        ;
                                    66      5003    8F  B0 00011        MOVW    #20483, $RMS_PTR                            ;
                            04      A6 20000040     8F  D0 00016        MOVL    #536870976, $RMS_PTR+4                      ;
                            16      A6              01  90 0001E        MOVB    #1, $RMS_PTR+22                             ;
                            1E      A6      0202    8F  B0 00022        MOVW    #514, $RMS_PTR+30                           ;
                            28      A6        50    A6  9E 00028        MOVAB   DUMP$GL_ONAM, $RMS_PTR+40                   ;
                            30      A6 00000000'    EF  9E 0002D        MOVAB   P.ABQ, $RMS_PTR+48                          ;
                            35      A6              04  90 00035        MOVB    #4, $RMS_PTR+53                             ;
```

```
0060  8F        00           6E   00  2C 00039          MOVC5   #0, (SP), #0, #96, $RMS_PTR                    ; 0858
                            50   A6     00040
          50    A6    6002        8F  B0 00042          MOVW    #24578, $RMS_PTR                               ;
          52    A6                01  8E 00048          MNEGB   #1, $RMS_PTR+2                                 ;
          54    A6    00B0        C6  9E 0004C          MOVAB   DUMP$GL_ORSS, $RMS_PTR+4                       ;
          5A    A6                01  8E 00052          MNEGB   #1, $RMS_PTR+10                                ;
          5C    A6    00B0        C6  9E 00056          MOVAB   DUMP$GL_ORSS, $RMS_PTR+12                      ;
          50           08         AC  D0 0005C          MOVL    IFAB, R0                                       ;
          60    A6     28         A0  D0 00060          MOVL    40(R0), $RMS_PTR+16                            ;
0044  8F        00           6E   00  2C 00065          MOVC5   #0, (SP), #0, #68, $RMS_PTR                    ; 0862
                            BC   A6     0006C
          BC    A6    4401        8F  B0 0006E          MOVW    #17409, $RMS_PTR                               ;
          F8    A6                66  9E 00074          MOVAB   DUMP$GL_OFAB, $RMS_PTR+60                      ;
    07  0261    C6                04  E1 00078          BBC     #4, DUMP$GL_FLAGS+1, 1$                        ; 0867
          05    A6     A0         8F  88 0007E          BISB2   #160, DUMP$GL_OFAB+5                           ; 0871
                                  25  11 00083          BRB     3$                                             ; 0867
    13  0261    C6                03  E1 00085 1$:      BBC     #3, DUMP$GL_FLAGS+1, 2$                        ; 0874
                50     04         AC  D0 0008B          MOVL    OUTPUT_DESC, R0                                ; 0877
                                  60  B5 0008F          TSTW    (R0)                                           ;
                                  17  13 00091          BEQL    3$                                             ;
          2C    A6     04         A0  D0 00093          MOVL    4(R0), DUMP$GL_OFAB+44                         ; 0880
          34    A6                60  90 00098          MOVB    (R0), DUMP$GL_OFAB+52                          ; 0881
                                  0C  11 0009C          BRB     3$                                             ; 0876
          2C    A6 00000000'      EF  9E 0009E 2$:      MOVAB   P.ABR, DUMP$GL_OFAB+44                         ; 0886
          34    A6                0A  90 000A6          MOVB    #10, DUMP$GL_OFAB+52                           ; 0887
                                  56  DD 000AA 3$:      PUSHL   R6                                             ; 0891
     00000000G  00                01  FB 000AC          CALLS   #1, SYS$CREATE                                 ;
                06                50  E8 000B3          BLBS    R0, 4$                                         ;
                7E     08         A6  7D 000B6          MOVQ    DUMP$GL_OFAB+8, -(SP)                          ; 0897
                                  11  11 000BA          BRB     5$                                             ; 0894
                       BC         A6  9F 000BC 4$:      PUSHAB  DUMP$GL_ORAB                                   ; 0901
     00000000G  00                01  FB 000BF          CALLS   #1, SYS$CONNECT                                ;
                15                50  E8 000C6          BLBS    R0, 6$                                         ;
                7E     C4         A6  7D 000C9          MOVQ    DUMP$GL_ORAB+8, -(SP)                          ; 0907
                                  56  DD 000CD 5$:      PUSHL   R6                                             ; 0904
                  00000000*       8F  DD 000CF          PUSHL   #<<<DUMP$_FACILITY@16>+4256>+2>                ; 0905
     00000000V  EF                04  FB 000D5          CALLS   #4, DUMP$FILE_ERROR                            ;
                                  10  11 000DC          BRB     7$                                             ; 0908
        01B8    C6     53         A6  9B 000DE 6$:      MOVZBW  DUMP$GL_ONAM+3, DUMP$GL_ODESC                  ; 0912
        01BC    C6     54         A6  D0 000E4          MOVL    DUMP$GL_ONAM+4, DUMP$GL_ODESC+4                ; 0913
                50                01  D0 000EA          MOVL    #1, R0                                         ; 0914
                                  04     000ED          RET                                                    ;
                            50    D4     000EE 7$:      CLRL    R0                                             ; 0915
                                  04     000F0          RET                                                    ;

; Routine Size:  241 bytes,    Routine Base:  $CODE$ + 067C
```

```
;   810          0916  1 GLOBAL ROUTINE dump$read(bufdesc)=
;   811          0917  2 BEGIN
;   812          0918  2 !
;   813          0919  2 ! This routine reads from the input file.
;   814          0920  2 !
;   815          0921  2 MAP
;   816          0922  2     bufdesc : REF BBLOCK;
;   817          0923  2 LOCAL
;   818          0924  2     iosb : VECTOR[4,WORD],
;   819          0925  2     status;
;   820          0926  2
;   821          0927  2
;   822          0928  2 IF NOT .dump$gl_flags[dump$v_records]                    ! If reading with QIO
;   823          0929  2 THEN
;   824          0930  3     BEGIN
;   825          0931  3     IF NOT .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_rnd]
;   826          0932  3     THEN
;   827          0933  4         BEGIN                                           ! One QIO = one block
;   828          0934  4         DECR i FROM 1 TO 0 DO
;   829          0935  5             BEGIN
;   830        P 0936  5             status = $QIOW(
;   831        P 0937  5                 CHAN=.dump$gl_channel,
;   832        P 0938  5                 FUNC=(IF .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_for]
;   833        P 0939  5                     THEN io$_readlblk
;   834        P 0940  5                     ELSE io$_readvblk),
;   835        P 0941  5                 IOSB=iosb,
;   836        P 0942  5                 P1=.dump$gl_buffer[dsc$a_pointer],
;   837          0943  5                 P2=.dump$gl_buffer[dsc$w_length]);
;   838          0944  5             bufdesc[dsc$w_length] = .iosb[1];   ! Bytes actually read
;   839          0945  5             bufdesc[dsc$a_pointer] = .dump$gl_buffer[dsc$a_pointer];
;   840          0946  5             IF .status THEN status = .iosb[0];
;   841          0947  5
;   842          0948  5             IF .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_trm]      ! Handle ^Z
;   843          0949  5             AND .iosb[2] EQL %O'032'                            ! from terminal
;   844          0950  5             THEN
;   845          0951  5                 status = ss$_endoffile;
;   846          0952  5
;   847          0953  5             IF .status EQL ss$_endoffile        ! Print message if end of file
;   848          0954  5             AND .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_sqd]
;   849          0955  5             AND .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_for]
;   850          0956  5             THEN
;   851          0957  6                 BEGIN
;   852          0958  6                 dump$blank_line();
;   853          0959  6                 dump$output_getmsg(dump$_endoffile, %B'0001');
;   854          0960  6                 IF .i EQL 0 THEN EXITLOOP;
;   855          0961  6                 END
;   856          0962  5             ELSE
;   857          0963  5                 EXITLOOP;
;   858          0964  4             END;
;   859          0965  4         END
;   860          0966  3     ELSE
;   861          0967  4         BEGIN
;   862          0968  4         IF .dump$gl_cur_block GTRU .dump$gl_max_block
;   863          0969  4         THEN
;   864          0970  4             RETURN ss$_endoffile;                       ! Return EOF status
;   865        P 0971  4         status = $QIOW(
;   866        P 0972  4             CHAN=.dump$gl_channel,
```

```
;  867        P 0973  4              FUNC=(IF .BBLOCK[dump$gl_ifab[fab$l_dev], dev$v_for]
;  868        P 0974  4                  THEN io$_readlblk
;  869        P 0975  4                  ELSE io$_readvblk),
;  870        P 0976  4              !OSB=iosb,
;  871        P 0977  4              P1=.dump$gl_buffer[dsc$a_pointer],
;  872        P 0978  4              P2=512,
;  873          0979  4              P3=.dump$gl_cur_block);
;  874          0980  4          IF .status THEN status = .iosb[0];
;  875          0981  4          bufdesc[dsc$w_length] = 512;
;  876          0982  4          bufdesc[dsc$a_pointer] = .dump$gl_buffer[dsc$a_pointer];
;  877          0983  4          dump$gl_cur_block = .dump$gl_cur_block + 1;     ! Advance pointer
;  878          0984  3          END;
;  879          0985  3      IF NOT .status
;  880          0986  3      AND .status NEQ ss$_endoffile
;  881          0987  3      AND .status NEQ ss$_parity
;  882          0988  3      AND .status NEQ ss$_datacheck
;  883          0989  3      AND .status NEQ ss$_endoftape
;  884          0990  3      AND .status NEQ ss$_illblknum
;  885          0991  3      THEN
;  886          0992  4          BEGIN
;  887          0993  4          SIGNAL(
;  888          0994  4              dump$_facility^16 + shr$_readerr + sts$k_error,
;  889          0995  4              1, dump$gl_idesc,
;  890          0996  4              .status);
;  891          0997  4          status = ss$_endoffile;
;  892          0998  4          END
;  893          0999  3      END
;  894          1000  2 ELSE
;  895          1001  3      BEGIN
;  896          1002  3      status = $GET(rab=dump$gl_irab);                    ! Get record
;  897          1003  3      bufdesc[dsc$w_length] = .dump$gl_irab[rab$w_rsz];
;  898          1004  3      bufdesc[dsc$a_pointer] = .dump$gl_irab[rab$l_rbf];
;  899          1005  3      IF NOT .status
;  900          1006  3      THEN
;  901          1007  4          BEGIN
;  902          1008  4          IF .status NEQ rms$_eof
;  903          1009  4          THEN
;  904          1010  4              SIGNAL(
;  905          1011  4                  dump$_facility^16 + shr$_readerr + sts$k_error,
;  906          1012  4                  1, dump$gl_idesc,
;  907          1013  4                  .dump$gl_irab[rab$l_sts], .dump$gl_irab[rab$l_stv]);
;  908          1014  4          status = ss$_endoffile;
;  909          1015  3          END;
;  910          1016  2      END;
;  911          1017  2
;  912          1018  2
;  913          1019  2 RETURN .status
;  914          1020  1 END;


                                                .EXTRN  SYS$QIOW, SYS$GET

                                 00FC 00000     .ENTRY  DUMP$READ, Save R2,R3,R4,R5,R6,R7          ; 0916
                     57 00000000G  00  9E 00002 MOVAB   LIB$SIGNAL, R7                             ;
                     56 00000000G  00  9E 00009 MOVAB   SYS$QIOW, R6                               ;
                     55 00000000'  EF  9E 00010 MOVAB   DUMP$GL_BUFFER+4, R5                       ;
```

```
                 5E               08  C2 00017        SUBL2   #8, SP                                        ;
                 53          04   AC  D0 0001A        MOVL    BUFDESC, R3                                   ; 0945
    03      05   A5               05  E1 0001E        BBC     #5, DUMP$GL_FLAGS+1, 1$                       ; 0928
                            0119  31  00023           BRW     14$                                           ;
                 50        FD14   C5  D0 00026  1$:   MOVL    DUMP$GL_IFAB, R0                              ; 0931
    03      43   A0               04  E1 0002B        BBC     #4, 67(R0), 2$                                ;
                            0081  31  00030           BRW     8$                                            ;
                 54               01  D0 00033  2$:   MOVL    #1, I                                         ; 0934
                                  7E  7C 00036  3$:   CLRQ    -(SP)                                         ; 0943
                                  7E  7C 00038        CLRQ    -(SP)                                         ;
                 7E          FC   A5  3C 0003A        MOVZWL  DUMP$GL_BUFFER, -(SP)                         ;
                                  65  DD 0003E        PUSHL   DUMP$GL_BUFFER+4                              ;
                                  7E  7C 00040        CLRQ    -(SP)                                         ;
                             20   AE  9F 00042        PUSHAB  IOSB                                          ;
                 50        FD14   C5  D0 00045        MOVL    DUMP$GL_IFAB, R0                              ;
                 04          43   A0  E9 0004A        BLBC    67(R0), 4$                                    ;
                                  21  DD 0004E        PUSHL   #33                                           ;
                                  02  11 00050        BRB     5$                                            ;
                                  31  DD 00052  4$:   PUSHL   #49                                           ;
                             F0   A5  DD 00054  5$:   PUSHL   DUMP$GL_CHANNEL                               ;
                                  7E  D4 00057        CLRL    -(SP)                                         ;
                 66               0C  FB 00059        CALLS   #12, SYS$QIOW                                 ;
                 52               50  D0 0005C        MOVL    R0, STATUS                                    ;
            04   BC          02   AE  B0 0005F        MOVW    IOSB+2, @BUFDESC                              ; 0944
            04   A3               65  D0 00064        MOVL    DUMP$GL_BUFFER+4, 4(R3)                       ; 0945
                 03               52  E9 00068        BLBC    STATUS, 6$                                    ; 0946
                 52               6E  3C 0006B        MOVZWL  IOSB, STATUS                                  ;
                 50        FD14   C5  D0 0006E  6$:   MOVL    DUMP$GL_IFAB, R0                              ; 0948
    0B      40   A0               02  E1 00073        BBC     #2, 64(R0), 7$                                ;
                 1A          04   AE  B1 00078        CMPW    IOSB+4, #26                                   ; 0949
                                  05  12 0007C        BNEQ    7$                                            ;
                 52        0870   8F  3C 0007E        MOVZWL  #2160, STATUS                                 ; 0951
      00000870   8F               52  D1 00083  7$:   CMPL    STATUS, #2160                                 ; 0953
                                  70  12 0008A        BNEQ    13$                                           ;
    6B      40   A0               05  E1 0008C        BBC     #5, 64(R0), 13$                               ; 0954
                 67          43   A0  E9 00091        BLBC    67(R0), 13$                                   ; 0955
      0000000G   00               00  FB 00095        CALLS   #0, DUMP$BLANK_LINE                           ; 0958
                                  01  DD 0009C        PUSHL   #1                                            ; 0959
                       0000000G   8F  DD 0009E        PUSHL   #DUMP$_ENDOFFILE                              ;
      0000000G   00               02  FB 000A4        CALLS   #2, DUMP$OUTPUT_GETMSG                        ;
                                  54  D5 000AB        TSTL    I                                             ; 0960
                                  4D  13 000AD        BEQL    13$                                           ;
                 84               54  F4 000AF        SOBGEQ  I, 3$                                         ; 0934
                                  48  11 000B2        BRB     13$                                           ; 0931
            20   A5          1C   A5  D1 000B4  8$:   CMPL    DUMP$GL_CUR_BLOCK, DUMP$GL_MAX_BLOCK          ; 0968
                                  06  1B 000B9        BLEQU   9$                                            ;
                 50        0870   8F  3C 000BB        MOVZWL  #2160, R0                                     ; 0970
                                  04  000C0           RET                                                   ;
                                  7E  7C 000C1  9$:   CLRQ    -(SP)                                         ; 0979
                                  7E  D4 000C3        CLRL    -(SP)                                         ;
                             1C   A5  DD 000C5        PUSHL   DUMP$GL_CUR_BLOCK                             ;
                 7E        0200   8F  3C 000C8        MOVZWL  #512, -(SP)                                   ;
                                  65  DD 000CD        PUSHL   DUMP$GL_BUFFER+4                              ;
                                  7E  7C 000CF        CLRQ    -(SP)                                         ;
                             20   AE  9F 000D1        PUSHAB  IOSB                                          ;
                 04          43   A0  E9 000D4        BLBC    67(R0), 10$                                   ;
                                  21  DD 000D8        PUSHL   #33                                           ;
```

```
                                02  11 000DA          BRB     11$                                                     ;
                                31  DD 000DC 10$:     PUSHL   #49                                                     ;
                             F0 A5  DD 000DE 11$:     PUSHL   DUMP$GL_CHANNEL                                         ;
                                7E  D4 000E1          CLRL    -(SP)                                                   ;
                          66    0C  FB 000E3          CALLS   #12, SYS$QIOW                                           ;
                          52    50  D0 000E6          MOVL    R0, STATUS                                              ;
                          03    52  E9 000E9          BLBC    STATUS, 12$                                             ; 0980
                          52    6E  3C 000EC          MOVZWL  IOSB, STATUS                                            ;
                       04 BC  0200 8F  B0 000EF 12$:    MOVW    #512, @BUFDESC                                          ; 0981
                       04 A3       65  D0 000F5          MOVL    DUMP$GL_BUFFER+4, 4(R3)                                 ; 0982
                          1C   A5  D6 000F9          INCL    DUMP$GL_CUR_BLOCK                                       ; 0983
                          7F   52  E8 000FC 13$:     BLBS    STATUS, 16$                                             ; 0985
                00000870  8F   52  D1 000FF          CMPL    STATUS, #2160                                           ; 0986
                                76  13 00106          BEQL    16$
                000001F4  8F   52  D1 00108          CMPL    STATUS, #500                                            ; 0987
                                6D  13 0010F          BEQL    16$
                0000005C  8F   52  D1 00111          CMPL    STATUS, #92                                             ; 0988
                                64  13 00118          BEQL    16$
                00000878  8F   52  D1 0011A          CMPL    STATUS, #2168                                           ; 0989
                                5B  13 00121          BEQL    16$
                000000DC  8F   52  D1 00123          CMPL    STATUS, #220                                            ; 0990
                                52  13 0012A          BEQL    16$
                                52  DD 0012C          PUSHL   STATUS                                                  ; 0996
                          FF54  C5  9F 0012E          PUSHAB  DUMP$GL_IDESC                                           ; 0993
                                01  DD 00132          PUSHL   #1
                    00000000*   8F  DD 00134          PUSHL   #<<<DUMP$_FACILITY@16>+4272>+2>                         ; 0994
                          67        04  FB 0013A          CALLS   #4, LIB$SIGNAL
                                3A  11 0013D          BRB     15$                                                     ; 0997
                          FD1C  C5  9F 0013F 14$:     PUSHAB  DUMP$GL_IRAB                                            ; 1002
                00000000G 00        01  FB 00143          CALLS   #1, SYS$GET
                          52        50  D0 0014A          MOVL    R0, STATUS                                              ;
                       04 BC  FD3E  C5  B0 0014D          MOVW    DUMP$GL_IRAB+34, @BUFDESC                               ; 1003
                       04 A3  FD44  C5  D0 00153          MOVL    DUMP$GL_IRAB+40, 4(R3)                                  ; 1004
                          22        52  E8 00159          BLBS    STATUS, 16$                                             ; 1005
                0001827A  8F        52  D1 0015C          CMPL    STATUS, #98938                                          ; 1008
                                14  13 00163          BEQL    15$                                                     ;
                          7E  FD24  C5  7D 00165          MOVQ    DUMP$GL_IRAB+8, -(SP)                                   ; 1013
                          FF54  C5  9F 0016A          PUSHAB  DUMP$GL_IDESC                                           ; 1010
                                01  DD 0016E          PUSHL   #1
                    00000000*   8F  DD 00170          PUSHL   #<<<DUMP$_FACILITY@16>+4272>+2>                         ; 1011
                          67        05  FB 00176          CALLS   #5, LIB$SIGNAL
                          52  0870  8F  3C 00179 15$:     MOVZWL  #2160, STATUS                                           ; 1014
                          50        52  D0 0017E 16$:     MOVL    STATUS, R0                                              ; 1019
                                    04  00181          RET                                                             ; 1020

; Routine Size:  386 bytes,    Routine Base:  $CODE$ + 076D
```

```
;  916          1021  1 GLOBAL ROUTINE dump$write(recdesc): NOVALUE=
;  917          1022  2 BEGIN
;  918          1023  2 !
;  919          1024  2 ! Write a record to the output file
;  920          1025  2 !
;  921          1026  2 ! Inputs:
;  922          1027  2 !
;  923          1028  2 !       recdesc pointer to string descriptor for record
;  924          1029  2 !
;  925          1030  2 MAP
;  926          1031  2     recdesc : REF BBLOCK;
;  927          1032  2
;  928          1033  2
;  929          1034  2 dump$gl_orab[rab$w_rsz] = .recdesc[dsc$w_length];
;  930          1035  2 dump$gl_orab[rab$l_rbf] = .recdesc[dsc$a_pointer];
;  931          1036  3 IF NOT $PUT(rab=dump$gl_orab)
;  932          1037  2 THEN
;  933          1038  2     SIGNAL(
;  934          1039  2         dump$_facility^16 + shr$_writeerr + sts$k_severe,
;  935          1040  2         1, dump$gl_odesc,
;  936          1041  2         .dump$gl_orab[rab$l_sts], .dump$gl_orab[rab$l_stv]);
;  937          1042  1 END;


                                                        .EXTRN  SYS$PUT

                                        0004 00000      .ENTRY  DUMP$WRITE, Save R2                             ; 1021
                              52 00000000'  EF   9E 00002       MOVAB   DUMP$GL_ORAB+34, R2                     ;
                              50        04  AC   D0 00009       MOVL    RECDESC, R0                             ; 1034
                              62            60   B0 0000D       MOVW    (R0), DUMP$GL_ORAB+34                   ;
                           06 A2        04  A0   D0 00010       MOVL    4(R0), DUMP$GL_ORAB+40                  ; 1035
                              DE        A2   9F 00015       PUSHAB  DUMP$GL_ORAB                            ; 1036
                00000000G  00             01   FB 00018       CALLS   #1, SYS$PUT                             ;
                           17             50   E8 0001F       BLBS    R0, 1$                                  ;
                           7E        E6  A2   7D 00022       MOVQ    DUMP$GL_ORAB+8, -(SP)                   ; 1041
                               01DA  C2   9F 00026       PUSHAB  DUMP$GL_ODESC                           ; 1038
                                     01   DD 0002A       PUSHL   #1                                      ;
                           00000000*  8F   DD 0002C       PUSHL   #<<<DUMP$_FACILITY@16>+4304>+4>         ; 1039
                00000000G  00             05   FB 00032       CALLS   #5, LIB$SIGNAL                          ;
                                          04   00039 1$:     RET                                             ; 1042

; Routine Size:  58 bytes,    Routine Base:  $CODE$ + 08EF
```

```
;  939          1043  1 ROUTINE dump$close_input(fab): NOVALUE=
;  940          1044  2 BEGIN
;  941          1045  2 !
;  942          1046  2 ! Close the input file
;  943          1047  2 !
;  944          1048  2 ! Inputs:
;  945          1049  2 !
;  946          1050  2 !       fab     Address of fab
;  947          1051  2 !
;  948          1052  2 MAP
;  949          1053  2     fab : REF BBLOCK;
;  950          1054  2 LOCAL
;  951          1055  2     status;
;  952          1056  2
;  953          1057  2
;  954          1058  2 IF .dump$gl_flags[dump$v_records]                          ! If RMS dump
;  955          1059  2 THEN
;  956          1060  3     BEGIN
;  957          1061  4     IF NOT $CLOSE(fab=.fab)                               ! Close fab
;  958          1062  3     THEN
;  959          1063  3         SIGNAL(
;  960          1064  3             dump$_facility^16 + shr$_closein + sts$k_error,
;  961          1065  3             1, dump$gl_idesc,
;  962          1066  3             .fab[fab$l_sts], .fab[fab$l_stv]);
;  963          1067  3     END
;  964          1068  2 ELSE
;  965          1069  3     BEGIN
;  966          1070  3     status = $DASSGN(CHAN=.dump$gl_channel);              ! Else deassign channel
;  967          1071  3     IF NOT .status
;  968          1072  3     THEN
;  969          1073  3         SIGNAL(
;  970          1074  3             dump$_facility^16 + shr$_closein + sts$k_error,
;  971          1075  3             1, dump$gl_idesc,
;  972          1076  3             .status);
;  973          1077  2     END;
;  974          1078  2
;  975          1079  2
;  976          1080  2 ! Free input buffer.
;  977          1081  2 !
;  978          1082  2 IF .dump$gl_buffer[dsc$a_pointer] NEQ 0
;  979          1083  2 THEN
;  980          1084  2     lib$free_vm(dump$gl_buffer[dsc$w_length], dump$gl_buffer[dsc$a_pointer]);
;  981          1085  1 END;


                                                .EXTRN  SYS$CLOSE, SYS$DASSGN

                                    001C 00000 DUMP$CLOSE_INPUT:
                                                .WORD   Save R2,R3,R4                   ; 1043
                    54 00000000G    00   9E 00002       MOVAB   LIB$SIGNAL, R4          ;
                    53 00000000'    EF   9E 00009       MOVAB   DUMP$GL_IDESC, R3       ;
            23  00B1    C3          05   E1 00010       BBC     #5, DUMP$GL_FLAGS+1, 1$ ; 1058
                    52      04      AC   D0 00016       MOVL    FAB, R2                 ; 1061
                                    52   DD 0001A       PUSHL   R2                      ;
               00000000G    00      01   FB 0001C       CALLS   #1, SYS$CLOSE           ;
                    30              50   E8 00023       BLBS    R0, 2$                  ;
```

```
                         7E      08   A2  7D 00026          MOVQ    8(R2), -(SP)                                    ; 1066
                                      53  DD 0002A          PUSHL   R3                                              ; 1063
                                      01  DD 0002C          PUSHL   #1                                              ;
                              00000000* 8F  DD 0002E          PUSHL   #<<<DUMP$_FACILITY@16>+4176>+2>                 ; 1064
                         64           05  FB 00034          CALLS   #5, LIB$SIGNAL                                  ;
                                      1D  11 00037          BRB     2$                                              ; 1058
                              009C    C3  DD 00039 1$:      PUSHL   DUMP$GL_CHANNEL                                 ; 1070
                00000000G     00           01  FB 0003D          CALLS   #1, SYS$DASSGN                                  ;
                              0F           50  E8 00044          BLBS    STATUS, 2$                                      ; 1071
                                           50  DD 00047          PUSHL   STATUS                                          ; 1076
                                           53  DD 00049          PUSHL   R3                                              ; 1073
                                           01  DD 0004B          PUSHL   #1                                              ;
                                   00000000* 8F  DD 0004D          PUSHL   #<<<DUMP$_FACILITY@16>+4176>+2>                 ; 1074
                                   64        04  FB 00053          CALLS   #4, LIB$SIGNAL                                  ;
                                      00AC    C3  D5 00056 2$:      TSTL    DUMP$GL_BUFFER+4                                ; 1082
                                              0F  13 0005A          BEQL    3$                                              ;
                                      00AC    C3  9F 0005C          PUSHAB  DUMP$GL_BUFFER+4                                ; 1084
                                      00A8    C3  9F 00060          PUSHAB  DUMP$GL_BUFFER                                  ;
                00000000G     00           02  FB 00064          CALLS   #2, LIB$FREE_VM                                 ;
                                           04  0006B 3$:      RET                                                     ; 1085

; Routine Size:  108 bytes,    Routine Base:  $CODE$ + 0929
```

```
;  983          1086  1 ROUTINE dump$close_output: NOVALUE=
;  984          1087  2 BEGIN
;  985          1088  2 !
;  986          1089  2 ! Close the output file
;  987          1090  2 !
;  988          1091  2 ! Inputs:
;  989          1092  2 !
;  990          1093  2 !
;  991          1094  3 IF NOT $CLOSE(fab=dump$gl_ofab)
;  992          1095  2 THEN
;  993          1096  2     SIGNAL(
;  994          1097  2         dump$_facility^16 + shr$_closeout + sts$k_error,
;  995          1098  2         1, dump$gl_odesc,
;  996          1099  2         .dump$gl_ofab[fab$l_sts], .dump$gl_ofab[fab$l_stv]);
;  997          1100  1 END;


                                        0004 00000 DUMP$CLOSE_OUTPUT:
                                                   .WORD   Save R2                                          ; 1086
                    52 00000000'   EF   9E 00002           MOVAB   DUMP$GL_OFAB, R2                         ;
                                   52   DD 00009           PUSHL   R2                                       ; 1094
          00000000G   00           01   FB 0000B           CALLS   #1, SYS$CLOSE                            ;
                      17           50   E8 00012           BLBS    R0, 1$                                   ;
                      7E       08  A2   7D 00015           MOVQ    DUMP$GL_OFAB+8, -(SP)                    ; 1099
                           01B8    C2   9F 00019           PUSHAB  DUMP$GL_ODESC                            ; 1096
                                   01   DD 0001D           PUSHL   #1                                       ;
                       00000000*   8F   DD 0001F           PUSHL   #<<<DUMP$_FACILITY@16>+4184>+2>          ; 1097
          00000000G   00           05   FB 00025           CALLS   #5, LIB$SIGNAL                           ;
                                        04 0002C 1$:       RET                                              ; 1100

; Routine Size:  45 bytes,    Routine Base:  $CODE$ + 0995
```

```
;  999     1101  1 ROUTINE dump$list_width(fab): NOVALUE=
; 1000     1102  2 BEGIN
; 1001     1103  2 !
; 1002     1104  2 ! Determine the width of the listing line
; 1003     1105  2 ! FAB is the fab of the open file, width returned in dump$gl_width,
; 1004     1106  2 ! and dump$gl_outdesc set up as string descriptor for output buffer
; 1005     1107  2 !
; 1006     1108  2 MAP
; 1007     1109  2     fab : REF BBLOCK;
; 1008     1110  2 BIND
; 1009     1111  2     nam = .fab[fab$l_nam] : BBLOCK;
; 1010     1112  2 LOCAL
; 1011     1113  2     devnamdesc : BBLOCK[dsc$c_s_bln],
; 1012     1114  2     devnambuf : VECTOR[nam$c_maxrss, BYTE],
; 1013     1115  2     devnambufdesc : BBLOCK[dsc$c_s_bln],
; 1014     1116  2     devinfobuf : BBLOCK[dib$k_length],
; 1015     1117  2     devinfodesc : BBLOCK[dsc$c_s_bln];
; 1016     1118  2 LITERAL
; 1017     1119  2     ch_escape = %O'033';                        ! ASCII <ESC>
; 1018     1120  2
; 1019     1121  2
; 1020     1122  2 dump$gl_width = dump$c_deflisiz;                ! Assume default
; 1021     1123  2
; 1022     1124  2 devnamdesc[dsc$a_pointer] = .nam[nam$l_dev];
; 1023     1125  2 devnamdesc[dsc$w_length] =
; 1024     1126  2     CH$FIND_CH(.nam[nam$b_dev], .nam[nam$l_dev], %C':')
; 1025     1127  2     - .nam[nam$l_dev];
; 1026     1128  2 devnambufdesc[dsc$w_length] = nam$c_maxrss;
; 1027     1129  2 devnambufdesc[dsc$a_pointer] = devnambuf;
; 1028     1130  2 $TRNLOG(LOGNAM=devnamdesc, RSLLEN=devnambufdesc, RSLBUF=devnambufdesc);
; 1029     1131  2 IF .devnambuf[0] EQL ch_escape                  ! If process permanent file
; 1030     1132  2 THEN
; 1031     1133  3     BEGIN
; 1032     1134  3     devnambufdesc[dsc$w_length] = .devnambufdesc[dsc$w_length] - 4;
; 1033     1135  3     devnambufdesc[dsc$a_pointer] = .devnambufdesc[dsc$a_pointer] + 4;
; 1034     1136  2     END;
; 1035     1137  2
; 1036     1138  2
; 1037     1139  2 ! Do a $GETDEV to get the width.
; 1038     1140  2 !
; 1039     1141  2 devinfodesc[dsc$w_length] = dib$k_length;       ! Set up descriptor for $GETDEV
; 1040     1142  2 devinfodesc[dsc$a_pointer] = devinfobuf;
; 1041     1143  3 IF $GETDEV(DEVNAM=devnambufdesc, SCDBUF=devinfodesc)
; 1042     1144  2 THEN
; 1043     1145  2     dump$gl_width = MINU(.devinfobuf[dib$w_devbufsiz], dump$c_maxlisiz);
; 1044     1146  2
; 1045     1147  2
; 1046     1148  2 ! Set up output buffer descriptor.
; 1047     1149  2 !
; 1048     1150  2 dump$gl_outdesc[dsc$w_length] = .dump$gl_width;
; 1049     1151  2 dump$gl_outdesc[dsc$a_pointer] = dump$ab_outbuf;
; 1050     1152  1 END;                                           ! Of dump$list_width


                                                    .EXTRN  SYS$TRNLOG, SYS$GETDEV
```

```
                                000C 00000 DUMP$LIST_WIDTH:
                                           .WORD   Save R2,R3                                          ; 1101
                    53 00000000'    EF  9E 00002       MOVAB   DUMP$GL_WIDTH, R3
                    5E      FE74    CE  9E 00009       MOVAB   -396(SP), SP                            ;
                    50        04    AC  D0 0000E       MOVL    FAB, R0                                 ; 1111
                    52        28    A0  D0 00012       MOVL    40(R0), R2
                    63        50    8F  9A 00016       MOVZBL  #80, DUMP$GL_WIDTH                      ; 1122
                FC  AD        44    A2  D0 0001A       MOVL    68(R2), DEVNAMDESC+4                    ; 1124
                    50        39    A2  9A 0001F       MOVZBL  57(R2), R0                              ; 1126
            44  B2            50        3A 3A 00023    LOCC    #58, R0, @68(R2)
                                        02 12 00028    BNEQ    1$
                                        51 D4 0002A    CLRL    R1
            F8  AD            51        44 A2 A3 0002C 1$:     SUBW3   68(R2), R1, DEVNAMDESC          ; 1127
                      7C      AE        FF 8F 9B 00032       MOVZBW  #255, DEVNAMBUFDESC               ; 1128
                    0080      CE      0084 CE 9E 00037       MOVAB   DEVNAMBUF, DEVNAMBUFDESC+4        ; 1129
                                           7E 7C 0003E       CLRQ    -(SP)                             ; 1130
                                           7E D4 00040       CLRL    -(SP)
                                      0088 CE 9F 00042       PUSHAB  DEVNAMBUFDESC
                                      008C CE 9F 00046       PUSHAB  DEVNAMBUFDESC
                                        F8 AD 9F 0004A       PUSHAB  DEVNAMDESC
                          00000000G     00    06 FB 0004D    CALLS   #6, SYS$TRNLOG
                                        1B  0084 CE 91 00054 CMPB    DEVNAMBUF, #27                    ; 1131
                                            09 12 00059    BNEQ    2$
                      7C    AE                04 A2 0005B    SUBW2   #4, DEVNAMBUFDESC                 ; 1134
                    0080    CE                04 C0 0005F    ADDL2   #4, DEVNAMBUFDESC+4               ; 1135
                            6E      74    8F  9B 00064 2$:     MOVZBW  #116, DEVINFODESC                 ; 1141
                      04    AE      08    AE  9E 00068       MOVAB   DEVINFOBUF, DEVINFODESC+4         ; 1142
                                            5E DD 0006D       PUSHL   SP                                ; 1143
                                            7E 7C 0006F       CLRQ    -(SP)                             ;
                                            7E D4 00071       CLRL    -(SP)                             ;
                                      008C  CE  9F 00073       PUSHAB  DEVNAMBUFDESC                     ;
                          00000000G     00    05 FB 00077    CALLS   #5, SYS$GETDEV                    ;
                                        12  50 E9 0007E       BLBC    R0, 4$                            ;
                                        50  0E  AE 3C 00081    MOVZWL  DEVINFOBUF+6, R0                  ; 1145
                                    0084 8F     50 B1 00085    CMPW    R0, #132                          ;
                                            04 1B 0008A       BLEQU   3$                                ;
                                        50  84  8F 9A 0008C    MOVZBL  #132, R0                          ;
                                        63      50 D0 00090 3$:  MOVL    R0, DUMP$GL_WIDTH                 ;
                                      F4  A3    63 B0 00093 4$:  MOVW    DUMP$GL_WIDTH, DUMP$GL_OUTDESC    ; 1150
                                      F8  A3  FF70 C3 9E 00097   MOVAB   DUMP$AB_OUTBUF, DUMP$GL_OUTDESC+4 ; 1151
                                                  04 0009D     RET                                       ; 1152

; Routine Size:  158 bytes,    Routine Base:  $CODE$ + 09C2
```

```
1 ROUTINE dump$file_error(message,fab,sts,stv): NOVALUE=
2 BEGIN
2 !++
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       This routine signals an error for a file.
2 !
2 ! Inputs:
2 !
2 !       message         Message
2 !       fab             Address of the fab
2 !       sts, stv        STS and STV values
2 !
2 !--
2
2 MAP
2     fab : REF BBLOCK;
2 BIND
2     nam = .fab[fab$l_nam] : BBLOCK;
2 LOCAL
2     filedesc : BBLOCK[dsc$c_s_bln];
2
2
2 CH$FILL(0, dsc$c_s_bln, filedesc);
2
2 IF .nam[nam$b_rsl] NEQ 0                             ! If resultant name present
2 THEN
3     BEGIN
3     filedesc[dsc$w_length] = .nam[nam$b_rsl];
3     filedesc[dsc$a_pointer] = .nam[nam$l_rsa];
3     END
2 ELSE IF .nam[nam$b_esl] NEQ 0                        ! If expanded name present
2 THEN
3     BEGIN
3     filedesc[dsc$w_length] = .nam[nam$b_esl];
3     filedesc[dsc$a_pointer] = .nam[nam$l_esa]
3     END
2 ELSE
3     BEGIN
3     filedesc[dsc$w_length] = .fab[fab$b_fns];    ! Use filename string
3     filedesc[dsc$a_pointer] = .fab[fab$l_fna];   ! if all else fails
2     END;
2
2
2 SIGNAL(.message, 1, filedesc, .sts, .stv);
1 END;                                                 ! Of dump$file_error


                                 00FC 00000 DUMP$FILE_ERROR:
                                                 .WORD   Save R2,R3,R4,R5,R6,R7              ; 1153
                  5E             08  C2 00002    SUBL2   #8, SP                              ;
                  57        08   AC  D0 00005    MOVL    FAB, R7                             ; 1171
                  56        28   A7  D0 00009    MOVL    40(R7), R6                          ;
    08       00   6E             00  2C 0000D    MOVC5   #0, (SP), #0, #8, FILEDESC          ; 1176
```

```
                                    6E     00012 ;
                                03  A6  95 00013          TSTB    3(R6)                                   ; 1178
                                    0B  13 00016          BEQL    1$                                      ;
                                6E  03  A6  9B 00018          MOVZBW  3(R6), FILEDESC                         ; 1181
                            04  AE  04  A6  D0 0001C          MOVL    4(R6), FILEDESC+4                       ; 1182
                                    19  11 00021          BRB     3$                                      ; 1178
                                0B  A6  95 00023 1$:      TSTB    11(R6)                                  ; 1184
                                    0B  13 00026          BEQL    2$                                      ;
                                6E  0B  A6  9B 00028          MOVZBW  11(R6), FILEDESC                        ; 1187
                            04  AE  0C  A6  D0 0002C          MOVL    12(R6), FILEDESC+4                      ; 1188
                                    09  11 00031          BRB     3$                                      ;
                                6E  34  A7  9B 00033 2$:      MOVZBW  52(R7), FILEDESC                        ; 1192
                            04  AE  2C  A7  D0 00037          MOVL    44(R7), FILEDESC+4                      ; 1193
                                7E  0C  AC  7D 0003C 3$:      MOVQ    STS, -(SP)                              ; 1197
                                08  AE  9F 00040          PUSHAB  FILEDESC                                ;
                                    01  DD 00043          PUSHL   #1                                      ;
                                04  AC  DD 00045          PUSHL   MESSAGE                                 ;
                    00000000G   00      05  FB 00048          CALLS   #5, LIB$SIGNAL                          ;
                                        04  0004F          RET                                             ; 1198

; Routine Size:  80 bytes,    Routine Base:  $CODE$ + 0A60
```

```
; 1099          1199  1 END
; 1100          1200  0 ELUDOM



                                                        .EXTRN  LIB$SIGNAL, LIB$STOP

;                               PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $GLOBAL$                       804  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                           40  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  _LIB$KEY0$                       8  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;  _LIB$STATE$                     86  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;  _LIB$KEY1$                      17  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;  $CODE$                        2736  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                         358  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                           Library Statistics
;                                        --------  Symbols  --------      Pages       Processing
;       File                             Total    Loaded    Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776       178          1         581        00:00.8
;  _$255$DUA28:[SYSLIB]TPAMAC.L32;1         42        28         66          14        00:00.1



;                                       COMMAND QUALIFIERS

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

; Size:          2736 code + 1313 data bytes
; Run Time:         00:36.6
; Elapsed Time:     02:26.3
; Lines/CPU Min:     1965
; Lexemes/CPU-Min: 55923
; Memory Used:   319 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