```
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
```

FILEID**FILES

```
FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
FF              II      LL          EE          SS
FF              II      LL          EE          SS
FF              II      LL          EE          SS
FF              II      LL          EE          SS
FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
FF              II      LL          EE                  SS
FF              II      LL          EE                  SS
FF              II      LL          EE                  SS      ....
FF              II      LL          EE                  SS      ....
FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS        ....
FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS        ....

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
;  2          0002  0 FILES (IDENT = 'V04-000') =
;  3          0003  1 BEGIN
;  4          0004  1
;  5          0005  1 !
;  6          0006  1 !*****************************************************************************
;  7          0007  1 !*                                                                           *
;  8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;  9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 10          0010  1 !*  ALL RIGHTS RESERVED.                                                     *
; 11          0011  1 !*                                                                           *
; 12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 17          0017  1 !*  TRANSFERRED.                                                             *
; 18          0018  1 !*                                                                           *
; 19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 21          0021  1 !*  CORPORATION.                                                             *
; 22          0022  1 !*                                                                           *
; 23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 25          0025  1 !*                                                                           *
; 26          0026  1 !*                                                                           *
; 27          0027  1 !*****************************************************************************
; 28          0028  1
; 29          0029  1 !++
; 30          0030  1 ! FACILITY:  ACC, Account file dumper
; 31          0031  1 !
; 32          0032  1 ! ABSTRACT:
;  3          0033  1 !
; 34          0034  1 !       This module contains the file manipulation code for
; 35          0035  1 !       the accounting utilities.
; 36          0036  1 !
; 37          0037  1 ! ENVIRONMENT:
; 38          0038  1 !
; 39          0039  1 !       VAX/VMS operating system. unprivileged user mode,
; 40          0040  1 !
; 41          0041  1 ! AUTHOR:  Elliott A. Drayton, June 1983
; 42          0042  1 !
; 43          0043  1 ! Modified by:
; 44          0044  1 !
; 45          0045  1 !       V04-008 EAD0196         Elliott A. Drayton              23-Jul-1984
; 46          0046  1 !               Made OUTPUT_NAM hold the address of the name block.
; 47          0047  1 !
; 48          0048  1 !       V04-007 EAD0187         Elliott A. Drayton               6-Jul-1984
; 49          0049  1 !               Removed LSTLUN.
; 50          0050  1 !
; 51          0051  1 !       V04-006 EAD0161         Elliott A. Drayton              20-Apr-1984
; 52          0052  1 !               Removed related name for INPUT_NAM.
; 53          0053  1 !
; 54          0054  1 !       V04-005 EAD0132         Elliott A. Drayton               9-Apr-1984
; 55          0055  1 !               Added routine WRITE_MSG.
; 56          0056  1 !
; 57          0057  1 !       V04-004 EAD0030         Elliott A. Drayton              23-Aug-1983
```

```
;    58          0058  1 !                Removed code to set up FORMS.
;    59          0059  1 !
;    60          0060  1 !
;    61          0061  1 !--
;    62          0062  1
;    63          0063  1 !----------------------------------------------------------------------+
;    64          0064  1 !                                                                      |
;    65          0065  1 !                                INCLUDE  FILES                        |
;    66          0066  1 !                                                                      |
;    67          0067  1 !----------------------------------------------------------------------+
;    68          0068  1
;    69          0069  1 REQUIRE 'SRC$:ERFDEF.REQ';              ! Common ERF definitions
;    70          0355  1 REQUIRE 'SRC$:RECSELDEF.REQ';           ! Defines syecom and emb fields.
```

```
;    72          0486  1 !----------------------------------------------------------------------+
;    73          0487  1 !                                                                      |
;    74          0488  1 !                          TABLE  OF  CONTENTS                         |
;    75          0489  1 !                                                                      |
;    76          0490  1 !----------------------------------------------------------------------+
;    77          0491  1 EXTERNAL ROUTINE
;    78          0492  1         LOG_FILENAME,
;    79          0493  1         OPEN_OUT_FILE,          ! Fortran routine need to do I/O from DEVICE MOD
;    80          0494  1         PARSE_OUTPUT_FILES,
;    81          0495  1         WRITE_MSG;
;    82          0496  1
;    83          0497  1 !----------------------------------------------------------------------+
;    84          0498  1 !                                                                      |
;    85          0499  1 !                          GENERAL  STORAGE  DEFINITIONS               |
;    86          0500  1 !                                                                      |
;    87          0501  1 !----------------------------------------------------------------------+
;    88          0502  1
;    89          0503  1 EXTERNAL
;    90          0504  1    LSTLUN_RAB_ADDRESS:          REF $BBLOCK [],
;    91          0505  1    SYS$OUTPUT_RAB_ADDRESS:      REF $BBLOCK [];
;    92          0506  1
;    93          0507  1 OWN
;    94          0508  1
;    95          0509  1 DATEXT: INITIAL ('.DAT'),               ! ".DAT" extension
;    96          0510  1 LISEXT: INITIAL ('.LIS'),               ! ".LIS" extension
;    97          0511  1
;    98          0512  1 INPUT_NAM_RESULT:                       ! Resultant input name
;    99          0513  1         VECTOR [NAM$C_MAXRSS,BYTE],     !  -allocate storage
;   100          0514  1
;   101          0515  1 INPUT_NAM_EXPANDED:                     ! Expanded input name
;   102          0516  1         VECTOR [NAM$C_MAXRSS,BYTE],     !  -allocate storage
;   103          0517  1
;   104          0518  1 RELATED_NAM_RESULT:                     ! Resultant related name
;   105          0519  1         VECTOR [NAM$C_MAXRSS,BYTE],     !  -allocate storage
;   106          0520  1
;   107          0521  1 OUTPUT_NAM_RESULT:                      ! Resultant output name
;   108          0522  1         VECTOR [NAM$C_MAXRSS,BYTE],     !  -allocate storage
;   109          0523  1
;   110          0524  1 OUTPUT_NAM_EXPANDED:                    ! Expanded output name
;   111          0525  1         VECTOR [NAM$C_MAXRSS,BYTE],     !  -allocate storage
;   112          0526  1
;   113          0527  1 REJECTED_NAM_RESULT:                    ! Resultant rejected name
;   114          0528  1         VECTOR [NAM$C_MAXRSS,BYTE],     !  -allocate storage
;   115          0529  1
;   116          0530  1 REJECTED_NAM_EXPANDED:                  ! Expanded rejected name
;   117          0531  1         VECTOR [NAM$C_MAXRSS,BYTE];     !  -allocate storage
;   118          0532  1
```

```
;  120          0533  1 GLOBAL
;  121          0534  1
;  122        P 0535  1 RELATED_NAM:    $NAM(                 ! Related NAM block
;  123        P 0536  1                 RSA = RELATED_NAM_RESULT,  !  -file name address after opening
;  124          0537  1                 RSS = NAM$C_MAXRSS),       !  -(buffer size)
;  125          0538  1
;  126        P 0539  1 INPUT_NAM:      $NAM(                 ! Input NAM block
;  127        P 0540  1                 ESA = INPUT_NAM_EXPANDED,  !  -file name address after parsing
;  128        P 0541  1                 ESS = NAM$C_MAXRSS,        !  -(buffer size)
;  129        P 0542  1                 RSA = INPUT_NAM_RESULT,    !  -file name address after opening
;  130          0543  1                 RSS = NAM$C_MAXRSS),       !  -(buffer size)
;  131          0544  1
;  132        P 0545  1 OUTPUT_NAM_BLK: $NAM(                 ! Output NAM block
;  133        P 0546  1                 RLF = INPUT_NAM,           !  -get further defaults from input
;  134        P 0547  1                 ESA = OUTPUT_NAM_EXPANDED, !  -file name address after parsing
;  135        P 0548  1                 ESS = NAM$C_MAXRSS,        !  -(buffer size)
;  136        P 0549  1                 RSA = OUTPUT_NAM_RESULT,   !  -file name address after open
;  137          0550  1                 RSS = NAM$C_MAXRSS),       !  -(buffer size)
;  138          0551  1
;  139        P 0552  1 REJECTED_NAM:   $NAM(                 ! Rejected NAM block
;  140        P 0553  1                 RLF = INPUT_NAM,           !  -related file name
;  141        P 0554  1                 ESA = REJECTED_NAM_EXPANDED,  !  -file name address after parsing
;  142        P 0555  1                 ESS = NAM$C_MAXRSS,        !  -(buffer size)
;  143        P 0556  1                 RSA = REJECTED_NAM_RESULT, !  -file name address after open
;  144          0557  1                 RSS = NAM$C_MAXRSS),       !  -(buffer size)
;  145          0558  1
;  146          0559  1 INPUT_XABFHC:   $XABFHC(),            ! Input FHC XAB block
;  147          0560  1
;  148        P 0561  1 INPUT_FAB:      $FAB(                 ! Input FAB block
;  149        P 0562  1                 XAB = INPUT_XABFHC,        !  -address of FHC XAB block
;  150        P 0563  1                 FOP = (SQO),               !  -sequential operations only
;  151        P 0564  1                 SHR = (PUT,UPI),           !  -allow un-interlocked, sharing
;  152        P 0565  1                 NAM = INPUT_NAM,           !  -address of NAM block
;  153        P 0566  1                 DNM = 'ERRLOG.SYS',        !  -default name
;  154          0567  1                 FAC = GET),                !  -open for input
;  155          0568  1
;  156        P 0569  1 INPUT_RAB:      $RAB(                 ! Input RAB block
;  157        P 0570  1                 USZ = 512,                 !  -(buffer size)
;  158        P 0571  1                 MBC = 16,                  !  -multi-block count
;  159        P 0572  1                 MBF = 2,                   !  -multi-buffer count
;  160        P 0573  1                 ROP = (RAH),               !  -read-ahead processing
;  161        P 0574  1                 CTX = MSG$_READERR,        !  -error message value
;  162          0575  1                 FAB = INPUT_FAB),          !  -address of FAB to be CONNECTed
;  163          0576  1
;  164        P 0577  1 OUTPUT_FAB:     $FAB(                 ! Output FAB block
;  165        P 0578  1                 CTX = MSG$_OPENOUT,        !  -error message value
;  166        P 0579  1                 FOP = (OFP, SQO),          !  -output file parse, sequential only
;  167        P 0580  1                 NAM = OUTPUT_NAM_BLK,      !  -address of NAM block
;  168        P 0581  1                 DNS = 4,                   !  -default extension size
;  169          0582  1                 DNA = DATEXT),             !  -default extension address
;  170          0583  1
;  171        P 0584  1 OUTPUT_RAB: $RAB(                   ! Output RAB block
;  172        P 0585  1                 CTX = MSG$_WRITEERR,       !  -specify error message
;  173          0586  1                 FAB = output_fab),         !  -address of FAB block
;  174          0587  1
;  175        P 0588  1 REJECTED_FAB:   $FAB(                 ! Rejected FAB block
;  176        P 0589  1                 DNM = '.REJ',              !  -default extension
```

```
;  177          P 0590  1          CTX = MSG$_OPENOUT,                 !  -error message value
;  178          P 0591  1          FOP = (OFP, SQO),                   !  -output file parse, sequential only
;  179            0592  1          NAM = REJECTED_NAM),                !  -address of NAM block
;  180            0593  1
;  181          P 0594  1 REJECTED_RAB: $RAB(                          ! Rejected RAB block
;  182          P 0595  1          CTX = MSG$_WRITEERR,                !  -specify error message
;  183          P 0596  1          MBC = 16,                           !  -multi-block count
;  184          P 0597  1          MBF = 2,                            !  -multi-buffer count
;  185          P 0598  1          ROP = (WBH),                        !  -write behind processing
;  186            0599  1          FAB = REJECTED_FAB),                !  -address of FAB block
;  187            0600  1
;  188            0601  1 OUTPUT_NAM:     LONG INITIAL (OUTPUT_NAM_BLK);
```

```
;  190     0602  1 UNDECLARE LOG_FILENAME;
;  191     0603  1
;  192     0604  1 Global routine LOG_FILENAME (rms) =
;  193     0605  1
;  194     0606  1 !----
;  195     0607  1 !
;  196     0608  1 ! Functional description
;  197     0609  1 !
;  198     0610  1 !       This routine is called to signal a message to
;  199     0611  1 !       the user based on an error code and file name
;  200     0612  1 !       that are imbedded in the passed parameter.
;  201     0613  1 !
;  202     0614  1 ! Input parameters
;  203     0615  1 !
;  204     0616  1 !       RMS = Either a FAB or a RAB
;  205     0617  1 !       RAB$L_FAB = pointer to fab block        (If input was a RAB)
;  206     0618  1 !       FAB$L_NAM = pointer to name block
;  207     0619  1 !       RAB$L_CTX = error message to be used    (If input was a RAB)
;  208     0620  1 !       FAB$L_CTX = error message to be used    (If input was a FAB)
;  209     0621  1 !
;  210     0622  1 !
;  211     0623  1 ! Output parameters
;  212     0624  1 !
;  213     0625  1 !       Expanded error messages to user
;  214     0626  1 !       Status is RETURNed
;  215     0627  1 !
;  216     0628  1 !----
;  217     0629  1
;  218     0630  2 BEGIN
;  219     0631  2
;  220     0632  2 MAP
;  221     0633  2     rms:  ref $bblock;                          ! Define block format
;  222     0634  2
;  223     0635  2
;  224     0636  2 LOCAL
;  225     0637  2         fab: ref $bblock,               ! Pointer to FAB block
;  226     0638  2         nam: ref $bblock,               ! Pointer to NAM block
;  227     0639  2         rms_sts,                        ! Temporary primary status holder
;  228     0640  2         rms_stv,                        ! Temporary secondary status holder
;  229     0641  2         rms_ctx,                        ! Temporary user context holder
;  230     0642  2         status: $bblock [long],         ! Local "catch all" status return
;  231     0643  2         desc:   vector [2, long];       ! Temporary string descriptor
;  232     0644  2
;  233     0645  2
;  234     0646  2 !
;  235     0647  2 !SET UP VALUES --
;  236     0648  2 !       Fetch the primary and secondary status values and the user
;  237     0649  2 !       context field from the RMS structure.  If a RAB was passed
;  238     0650  2 !       then fetch the address of the associated FAB.
;  239     0651  2 !
;  240     0652  2
;  241     0653  2 If .rms [rab$b_bid] eql rab$c_bid then  ! If this is a rab
;  242     0654  3         BEGIN
;  243     0655  3         fab = .rms [rab$l_fab];
;  244     0656  3         rms_sts = .rms [rab$l_sts];
;  245     0657  3         rms_stv = .rms [rab$l_stv];
;  246     0658  3         rms_ctx = .rms [rab$l_ctx];
```

```
;  247     0659  3          END
;  248     0660  3      else BEGIN
;  249     0661  3           fab = .rms;
;  250     0662  3           rms_sts = .rms [fab$l_sts];
;  251     0663  3           rms_stv = .rms [fab$l_stv];
;  252     0664  3           rms_ctx = .rms [fab$l_ctx];
;  253     0665  2           END;
;  254     0666  2
;  255     0667  2 nam = .fab [fab$l_nam];                 ! Fetch address of NAM block
;  256     0668  2
;  257     0669  2
;  258     0670  2
;  259     0671  2 !
;  260     0672  2 !CHECK FOR EOF --
;  261     0673  2 !       End of file errors are not reported by this routine.
;  262     0674  2 !
;  263     0675  2
;  264     0676  2 If  .rms [rab$b_bid] eql rab$c_bid      ! If this is a rab
;  265     0677  2 and .rms_sts eql rms$_eof               !  - and error is end of file
;  266     0678  2 and .rms_ctx eql msg$_readerr           !  - and this was a read call
;  267     0679  2     then return rms$_eof;               ! don't bother to report it
;  268     0680  2
;  269     0681  2
;  270     0682  2
;  271     0683  2 !
;  272     0684  2 !FETCH FILE NAME --
;  273     0685  2 !       Find the best filename available.  Start with the
;  274     0686  2 !       resultant name; if not present try for the expanded
;  275     0687  2 !       name; if also missing then settle for the original
;  276     0688  2 !       file name.
;  277     0689  2 !
;  278     0690  2
;  279     0691  2 If .nam[nam$b_rsl] neq 0 then           ! IF result string nonblank,
;  280     0692  3     BEGIN
;  281     0693  3     desc[0] = .nam[nam$b_rsl];          ! then display it
;  282     0694  3     desc[1] = .nam[nam$l_rsa];
;  283     0695  3     END
;  284     0696  3
;  285     0697  2 else if .nam[nam$b_esl] neq 0 then      ! Or if expanded name nonblank
;  286     0698  3     BEGIN
;  287     0699  3     desc[0] = .nam[nam$b_esl];          ! then display it
;  288     0700  3     desc[1] = .nam[nam$l_esa];
;  289     0701  3     END
;  290     0702  3
;  291     0703  3 else BEGIN
;  292     0704  3     desc[0] = .fab[fab$b_fns];          ! Otherwise, use original
;  293     0705  3     desc[1] = .fab[fab$l_fna];          ! name string in FAB
;  294     0706  2     END;
;  295     0707  2
;  296     0708  2
;  297     0709  2
;  298     0710  2 !
;  299     0711  2 !NOTIFY THE USER --
;  300     0712  2 !       Construct an error message using the user supplied context (CTX)
;  301     0713  2 !       field and the RMS supplied primary (STS) and secondary (STV)
;  302     0714  2 !       status fields.  Signal it to the user.
;  303     0715  2 !
```

```
;  304          0716  2
;  305          0717  2 signal (.rms_ctx, 1 ,desc,                ! Output an error message
;  306          0718  2                   .rms_sts,               ! with RMS error code
;  307          0719  2                   .rms_stv);              ! and secondary code
;  308          0720  2
;  309          0721  2
;  310          0722  2 return .rms_sts;                          ! Pass on the status
;  311          0723  2
;  312          0724  1 END;


                                                        .TITLE  FILES
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT,NOWRT,NOEXE,  PIC,2

                53  59  53  2E  47  4F  4C  52  52  45  00000 P.AAA:   .ASCII  \ERRLOG.SYS\                    ;
                                      4A  45  52  2E  0000A P.AAB:   .ASCII  \.REJ\                        ;

                                                        .PSECT  $OWN$,NOEXE,  PIC,2

                                      54  41  44  2E  00000 DATEXT:  .ASCII  \.DAT\                        ;
                                      53  49  4C  2E  00004 LISEXT:  .ASCII  \.LIS\                        ;
                                                      00008 INPUT_NAM_RESULT:
                                                              .BLKB   255
                                                      00107   .BLKB   1
                                                      00108 INPUT_NAM_EXPANDED:
                                                              .BLKB   255
                                                      00207   .BLKB   1
                                                      00208 RELATED_NAM_RESULT:
                                                              .BLKB   255
                                                      00307   .BLKB   1
                                                      00308 OUTPUT_NAM_RESULT:
                                                              .BLKB   255
                                                      00407   .BLKB   1
                                                      00408 OUTPUT_NAM_EXPANDED:
                                                              .BLKB   255
                                                      00507   .BLKB   1
                                                      00508 REJECTED_NAM_RESULT:
                                                              .BLKB   255
                                                      00607   .BLKB   1
                                                      00608 REJECTED_NAM_EXPANDED:
                                                              .BLKB   255

                                                        .PSECT  $GLOBAL$,NOEXE,  PIC,2

                      C2  00000 RELATED_NAM::
                                                              .BYTE   2                                     ;
                      60  00001                               .BYTE   96                                    ;
                      FF  00002                               .BYTE   -1                                    ;
                      00  00003                               .BYTE   0                                     ;
                00000000' 00004                               .ADDRESS RELATED_NAM_RESULT                   ;
                      00  00008                               .BYTE   0                                     ;
                      00  00009                               .BYTE   0                                     ;
                      00  0000A                               .BYTE   0                                     ;
                      00  0000B                               .BYTE   0                                     ;
                00000000  0000C                               .LONG   0                                     ;
```

```
00000000  00010          .LONG   0                                                       ;
    0000#  00014          .WORD   0[8]                                                    ;
    0000#  00024          .WORD   0[3]                                                    ;
    0000#  0002A          .WORD   0[3]                                                    ;
00000000  00030          .LONG   0                                                       ;
00000000  00034          .LONG   0                                                       ;
      00  00038          .BYTE   0                                                       ;
      00  00039          .BYTE   0                                                       ;
      00  0003A          .BYTE   0                                                       ;
      00  0003B          .BYTE   0                                                       ;
      00  0003C          .BYTE   0                                                       ;
      00  0003D          .BYTE   0                                                       ;
      00# 0003E          .BYTE   0[2]                                                    ;
00000000  00040          .LONG   0                                                       ;
00000000  00044          .LONG   0                                                       ;
00000000  00048          .LONG   0                                                       ;
00000000  0004C          .LONG   0                                                       ;
00000000  00050          .LONG   0                                                       ;
00000000  00054          .LONG   0                                                       ;
00000000# 00058          .LONG   0[2]                                                    ;
      02  00060 INPUT_NAM::                                                             ;
                         .BYTE   2                                                       ;
      60  00061          .BYTE   96                                                      ;
      FF  00062          .BYTE   -1                                                      ;
      00  00063          .BYTE   0                                                       ;
00000000' 00064          .ADDRESS INPUT_NAM_RESULT                                       ;
      00  00068          .BYTE   0                                                       ;
      00  00069          .BYTE   0                                                       ;
      FF  0006A          .BYTE   -1                                                      ;
      00  0006B          .BYTE   0                                                       ;
00000000' 0(             .ADDRESS INPUT_NAM_EXPANDED                                     ;
00000000  0 ,70          .LONG   0                                                       ;
    0000#  00074          .WORD   0[8]                                                    ;
    0000#  00084          .WORD   0[3]                                                    ;
    0000#  0008A          .WORD   0[3]                                                    ;
00000000  00090          .LONG   0                                                       ;
00000000  00094          .LONG   0                                                       ;
      00  00098          .BYTE   0                                                       ;
      00  00099          .BYTE   0                                                       ;
      00  0009A          .BYTE   0                                                       ;
      00  0009B          .BYTE   0                                                       ;
      00  0009C          .BYTE   0                                                       ;
      00  0009D          .BYTE   0                                                       ;
      00# 0009E          .BYTE   0[2]                                                    ;
00000000  000A0          .LONG   0                                                       ;
00000000  000A4          .LONG   0                                                       ;
00000000  000A8          .LONG   0                                                       ;
00000000  000AC          .LONG   0                                                       ;
00000000  000B0          .LONG   0                                                       ;
00000000  000B4          .LONG   0                                                       ;
00000000# 000B8          .LONG   0[2]                                                    ;
      02  000C0 OUTPUT_NAM_BLK::                                                        ;
                         .BYTE   2                                                       ;
      60  000C1          .BYTE   96                                                      ;
      FF  000C2          .BYTE   -1                                                      ;
      00  000C3          .BYTE   0                                                       ;
00000000' 000C4          .ADDRESS OUTPUT_NAM_RESULT                                      ;
```

```
      00  000C8          .BYTE   0                                                ;
      00  000C9          .BYTE   0                                                ;
      FF  000CA          .BYTE   -1                                               ;
      00  000CB          .BYTE   0                                                ;
00000000' 000CC          .ADDRESS OUTPUT_NAM_EXPANDED                             ;
00000000' 000D0          .ADDRESS INPUT_NAM                                       ;
    0000# 000D4          .WORD   0[8]                                             ;
    0000# 000E4          .WORD   0[3]                                             ;
    0000# 000EA          .WORD   0[3]                                             ;
00000000  000F0          .LONG   0                                                ;
00000000  000F4          .LONG   0                                                ;
      00  000F8          .BYTE   0                                                ;
      00  000F9          .BYTE   0                                                ;
      00  000FA          .BYTE   0                                                ;
      00  000FB          .BYTE   0                                                ;
      00  000FC          .BYTE   0                                                ;
      00  000FD          .BYTE   0                                                ;
      00# 000FE          .BYTE   0[2]                                             ;
00000000  00100          .LONG   0                                                ;
00000000  00104          .LONG   0                                                ;
00000000  00108          .LONG   0                                                ;
00000000  0010C          .LONG   0                                                ;
00000000  00110          .LONG   0                                                ;
00000000  00114          .LONG   0                                                ;
00000000# 00118          .LONG   0[2]                                             ;
      02  00120 REJECTED_NAM::
                         .BYTE   2                                                ;
      60  00121          .BYTE   96                                               ;
      FF  00122          .BYTE   -1                                               ;
      00  00123          .BYTE   0                                                ;
00000000' 00124          .ADDRESS REJECTED_NAM_RESULT                             ;
      00  00128          .BYTE   0                                                ;
      00  00129          .BYTE   0                                                ;
      FF  0012A          .BYTE   -1                                               ;
      00  0012B          .BYTE   0                                                ;
00000000' 0012C          .ADDRESS REJECTED_NAM_EXPANDED                           ;
00000000' 00130          .ADDRESS INPUT_NAM                                       ;
    0000# 00134          .WORD   0[8]                                             ;
    0000# 00144          .WORD   0[3]                                             ;
    0000# 0014A          .WORD   0[3]                                             ;
00000000  00150          .LONG   0                                                ;
00000000  00154          .LONG   0                                                ;
      00  00158          .BYTE   0                                                ;
      00  00159          .BYTE   0                                                ;
      00  0015A          .BYTE   0                                                ;
      00  0015B          .BYTE   0                                                ;
      00  0015C          .BYTE   0                                                ;
      00  0015D          .BYTE   0                                                ;
      00# 0015E          .BYTE   0[2]                                             ;
00000000  00160          .LONG   0                                                ;
00000000  00164          .LONG   0                                                ;
00000000  00168          .LONG   0                                                ;
00000000  0016C          .LONG   0                                                ;
00000000  00170          .LONG   0                                                ;
00000000  00174          .LONG   0                                                ;
00000000# 00178          .LONG   0[2]                                             ;
      1D  00180 INPUT_XABFHC::
```

```
                        .BYTE   29                                              ;
      2C  00181         .BYTE   44                                              ;
    0000  00182         .WORD   0                                               ;
00000000  00184         .LONG   0                                               ;
00000000# 00188         .LONG   0[9]                                            ;
      03  001AC INPUT_FAB::                                                    ;
                        .BYTE   3                                               ;
      50  001AD         .BYTE   80                                              ;
    0000  001AE         .WORD   0                                               ;
00000040  001B0         .LONG   64                                              ;
00000000  001B4         .LONG   0                                               ;
00000000  001B8         .LONG   0                                               ;
00000000  001BC         .LONG   0                                               ;
    0000  001C0         .WORD   0                                               ;
      02  001C2         .BYTE   2                                               ;
      41  001C3         .BYTE   65                                              ;
00000000  001C4         .LONG   0                                               ;
      00  001C8         .BYTE   0                                               ;
      00  001C9         .BYTE   0                                               ;
      00  001CA         .BYTE   0                                               ;
      02  001CB         .BYTE   2                                               ;
00000000  001CC         .LONG   0                                               ;
00000000' 001D0         .ADDRESS INPUT_XABFHC                                   ;
00000000' 001D4         .ADDRESS INPUT_NAM                                      ;
00000000  001D8         .LONG   0                                               ;
00000000' 001DC         .ADDRESS P.AAA                                          ;
      00  001E0         .BYTE   0                                               ;
      0A  001E1         .BYTE   10                                              ;
    0000  001E2         .WORD   0                                               ;
00000000  001E4         .LONG   0                                               ;
    0000  001E8         .WORD   0                                               ;
      00  001EA         .BYTE   0                                               ;
      00  001EB         .BYTE   0                                               ;
00000000  001EC         .LONG   0                                               ;
00000000  001F0         .LONG   0                                               ;
    0000  001F4         .WORD   0                                               ;
      00  001F6         .BYTE   0                                               ;
      00  001F7         .BYTE   0                                               ;
00000000  001F8         .LONG   0                                               ;
      01  001FC INPUT_RAB::                                                    .
                        .BYTE   1
      44  001FD         .BYTE   68                                              ;
    0000  001FE         .WORD   0                                               ;
00000200  00200         .LONG   512                                             ;
00000000  00204         .LONG   0                                               ;
00000000  00208         .LONG   0                                               ;
    0000# 0020C         .WORD   0[3]                                            ;
    0000  00212         .WORD   0                                               ;
000810B2  00214         .LONG   528562                                          ;
    0000  00218         .WORD   0                                               ;
      00  0021A         .BYTE   0                                               ;
      00  0021B         .BYTE   0                                               ;
    0200  0021C         .WORD   512                                             ;
    0000  0021E         .WORD   0                                               ;
00000000  00220         .LONG   0                                               ;
00000000  00224         .LONG   0                                               ;
00000000  00228         .LONG   0                                               ;
```

```
00000000  0022C          .LONG   0                                                       ;
      00  00230          .BYTE   0                                                       ;
      00  00231          .BYTE   0                                                       ;
      02  00232          .BYTE   2                                                       ;
      10  00233          .BYTE   16                                                      ;
00000000  00234          .LONG   0                                                       ;
00000000' 00238          .ADDRESS INPUT_FAB                                              ;
00000000  0023C          .LONG   0                                                       ;
      03  00240 OUTPUT_FAB::                                                            ;
                         .BYTE   3                                                       ;
      50  00241          .BYTE   80                                                      ;
    0000  00242          .WORD   0                                                       ;
20000040  00244          .LONG   536870976                                               ;
00000000  00248          .LONG   0                                                       ;
00000000  0024C          .LONG   0                                                       ;
00000000  00250          .LONG   0                                                       ;
    0000  00254          .WORD   0                                                       ;
      02  00256          .BYTE   2                                                       ;
      00  00257          .BYTE   0                                                       ;
000810A2  00258          .LONG   528546                                                  ;
      00  0025C          .BYTE   0                                                       ;
      00  0025D          .BYTE   0                                                       ;
      00  0025E          .BYTE   0                                                       ;
      02  0025F          .BYTE   2                                                       ;
00000000  00260          .LONG   0                                                       ;
00000000  00264          .LONG   0                                                       ;
00000000' 00268          .ADDRESS OUTPUT_NAM_BLK                                         ;
00000000  0026C          .LONG   0                                                       ;
00000000' 00270          .ADDRESS DATEXT                                                 ;
      00  00274          .BYTE   0                                                       ;
      04  00275          .BYTE   4                                                       ;
    0000  00276          .WORD   0                                                       ;
00000000  00278          .LONG   0                                                       ;
    0000  0027C          .WORD   0                                                       ;
      00  0027E          .BYTE   0                                                       ;
      00  0027F          .BYTE   0                                                       ;
00000000  00280          .LONG   0                                                       ;
00000000  00284          .LONG   0                                                       ;
    0000  00288          .WORD   0                                                       ;
      00  0028A          .BYTE   0                                                       ;
      00  0028B          .BYTE   0                                                       ;
00000000  0028C          .LONG   0                                                       ;
      01  00290 OUTPUT_RAB::                                                            ;
                         .BYTE   1                                                       ;
      44  00291          .BYTE   68                                                      ;
    0000  00292          .WORD   0                                                       ;
00000000  00294          .LONG   0                                                       ;
00000000  00298          .LONG   0                                                       ;
00000000  0029C          .LONG   0                                                       ;
    0000# 002A0          .WORD   0[3]                                                    ;
    0000  002A6          .WORD   0                                                       ;
00081002  002A8          .LONG   528594                                                  ;
    0000  002AC          .WORD   0                                                       ;
      00  002AE          .BYTE   0                                                       ;
      00  002AF          .BYTE   0                                                       ;
    0000  002B0          .WORD   0                                                       ;
    0000  002B2          .WORD   0                                                       ;
```

```
00000000  002B4          .LONG   0                                                       ;
00000000  002B8          .LONG   0                                                       ;
00000000  002BC          .LONG   0                                                       ;
00000000  002C0          .LONG   0                                                       ;
       00  002C4          .BYTE   0                                                       ;
       00  002C5          .BYTE   0                                                       ;
       00  002C6          .BYTE   0                                                       ;
       00  002C7          .BYTE   0                                                       ;
00000000  002C8          .LONG   0                                                       ;
00000000' 002CC          .ADDRESS OUTPUT_FAB                                             ;
00000000  002D0          .LONG   0                                                       ;
       03  002D4 REJECTED_FAB::
                         .BYTE   3                                                       ;
       50  002D5          .BYTE   80                                                      ;
     0000  002D6          .WORD   0                                                       ;
20000040  002D8          .LONG   536870976                                               ;
00000000  002DC          .LONG   0                                                       ;
00000000  002E0          .LONG   0                                                       ;
00000000  002E4          .LONG   0                                                       ;
     0000  002E8          .WORD   0                                                       ;
       02  002EA          .BYTE   2                                                       ;
       00  002EB          .BYTE   0                                                       ;
000810A2  002EC          .LONG   528546                                                  ;
       00  002F0          .BYTE   0                                                       ;
       00  002F1          .BYTE   0                                                       ;
       00  002F2          .BYTE   0                                                       ;
       02  002F3          .BYTE   2                                                       ;
00000000  002F4          .LONG   0                                                       ;
00000000  002F8          .LONG   0                                                       ;
00000000' 002FC          .ADDRESS REJECTED_NAM                                           ;
00000000  00300          .LONG   0                                                       ;
00000000' 00304          .ADDRESS P.AAB                                                  ;
       00  00308          .BYTE   0                                                       ;
       04  00309          .BYTE   4                                                       ;
     0000  0030A          .WORD   0                                                       ;
00000000  0030C          .LONG   0                                                       ;
     0000  00310          .WORD   0                                                       ;
       00  00312          .BYTE   0                                                       ;
       00  00313          .BYTE   0                                                       ;
00000000  00314          .LONG   0                                                       ;
00000000  00318          .LONG   0                                                       ;
     0000  0031C          .WORD   0                                                       ;
       00  0031E          .BYTE   0                                                       ;
       00  0031F          .BYTE   0                                                       ;
00000000  00320          .LONG   0                                                       ;
       01  00324 REJECTED_RAB::
                         .BYTE   1                                                       ;
       44  00325          .BYTE   68                                                      ;
     0000  00326          .WORD   0                                                       ;
00000400  00328          .LONG   1024                                                    ;
00000000  0032C          .LONG   0                                                       ;
00000000  00330          .LONG   0                                                       ;
     0000# 00334          .WORD   0[3]                                                    ;
     0000  0033A          .WORD   0                                                       ;
000810D2  0033C          .LONG   528594                                                  ;
     0000  00340          .WORD   0                                                       ;
       00  00342          .BYTE   0                                                       ;
```

```
                        00  00343          .BYTE   0                                                       ;
                      0000  00344          .WORD   0                                                       ;
                      0000  00346          .WORD   0                                                       ;
                  00000000  00348          .LONG   0                                                       ;
                  00000000  0034C          .LONG   0                                                       ;
                  00000000  00350          .LONG   0                                                       ;
                  00000000  00354          .LONG   0                                                       ;
                        00  00358          .BYTE   0                                                       ;
                        00  00359          .BYTE   0                                                       ;
                        02  0035A          .BYTE   2                                                       ;
                        10  0035B          .BYTE   16                                                      ;
                  00000000  0035C          .LONG   0                                                       ;
                  00000000' 00360          .ADDRESS REJECTED_FAB                                           ;
                  00000000  00364          .LONG   0                                                       ;
                  00000000' 00368 OUTPUT_NAM::
                                           .ADDRESS OUTPUT_NAM_BLK                                         ;

                                           .EXTRN  LOG_FILENAME, OPEN_OUT_FILE
                                           .EXTRN  PARSE_OUTPUT_FILES
                                           .EXTRN  WRITE_MSG, LSTLUN_RAB_ADDRESS
                                           .EXTRN  SYS$OUTPUT_RAB_ADDRESS

                                           .PSECT  $CODE,NOWRT,  PIC,2

                      003C  00000          .ENTRY  LOG_FILENAME, Save R2,R3,R4,R5                          ; 0604
5E                  08  C2  00002          SUBL2   #8, SP                                                  ;
50          04      AC  D0  00005          MOVL    RMS, R0                                                 ; 0653
                    52  D4  00009          CLRL    R2                                                      ;
01                      60  91  0000B       CMPB    (R0), #1                                                ;
                        08  12  0000E       BNEQ    1$                                                      ;
                        52  D6  00010       INCL    R2                                                      ;
                    51      3C  A0  D0  00012  MOVL  60(R0), FAB                                           ; 0655
                        03  11  00016       BRB     2$                                                      ; 0656
                    51          50  D0  00018  1$:  MOVL  R0, FAB                                          ; 0661
                    53      08  A0  D0  0001B  2$:  MOVL  8(R0), RMS_STS                                   ; 0662
                    55      0C  A0  D0  0001F      MOVL  12(R0), RMS_STV                                   ; 0663
                    54      18  A0  D0  00023      MOVL  24(R0), RMS_CTX                                   ; 0664
                    50      28  A1  D0  00027      MOVL  40(FAB), NAM                                      ; 0667
                    1A          52  E9  0002B      BLBC  R2, 3$                                            ; 0676
0001827A            8F          53  D1  0002E      CMPL  RMS_STS, #98938                                   ; 0677
                        11  12  00035       BNEQ    3$                                                      ;
000810B2            8F          54  D1  00037      CMPL  RMS_CTX, #528562                                  ; 0678
                        08  12  0003E       BNEQ    3$                                                      ;
                    50  0001827A    8F  D0  00040  MOVL  #98938, R0                                         ; 0679
                        04  00047       RET                                                             ;
                    03      A0  95  00048  3$:  TSTB  3(NAM)                                               ; 0691
                        0B  13  0004B       BEQL    4$                                                      ;
                6E  03      A0  9A  0004D      MOVZBL  3(NAM), DESC                                        ; 0693
          04    AE  04      A0  D0  00051      MOVL    4(NAM), DESC+4                                      ; 0694
                        19  11  00056       BRB     6$                                                      ; 0691
                    0B      A0  95  00058  4$:  TSTB  11(NAM)                                              ; 0697
                        0B  13  0005B       BEQL    5$                                                      ;
                6E  0B      A0  9A  0005D      MOVZBL  11(NAM), DESC                                       ; 0699
          04    AE  0C      A0  D0  00061      MOVL    12(NAM), DESC+4                                     ; 0700
                        09  11  00066       BRB     6$                                                      ; 0697
                6E  34      A1  9A  00068  5$:  MOVZBL  52(FAB), DESC                                      ; 0704
          04    AE  2C      A1  D0  0006C      MOVL    44(FAB), DESC+4                                     ; 0705
```

```
                                 28  BB 00071 6$:    PUSHR   #^M<R3,R5>                      ; 0718
                              08 AE  9F 00073        PUSHAB  DESC                            ; 0717
                                 01  DD 00076        PUSHL   #1                              ;
                                 54  DD 00078        PUSHL   RMS_CTX                         ;
                    00000000G    00  FB 0007A        CALLS   #5, LIB$SIGNAL                  ;
                                 50  D0 00081        MOVL    RMS_STS, R0                     ; 0722
                                     04 00084        RET                                     ; 0724

; Routine Size:  133 bytes,    Routine Base:  $CODE + 0000
```

```
;  314     0725  1 UNDECLARE PARSE_OUTPUT_FILES;
;  315     0726  1
;  316     0727  1 GLOBAL ROUTINE PARSE_OUTPUT_FILES =
;  317     0728  1
;  318     0729  1 !----
;  319     0730  1 !
;  320     0731  1 ! Functional description
;  321     0732  1 !
;  322     0733  1 !       This routine is called to process output files.
;  323     0734  1 !       If the files are binary (/BINARY or /REJECTED)
;  324     0735  1 !       RMS is used, else fortran io is used.
;  325     0736  1 !
;  326     0737  1 ! Input parameters
;  327     0738  1 !
;  328     0739  1 !       None
;  329     0740  1 !
;  330     0741  1 ! Output parameters
;  331     0742  1 !
;  332     0743  1 !       Any errors encountered are RETURNed immediately.
;  333     0744  1 !       TRUE is returned on a normal exit.
;  334     0745  1 !
;  335     0746  1 !----
;  336     0747  1
;  337     0748  2 BEGIN
;  338     0749  2
;  339     0750  2 LOCAL
;  340     0751  2         desc:   vector [2, long];       ! Temporary string descriptor
;  341     0752  2
;  342     0753  2 OWN
;  343     0754  2         output_desc:    $bblock [dsc$k_d_bln]
;  344     0755  2                         preset([dsc$b_class] = dsc$k_class_d),
;  345     0756  2         rejected_desc:  $bblock [dsc$k_d_bln]
;  346     0757  2                         preset([dsc$b_class] = dsc$k_class_d);
;  347     0758  2
;  348     0759  2
;  349     0760  2
;  350     0761  2 !
;  351     0762  2 !PARSE COMMAND LINE OUTPUTS ---
;  352     0763  2 ! Parse the /OUTPUT, /BINARY and /REJECTED qualifiers.  Store any output
;  353     0764  2 ! file names obtained in the FAB for future processing.
;  354     0765  2 !
;  355     0766  2
;  356     0767  3 If GET_VALUE ( 'BINARY', output_desc )
;  357     0768  3
;  358     0769  3 then    BEGIN
;  359     0770  3         Output_fab [fab$b_fns] = .output_desc [dsc$w_length];
;  360     0771  3         Output_fab [fab$l_fna] = .output_desc [dsc$a_pointer];
;  361     0772  3
;  362   P 0773  3         CALL_FUNCTION ($create (        ! Call RMS with
;  363   P 0774  3                 fab = output_fab,       !  -address of FAB
;  364     0775  3                 err = log_filename));   !  -error action routine
;  365     0776  3
;  366   P 0777  3         CALL_FUNCTION ($connect (       ! Call RMS with
;  367   P 0778  3                 rab = output_rab,       !  -address of RAB
;  368     0779  3                 err = log_filename));   !  -error action routine
;  369     0780  3         END
;  370     0781  3
```

```
;  371          0782  2 else
;  372          0783  2         Begin
;  373          0784  2         GET_VALUE ('OUTPUT', output_desc);
;  374          0785  2
;  375          0786  2         output_fab [fab$b_fns] = .output_desc [dsc$w_length];
;  376          0787  2         output_fab [fab$l_fna] = .output_desc [dsc$a_pointer];
;  377          0788  2
;  378          0789  2         Open_out_file ( output_desc );
;  379          0790  2         End ;
;  380          0791  2
;  381          0792  2
;  382          0793  2 If GET_VALUE ('REJECTED', rejected_desc) then! /REJECTED value
;  383          0794  2         BEGIN
;  384          0795  2         rejected_fab [fab$b_fns] = .rejected_desc [dsc$w_length];
;  385          0796  2         rejected_fab [fab$l_fna] = .rejected_desc [dsc$a_pointer];
;  386        P 0797  2         CALL_FUNCTION ($create (          ! Call RMS with
;  387        P 0798  2                 fab = rejected_fab,       !  -address of FAB
;  388          0799  2                 err = log_filename));     !  -error action routine
;  389          0800  2
;  390        P 0801  2         CALL_FUNCTION ($connect (         ! Call RMS with
;  391        P 0802  2                 rab = rejected_rab,       !  -address of RAB
;  392          0803  2                 err = log_filename));     !  -error action routine
;  393          0804  2         END;
;  394          0805  2
;  395          0806  2 RETURN TRUE;
;  396          0807  1 END;


                                                        .PSECT  $PLIT,NOWRT,NOEXE,  PIC,2

                                                0000E           .BLKB   2
00  00  59  52  41  4E  49  42                     00010 P.AAD:    .ASCII  \BINARY\<0><0>                  ;
                                                00000006        00018 P.AAC:    .LONG   6                       ;
                                                00000000'       0001C           .ADDRESS P.AAD          ;
00  00  54  55  50  54  55  4F                     00020 P.AAF:    .ASCII  \OUTPUT\<0><0>                  ;
                                                00000006        00028 P.AAE:    .LONG   6                       ;
                                                00000000'       0002C           .ADDRESS P.AAF          ;
44  45  54  43  45  4A  45  52                     00030 P.AAH:    .ASCII  \REJECTED\                      ;
                                                00000008        00038 P.AAG:     LONG   8                       ;
                                                00000000'       0003C           .ADDRESS P.AAH          ;

                                                                 PSECT  $OWN$,NOEXE,  PIC,2

                                                00707           .BLKB   1
                                        00#     00708 OUTPUT_DESC:
                                                                .BYTE   0[3]                            ;
                                        02      00708           .BYTE   2                               ;
                                                0070C           .BLKB   4
                                        00#     00710 REJECTED_DESC:
                                                                .BYTE   0[3]                            ;
                                        02      00713           .BYTE   2                               ;
                                                00714           .BLKB   4

                                                                .EXTRN  CLI$GET_VALUE, SYS$CREATE
                                                                .EXTRN  SYS$CONNECT
```

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

                            01FC 00000  .ENTRY  PARSE_OUTPUT_FILES, Save R2,R3,R4,R5,R6,R7,-  ; 0727
                                                R8                                            ;
                    58 00000000G   00   9E 00002        MOVAB   SYS$CONNECT, R8               ;
                    57 00000000G   00   9E 00009        MOVAB   SYS$CREATE, R7                ;
                    56 00000000G   00   9E 00010        MOVAB   CLI$GET_VALUE, R6             ;
                    55 00C00000'   00   9E 00017        MOVAB   P.AAC, R5                     ;
                    54     FF59    CF   9E 0001E        MOVAB   LOG_FILENAME, R4              ;
                    53 00000000'   00   9E 00023        MOVAB   OUTPUT_DESC, R3               ;
                    52 00000000'   00   9E 0002A        MOVAB   OUTPUT_FAB+52, R2             ;
                    5E                  08   C2 00031    SUBL2   #8, SP                        ;
                                        53   DD 00034    PUSHL   R3                            ; 0767
                                        55   DD 00036    PUSHL   R5                            ;
                    66                  02   FB 00038    CALLS   #2, CLI$GET_VALUE             ;
                    1F                  50   E9 0003B    BLBC    R0, 1$                        ;
                    62                  63   90 0003E    MOVB    OUTPUT_DESC, OUTPUT_FAB+52    ; 0770
                F8  A2          04      A3   D0 00041    MOVL    OUTPUT_DESC+4, OUTPUT_FAB+44  ; 0771
                                        54   DD 00046    PUSHL   R4                            ; 0775
                    CC          A2      9F   00048      PUSHAB  OUTPUT_FAB                    ;
                    67                  02   FB 0004B    CALLS   #2, SYS$CREATE                ;
                    57                  50   E9 0004E    BLBC    STATUS, 4$                    ;
                                        54   DD 00051    PUSHL   R4                            ; 0779
                    1C          A2      9F   00053      PUSHAB  OUTPUT_RAB                    ;
                    68                  02   FB 00056    CALLS   #2, SYS$CONNECT               ;
                    1A                  50   E8 00059    BLBS    STATUS, 2$                    ;
                                        04   00050      RET                                   ;
                                        53   DD 0005D 1$: PUSHL   R3                            ; 0784
                    10          A5      9F   0005F      PUSHAB  P.AAE                         ;
                    66                  02   FB 00062    CALLS   #2, CLI$GET_VALUE             ;
                    62                  63   90 00065    MOVB    OUTPUT_DESC, OUTPUT_FAB+52    ; 0786
                F8  A2          04      A3   D0 00068    MOVL    OUTPUT_DESC+4, OUTPUT_FAB+44  ; 0787
                                        53   DD 0006D    PUSHL   R3                            ; 0789
        00000000G   00                  01   FB 0006F    CALLS   #1, OPEN_OUT_FILE             ;
                            08          A3   9F 00076 2$: PUSHAB  REJECTED_DESC                 ; 0793
                            20          A5   9F 00079    PUSHAB  P.AAG                         ;
                    66                  02   FB 0007C    CALLS   #2, CLI$GET_VALUE             ;
                    23                  50   E9 0007F    BLBC    R0, 3$                        ;
                0094    C2          08      A3   90 00082    MOVB    REJECTED_DESC, REJECTED_FAB+52    ; 0795
                008C    C2          0C      A3   D0 00088    MOVL    REJECTED_DESC+4, REJECTED_FAB+44  ; 0796
                                        54   DD 0008E    PUSHL   R4                            ; 0799
                            60          A2   9F 00090    PUSHAB  REJECTED_FAB                  ;
                    67                  02   FB 00093    CALLS   #2, SYS$CREATE                ;
                    0F                  50   E9 00096    BLBC    STATUS, 4$                    ;
                                        54   DD 00099    PUSHL   R4                            ; 0803
                            00B0        C2   9F 0009B    PUSHAB  REJECTED_RAB                  ;
                    68                  02   FB 0009F    CALLS   #2, SYS$CONNECT               ;
                    03                  50   E9 000A2    BLBC    STATUS, 4$                    ;
                    50                  01   D0 000A5 3$: MOVL    #1, R0                        ; 0806
                                        04   000A8 4$:    RET                                   ; 0807

; Routine Size:  169 bytes,    Routine Base:  $CODE + 0085
```

```
;  398          0808  1 UNDECLARE WRITE_MSG;
;  399          0809  1
;  400          0810  1 Global routine WRITE_MSG (msg_desc, output_flag : ref vector) =
;  401          0811  1 !---
;  402          0812  1 !
;  403          0813  1 !       This routine writes a message to the output stream.
;  404          0814  1 !
;  405          0815  1 ! Inputs:
;  406          0816  1 !
;  407          0817  1 !       msg_desc = Address of descriptor for the message
;  408          0818  1 !    output_flag = Address of flag
;  409          0819  1 !
;  410          0820  1 ! Outputs:
;  411          0821  1 !
;  412          0822  1 !
;  413          0823  1 !---
;  414          0824  2 Begin
;  415          0825  2
;  416          0826  2 Local
;  417          0827  2         Rab_address : REF BLOCK[,BYTE];
;  418          0828  2
;  419          0829  2 Map
;  420          0830  2         Msg_desc : REF BLOCK[,BYTE];
;  421          0831  2
;  422          0832  2 If .Lstlun_rab_address NEQ 0 then
;  423          0833  3    Begin
;  424          0834  3    Lstlun_rab_address[rab$l_rbf] = .msg_desc[dsc$a_pointer];
;  425          0835  3    Lstlun_rab_address[rab$w_rsz] = .msg_desc[dsc$w_length];
;  426          0836  3    Rab_address = .lstlun_rab_address;
;  427          0837  3    End
;  428          0838  2 else
;  429          0839  3    Begin
;  430          0840  3    If .output_flag EQLU 1 then return true;
;  431          0841  3    Sys$output_rab_address[rab$l_rbf] = .msg_desc[dsc$a_pointer];
;  432          0842  3    Sys$output_rab_address[rab$w_rsz] = .msg_desc[dsc$w_length];
;  433          0843  3    Rab_address = .sys$output_rab_address;
;  434          0844  2    End;
;  435          0845  2
;  436          0846  2 Rab_address[rab$l_ctx] = msg$_writeerr;
;  437          0847  2 CALL_FUNCTION ($put (rab = .rab_address, err = log_filename));
;  438          0848  2
;  439          0849  2 Return true;
;  440          0850  1 End;


                                                        .EXTRN  SYS$PUT

                                    0000 00000          .ENTRY  WRITE_MSG, Save nothing                 ; 0810
                51          04  AC  D0 00002          MOVL    MSG_DESC, R1                            ; 0834
                50 00000000G    00  D0 00006          MOVL    LSTLUN_RAB_ADDRESS, R0                  ; 0832
                                0D  12 0000D          BNEQ    1$
                01          08  AC  D1 0000F          CMPL    OUTPUT_FLAG, #1                         ; 0840
                                2C  13 00013          BEQL    2$
                50 00000000G    00  D0 00015          MOVL    SYS$OUTPUT_RAB_ADDRESS, R0              ; 0841
            28  A0          04  A1  D0 0001C 1$:      MOVL    4(R1), 40(R0)
            22  A0          04  BC  B0 00021          MOVW    @MSG_DESC, 34(R0)                       ; 0842
```

```
                    51              50  DO 00026          MOVL    R0, RAB_ADDRESS                         ; 0843
                18  A1 000810D2     8F  DO 00029          MOVL    #528594, 24(RAB_ADDRESS)                ; 0846
                        FE9D        CF  9F 00031          PUSHAB  LOG_FILENAME                            ; 0847
                                    51  DD 00035          PUSHL   RAB_ADDRESS                             ;
            000000006  00           02  FB 00037          CALLS   #2, SYS$PUT                             ;
                       03           50  E9 0003E          BLBC    STATUS, 3$                              ;
                       50           01  DO 00041 2$:      MOVL    #1, R0                                  ; 0849
                                        04 00044 3$:      RET                                             ; 0850

; Routine Size:  69 bytes,    Routine Base:  $CODE + 012E
```

```
;  442          0851  1 END
;  443          0852  0 ELUDOM



                                                        .EXTRN  LIB$SIGNAL

;                                PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $OWN$                         1816  NOVEC,  WRT,  RD  NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  $GLOBAL$                       876  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  $PLIT                           64  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  $CODE                          371  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                              Library Statistics
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619        84         0        1000        00:01.9



;                                       COMMAND QUALIFIERS

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

; Size:         371 code + 2756 data bytes
; Run Time:         00:20.9
; Elapsed Time:     00:40.8
; Lines/CPU Min:     2447
; Lexemes/CPU-Min: 40713
; Memory Used:  164 pages
; Compilation Complete
```

GETCODE.LIS

INITPROC1.LIS

INITREAL.LIS

EXECIMAGE.LIS

ERFSUMM.LIS

INITBUS.LIS

INITPROC4.LIS

KMS3271.LIS

ERFTAPEVE.LIS

HEADER.LIS

FILES.LIS

ERLLOGSYS.LIS

INITPROC2.LIS

INIT_TAPE.LIS

ERLLOGMSG.LIS

INITDISK.LIS

INITPROC5.LIS

IMAGELOAD.LIS

INITPROC3.LIS

INTERVENE.LIS

ERFRTVEC.LIS

ERFSUMVEC.LIS