```
MMM          MMM     SSSSSSSSSSS     GGGGGGGGGGG  FFFFFFFFFFFFFFF      IIIIIIIII     LLL
MMM          MMM     SSSSSSSSSSS     GGGGGGGGGGG  FFFFFFFFFFFFFFF      IIIIIIIII     LLL
MMM          MMM     SSSSSSSSSSS     GGGGGGGGGGG  FFFFFFFFFFFFFFF      IIIIIIIII     LLL
MMMMMM    MMMMMM  SSS             GGG             FFF                     III        LLL
MMMMMM    MMMMMM  SSS             GGG             FFF                     III        LLL
MMMMMM    MMMMMM  SSS             GGG             FFF                     III        LLL
MMM   MMM    MMM  SSS             GGG             FFF                     III        LLL
MMM   MMM    MMM  SSS             GGG             FFF                     III        LLL
MMM   MMM    MMM  SSS             GGG             FFF                     III        LLL
MMM          MMM     SSSSSSSSS    GGG             FFFFFFFFFFF             III        LLL
MMM          MMM     SSSSSSSSS    GGG             FFFFFFFFFFF             III        LLL
MMM          MMM     SSSSSSSSS    GGG             FFFFFFFFFFF             III        LLL
MMM          MMM             SSS  GGG   GGGGGGGGG  FFF                     III        LLL
MMM          MMM             SSS  GGG   GGGGGGGGG  FFF                     III        LLL
MMM          MMM             SSS  GGG   GGGGGGGGG  FFF                     III        LLL
MMM          MMM             SSS  GGG         GGG  FFF                     III        LLL
MMM          MMM             SSS  GGG         GGG  FFF                     III        LLL
MMM          MMM             SSS  GGG         GGG  FFF                     III        LLL
MMM          MMM  SSSSSSSSSSS     GGGGGGGGG    FFF                 IIIIIIIII     LLLLLLLLLLLLLLL
MMM          MMM  SSSSSSSSSSS     GGGGGGGGG    FFF                 IIIIIIIII     LLLLLLLLLLLLLLL
MMM          MMM  SSSSSSSSSSS     GGGGGGGGG    FFF                 IIIIIIIII     LLLLLLLLLLLLLLL
```

FILEID**MAIN

```
MM      MM    AAAAAA      IIIIII    NN      NN
MM      MM    AAAAAA      IIIIII    NN      NN
MMMM  MMMM  AA      AA      II      NN      NN
MMMM  MMMM  AA      AA      II      NN      NN
MM  MM  MM  AA      AA      II      NNNN    NN
MM  MM  MM  AA      AA      II      NNNNN   NN
MM      MM  AA      AA      II      NN  NN  NN
MM      MM  AA      AA      II      NN  NN  NN
MM      MM  AAAAAAAAAA      II      NN    NNNN
MM      MM  AAAAAAAAAA      II      NN    NNNN
MM      MM  AA      AA      II      NN      NN      ....
MM      MM  AA      AA      II      NN      NN      ....
MM      MM  AA      AA    IIIIII    NN      NN      ....
MM      MM  AA      AA    IIIIII    NN      NN      ....

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 main (%TITLE 'MESSAGE FILE COMPILER' IDENT = 'V04-000', MAIN = start) =
;     2        0002  1 BEGIN
;     3        0003  1
;     4        0004  1 !
;     5        0005  1 !*****************************************************************************
;     6        0006  1 !*                                                                           *
;     7        0007  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;     8        0008  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;     9        0009  1 !*  ALL RIGHTS RESERVED.                                                     *
;    10        0010  1 !*                                                                           *
;    11        0011  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    12        0012  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    13        0013  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    14        0014  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    15        0015  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    16        0016  1 !*  TRANSFERRED.                                                             *
;    17        0017  1 !*                                                                           *
;    18        0018  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    19        0019  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    20        0020  1 !*  CORPORATION.                                                             *
;    21        0021  1 !*                                                                           *
;    22        0022  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    23        0023  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    24        0024  1 !*                                                                           *
;    25        0025  1 !*                                                                           *
;    26        0026  1 !*****************************************************************************
;    27        0027  1
;    28        0028  1 !++
;    29        0029  1 ! FACILITY:  Message compiler
;    30        0030  1 !
;    31        0031  1 ! ABSTRACT:
;    32        0032  1 !
;    33        0033  1 !       This compiler translated message definition language
;    34        0034  1 !       into object modules
;    35        0035  1 !
;    36        0036  1 ! ENVIRONMENT:
;    37        0037  1 !
;    38        0038  1 !       VAX/VMS operating system. unprivileged user mode,
;    39        0039  1 !
;    40        0040  1 ! AUTHOR:  Tim Halvorsen, Nov 1979
;    41        0041  1 !
;    42        0042  1 ! Modified by:
;    43        0043  1 !
;    44        0044  1 !       V03-006 GJA0088         Greg Awdziewicz         11-Juj-1984
;    45        0045  1 !               - Make /list the default for batch mode compiles.
;    46        0046  1 !               (Change check for presence of global use of positional
;    47        0047  1 !               qualifiers.)
;    48        0048  1 !               - Centralize ASCID strings by using SD_ macro.
;    49        0049  1 !
;    50        0050  1 !       V03-005 GJA0074         Greg Awdziewicz         6-Mar-1984
;    51        0051  1 !               - Change output file handling from using $Parse to use
;    52        0052  1 !               the new version of LIB$find_file routine which handles
;    53        0053  1 !               multiple input file stickyness.
;    54        0054  1 !               - Change input file searching to use the new sticky-context
;    55        0055  1 !               argument in the LIB$file_scan routine.
;    56        0056  1 !
;    57        0057  1 !       V03-004 GJA0066         Greg Awdziewicz         3-Feb-1984
```

```
;    58          0058  1 !               - Reset the FAB$V_OFP bit for the output qualifiers
;    59          0059  1 !               which have values specified so that Message's treatment
;    60          0060  1 !               is consistent with other language processors.
;    61          0061  1 !
;    62          0062  1 !       V03-003 GJA0062         Greg Awdziewicz         9-Jan-1984
;    63          0063  1 !               - Remove references to pseudo qualifier /noObject now that
;    64          0064  1 !               NEG is allowed in CLD as an operator.
;    65          0065  1 !
;    66          0066  1 !       V03-002 GJA0048         Greg Awdziewicz         7-Sep-1983
;    67          0067  1 !               Change to new CLI interface.  (Incorporating previous module,
;    68          0068  1 !               GETQUALS, into this module)
;    69          0069  1 !
;    70          0070  1 !       V03-001 JWT0058         Jim Teague              22-Sep-1982
;    71          0071  1 !               Failure to find input file is FATAL, like other compilers.
;    72          0072  1 !
;    73          0073  1 !       V02-006 JWT0011         Jim Teague              15-Feb-1982
;    74          0074  1 !               Special cased the elimination of error summaries when
;    75          0075  1 !               no input files found.
;    76          0076  1 !
;    77          0077  1 !       V02-005 BLS0145         Benn Schreiber           6-Feb-1982
;    78          0078  1 !               Change FMG$FILE_SCAN to LIB$FILE_SCAN
;    79          0079  1 !
;    80          0080  1 !       V02-004 JWT0005         Jim Teague              11-Dec-1981
;    81          0081  1 !               Set FOP=OFP in output fabs to keep output files
;    82          0082  1 !               in the default directory
;    83          0083  1 !
;    84          0084  1 !       V02-003 JWT0002         Jim Teague              06-NOV-1981
;    85          0085  1 !               Added /SDL capability
;    86          0086  1 !
;    87          0087  1 !       V02-002 PCG0002         Peter George            24-Aug-1981
;    88          0088  1 !               Allow use of SYS$INPUT as input file spec.
;    89          0089  1 !
;    90          0090  1 !       V02-001 PCG0001         Peter George            29-Dec-1980
;    91          0091  1 !               Add /MDL functionality.
;    92          0092  1 !
;    93          0093  1 !--
;    94          0094  1
;    95          0095  1 !
;    96          0096  1 ! Include files
;    97          0097  1 !
;    98          0098  1
;    99          0099  1 LIBRARY 'SYS$LIBRARY:STARLET.L32';      ! VAX/VMS common definitions
;   100          0100  1
;   101          0101  1 SWITCHES LIST(REQUIRE);                 ! Print require file
;   102          0102  1 REQUIRE 'SRC$:MSG.REQ';                 ! Command definitions
```

```
;  R0103  1     %SBTTL 'Required file for MESSAGE source modules'
;  R0104  1     !---
;  R0105  1     !
;  R0106  1     !       Require file for all modules in the message compiler
;  R0107  1     !
;  R0108  1     ! Version:      'V04-000'
;  R0109  1     !
;  R0110  1     !****************************************************************************
;  R0111  1     !*                                                                          *
;  R0112  1     !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;  R0113  1     !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;  R0114  1     !*  ALL RIGHTS RESERVED.                                                    *
;  R0115  1     !*                                                                          *
;  R0116  1     !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;  R0117  1     !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;  R0118  1     !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;  R0119  1     !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;  R0120  1     !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;  R0121  1     !*  TRANSFERRED.                                                            *
;  R0122  1     !*                                                                          *
;  R0123  1     !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;  R0124  1     !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;  R0125  1     !*  CORPORATION.                                                            *
;  R0126  1     !*                                                                          *
;  R0127  1     !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;  R0128  1     !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;  R0129  1     !*                                                                          *
;  R0130  1     !*                                                                          *
;  R0131  1     !****************************************************************************
;  R0132  1     !
;  R0133  1     !++
;  R0134  1     ! FACILITY:  MESSAGE compiler
;  R0135  1     !
;  R0136  1     ! ABSTRACT:
;  R0137  1     !
;  R0138  1     !       This is the common require file for all modules in the
;  R0139  1     !       message compiler.
;  R0140  1     !
;  R0141  1     ! ENVIRONMENT:
;  R0142  1     !
;  R0143  1     !       VAX/VMS operating system, unprivileged user mode utility,
;  R0144  1     !       operates at non-AST level.
;  R0145  1     !
;  R0146  1     ! AUTHOR:  Tim Halvorsen,  Nov 1979
;  R0147  1     !
;  R0148  1     ! MODIFIED BY:
;  R0149  1     !
;  R0150  1     !       V03-002 GJA0052         Greg Awdziewicz         23-Nov-1983
;  R0151  1     !               Add size literal and field definitions for the queue of input
;  R0152  1     !               file specifications.
;  R0153  1     !               Add subtitles.
;  R0154  1     !               Add error code for parsefail.
;  R0155  1     !
;  R0156  1     !       V03-001 BLS0168         Benn Schreiber          2-Apr-1982
;  R0157  1     !               Do not require MSGDEF as it is now in STARLET and LIB
;  R0158  1     !
;  R0159  1     !----
```

```
;  R0160  1     %SBTTL 'Internal definitions'
;  R0161  1
;  R0162  1     !
;  R0163  1     !       Define message structure definitions
;  R0164  1     !
;  R0165  1
;  R0166  1     REQUIRE 'LIB$:MSGINT.B32';              ! Message compiler internal definitions
```

```
;  R0167  1        !-----
;  R0168  1        !
;  R0169  1        !       Internal compiler structures
;  R0170  1        !
;  R0171  1        !-----
;  R0172  1        !       Version 'V04-000'
;  R0173  1        !
;  R0174  1        !****************************************************************************
;  R0175  1        !*                                                                          *
;  R0176  1        !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;  R0177  1        !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;  R0178  1        !*  ALL RIGHTS RESERVED.                                                    *
;  R0179  1        !*                                                                          *
;  R0180  1        !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;  R0181  1        !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;  R0182  1        !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;  R0183  1        !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;  R0184  1        !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;  R0185  1        !*  TRANSFERRED.                                                            *
;  R0186  1        !*                                                                          *
;  R0187  1        !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;  R0188  1        !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;  R0189  1        !*  CORPORATION.                                                            *
;  R0190  1        !*                                                                          *
;  R0191  1        !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;  R0192  1        !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;  R0193  1        !*                                                                          *
;  R0194  1        !*                                                                          *
;  R0195  1        !****************************************************************************
;  R0196  1
;  R0197  1
;  R0198  1        !
;  R0199  1        !       Define facility definition block
;  R0200  1        !
;  R0201  1
;  R0202  1        !...$FACDEF
;  R0203  1
;  R0204  1        MACRO           FAC$L_LINK      = 0,0,32,0%;            ! LINK TO NEXT IN CHAIN
;  R0205  1        MACRO           FAC$W_NUMBER    = 4,0,16,0%;    ! FACILITY NUMBER
;  R0206  1        MACRO           FAC$B_NAMELEN   = 6,0,8,0%;     ! LENGTH OF FACILITY NAME
;  R0207  1        MACRO           FAC$T_NAME      = 7,0,0,0%;     ! FACILITY NAME (15 CHARS MAXIMUM)
;  R0208  1        LITERAL         FAC$S_NAME      = 15;
;  R0209  1        LITERAL         FAC$C_LENGTH    = 22;
;  R0210  1        LITERAL         FAC$K_LENGTH    = 22;                   ! LENGTH OF BLOCK
;  R0211  1
;  R0212  1        !
;  R0213  1        !       Define message code definition block
;  R0214  1        !
;  R0215  1
;  R0216  1        !...$CODEDEF
;  R0217  1
;  R0218  1        MACRO           CODE$L_LINK     = 0,0,32,0%;            ! LINK TO NEXT IN CHAIN
;  R0219  1        MACRO           CODE$L_NUMBER   = 4,0,32,0%;    ! MESSAGE NUMBER
;  R0220  1        LITERAL         CODE$C_LENGTH   = 8;
;  R0221  1        LITERAL         CODE$K_LENGTH   = 8;                    ! LENGTH OF FIXED PORTION
;  R0222  1        LITERAL         CODE$C_MSG      = 8;
;  R0223  1        LITERAL         CODE$K_MSG      = 8;                    ! MESSAGE RECORD FOLLOWS (MSG STRUCTURE)
```

```
;  R0224  1
;  R0225  1        !
;  R0226  1        !       Define symbol table entry
;  R0227  1        !
;  R0228  1
;  R0229  1        !...$SYMDEF
;  R0230  1
;  R0231  1        MACRO           SYM$L_LINK      = 0,0,32,0%;            ! LINK TO NEXT IN CHAIN
;  R0232  1        MACRO           SYM$L_VALUE     = 4,0,32,0%;            ! VALUE OF SYMBOL
;  R0233  1        MACRO           SYM$B_SYMLEN    = 8,0,8,0%;     ; LENGTH OF SYMBOL NAME
;  R0234  1        MACRO           SYM$T_SYMBOL    = 9,0,0,0%;     ! SYMBOL NAME (31 CHARS MAXIMUM)
;  R0235  1        LITERAL         SYM$S_SYMBOL    = 31;
;  R0236  1        LITERAL         SYM$C_LENGTH    = 40;
;  R0237  1        LITERAL         SYM$K_LENGTH    = 40;                   ! LENGTH OF ENTRY
```

```
;  P0238  1      %SBTTL 'Required file for MESSAGE source modules'
;  R0239  1      !
;  R0240  1      !       Define fields for the queue element for an input filename and
;  R0241  1      !       define a literal for size of the element:
;  R0242  1      !
;  R0243  1
;  R0244  1      FIELD
;  R0245  1          info_queue_flds =
;  R0246  1          SET
;  R0247  1              queue_flink =   [0, 0, 32, 0],          ! Forward link for queue.
;  R0248  1              queue_blink =   [4, 0, 32, 0],          ! Backward link.
;  R0249  1              name_desc =     [8, 0, 0, 0],           ! Descriptor for filename:
;  R0250  1                  name_length =       [8, 0, 16, 0],  ! ... length subfield,
;  R0251  1                  name_addr =         [12, 0, 32, 0]  ! ... address subfield.
;  R0252  1          TES;
;  R0253  1
;  R0254  1      LITERAL
;  R0255  1          queue_element_size= 16;     ! Total (byte)size of queue element.
;  R0256  1
;  R0257  1      !
;  R0258  1      !       Define VMS block structures
;  R0259  1      !
;  R0260  1
;  R0261  1      STRUCTURE
;  R0262  1              bblock [o, p, s, e; n] =
;  R0263  1                      [n]
;  R0264  1                      (bblock+o)<p,s,e>;
```

```
;  R0265  1
;  R0266  1        !
;  R0267  1        ! $SHR_MESSAGES - a macro which defines facility-specific message codes
;  R0268  1        !       which are based on the system-wide shared message codes.
;  R0269  1        !
;  R0270  1        !       $SHR_MESSAGES( name, code, (msg,severity), ... )
;  R0271  1        !
;  R0272  1        !       where:
;  R0273  1        !           "name" is the name of the facility (e.g., COPY)
;  R0274  1        !           "code" is the corresponding facility code (e.g., 103)
;  R0275  1        !           "msg" is the name of the shared message (e.g., BEGIN)
;  R0276  1        !           "severity" is the desired message severity (e.g., 1, 0, 2)
;  R0277  1        !
;  R0278  1
;  R0279  1        MACRO
; MR0280  1            $SHR_MESSAGES( FACILITY_NAME, FACILITY_CODE ) =
; MR0281  1                LITERAL
;  R0282  1                SHR$MSG_IDS( FACILITY_NAME, FACILITY_CODE, %REMAINING ); %,
;  R0283  1
; MR0284  1            SHR$MSG_IDS( FACILITY_NAME, FACILITY_CODE) [ VALUE ] =
;  R0285  1                SHR$MSG_CALC( FACILITY_NAME, FACILITY_CODE, %REMOVE(VALUE) ) %,
;  R0286  1
; MR0287  1            SHR$MSG_CALC( FACILITY_NAME, FACILITY_CODE, MSG_ID, SEVERITY ) =
; MR0288  1                %NAME(FACILITY_NAME,'$_',MSG_ID) = %NAME('SHR$_',MSG_ID) + FACILITY_CODE*65536 +
; MR0289  1                    %IF %DECLARED(%NAME('STS$K_',SEVERITY))
; MR0290  1                        %THEN %NAME('STS$K_',SEVERITY)
;  R0291  1                        %ELSE SEVERITY %FI %;
```

```
;  R0292  1        !
;  R0293  1        !       Equated symbols
;  R0294  1        !
;  R0295  1
;  R0296  1        LITERAL
;  R0297  1            true        = 1,                    ! Boolean true
;  R0298  1            false       = 0,                    ! Boolean false
;  R0299  1            ok          = 1;                    ! Success return code
;  R0300  1
;  R0301  1        !
;  R0302  1        !       Define CLI qualifier flags (see cli_flags)
;  R0303  1        !
;  R0304  1
;  R0305  1        LITERAL
; PR0306  1            $EQULST (QUAL_,,0,1,
; PR0307  1                (listing,),                     ! /LISTING qualifier present
; PR0308  1                (object,),                      ! /OBJECT qualifier present
; PR0309  1                (file,),                        ! /FILE qualifier present
; PR0310  1                (mdl,),                         ! /MDL qualifier present
; PR0311  1                (sdl,),                         ! /SDL qualifier present
; PR0312  1                (input,),                       ! Input parameter present
; PR0313  1                (symbols,),                     ! /SYMBOLS qualifier present
;  R0314  1                (text,));                       ! /TEXT qualifier present
```

```
;  R0315  1
;  R0316  1     !
;  R0317  1     !       Define message codes.
;  R0318  1     !
;  R0319  1
; PR0320  1     $shr_messages(msg,151,
; PR0321  1             (syntax,severe),        ! Syntax error
; PR0322  1             (openin,error),         ! Unable to access input file
; PR0323  1             (openout,severe),       ! Unable to access output file
; PR0324  1             (readerr,severe),       ! Error reading input file
; PR0325  1             (closedel,error),       ! Unable to close input file
; PR0326  1             (parsefail,error),      ! Error parsing file specification
; PR0327  1             (searchfail,error),     ! Error searching for file
; PR0328  1             (writeerr,error),       ! Error writing file
;  R0329  1             );
;  R0330  1
;  R0331  1     MACRO
; MR0332  1         emsg(ident) =
; MR0333  1             BEGIN
; MR0334  1             %IF NOT %DECLARED(%NAME('msg$_',ident))
; MR0335  1                     %THEN EXTERNAL LITERAL %NAME('msg$_',ident); %FI
; MR0336  1             %NAME('msg$_',ident)
;  R0337  1             END%;
;  R0338  1
;  R0339  1     %SBTTL ''
```

```
;  104          0340  1 %SBTTL 'Table of Contents, Variables, Externals, etc.'
;  105          0341  1 !
;  106          0342  1 ! Table of contents
;  107          0343  1 !
;  108          0344  1
;  109          0345  1 FORWARD ROUTINE
;  110          0346  1     start,                          ! Main routine
;  111          0347  1     get_quals:          NOVALUE,    ! Get command qualifiers
;  112          0348  1     rms_error,                      ! RMS general error routine
;  113          0349  1     search_error,                   ! Signal error searching for file
;  114          0350  1     do_file,                        ! Process each input file
;  115          0351  1     get_module_name,                ! Compute module name string
;  116          0352  1     handler;                        ! Signal handler
;  117          0353  1
;  118          0354  1 !
;  119          0355  1 ! Macro SD_ for string descriptors and the strings
;  120          0356  1 !
;  121          0357  1 MACRO
;  122          0358  1     sd [string] = BIND %NAME('sd_', string) =$descriptor (string)%;
;  123          0359  1
;  124          0360  1 sd ('$LINE', 'FILE_NAME', 'LIST', 'MDL', 'OBJECT', 'P1', 'SDL', 'SYMBOLS', 'TEXT');
;  125          0361  1
;  126          0362  1 !
;  127          0363  1 ! OWN storage
;  128          0364  1 !
;  129          0365  1
;  130          0366  1 LITERAL
;  131          0367  1     buffer_size = 512;              ! Length of record buffer
;  132          0368  1
;  133          0369  1 OWN
;  134          0370  1     input_exname:                   ! Input expanded file name buffer
;  135          0371  1         VECTOR [nam$c_maxrss,BYTE],
;  136          0372  1     input_rsname:                   ! Input result file name buffer
;  137          0373  1         VECTOR [nam$c_maxrss,BYTE],
;  138        P 0374  1     input_nam:  $NAM(               ! File name block
;  139        P 0375  1                 RSA = input_rsname,   ! Result string buffer
;  140        P 0376  1                 RSS = nam$c_maxrss,
;  141        P 0377  1                 ESA = input_exname,   ! Expanded string buffer
;  142          0378  1                 ESS = nam$c_maxrss),
;  143          0379  1     input_xabdat:
;  144          0380  1                 $XABDAT(),          ! Date/time XAB for input file
;  145          0381  1
;  146          0382  1     listing_exname:
;  147          0383  1         VECTOR [nam$c_maxrss,BYTE],   ! Expanded file name buffer
;  148          0384  1     listing_relname:                ! Related name buffer.
;  149          0385  1         VECTOR [nam$c_maxrss, BYTE],
;  150          0386  1     listing_rel_nam:                ! Name block for related input file.
;  151        P 0387  1                 $NAM(
;  152          0388  1                 RSA = listing_relname),
;  153          0389  1     listing_rsname:
;  154          0390  1         VECTOR [nam$c_maxrss,BYTE],   ! Result file name buffer
;  155        P 0391  1     listing_nam: $NAM(
;  156        P 0392  1                 ESA = listing_exname,   ! Expanded string buffer
;  157        P 0393  1                 ESS = nam$c_maxrss,
;  158        P 0394  1                 RSA = listing_rsname,   ! Result string buffer
;  159        P 0395  1                 RSS = nam$c_maxrss,
;  160          0396  1                 RLF = listing_rel_nam), ! Related NAM block
```

```
;  161          0397  1
;  162          0398  1     mdl_exname:
;  163          0399  1         VECTOR [nam$c_maxrss,BYTE],     ! Expanded file name buffer
;  164          0400  1     mdl_relname:                        ! Related name buffer.
;  165          0401  1         VECTOR [nam$c_maxrss, BYTE],
;  166          0402  1     mdl_rel_nam:                        ! Name block for related input file.
;  167        P 0403  1                 $NAM(
;  168          0404  1                 RSA = mdl_relname),
;  169          0405  1     mdl_rsname:
;  170          0406  1         VECTOR [nam$c_maxrss,BYTE],     ! Result flie name buffer
;  171        P 0407  1     mdl_nam: $NAM(
;  172        P 0408  1                 ESA = mdl_exname,       ! Expanded string buffer
;  173        P 0409  1                 ESS = nam$c_maxrss,
;  174        P 0410  1                 RSA = mdl_rsname,       ! Result string buffer
;  175        P 0411  1                 RSS = nam$c_maxrss,
;  176          0412  1                 RLF = mdl_rel_nam),     ! Related NAM block
;  177          0413  1
;  178          0414  1     object_exname:
;  179          0415  1         VECTOR [nam$c_maxrss,BYTE],     ! Expanded file name buffer
;  180          0416  1     object_relname:                     ! Related name buffer.
;  181          0417  1         VECTOR [nam$c_maxrss, BYTE],
;  182          0418  1     object_rel_nam:                     ! Name block for related input file.
;  183        P 0419  1                 $NAM(
;  184          0420  1                 RSA = object_relname),
;  185          0421  1     object_rsname:
;  186          0422  1         VECTOR [nam$c_maxrss,BYTE],     ! Result file name buffer
;  187        P 0423  1     object_nam: $NAM(
;  188        P 0424  1                 ESA = object_exname,    ! Expanded string buffer
;  189        P 0425  1                 ESS = nam$c_maxrss,
;  190        P 0426  1                 RSA = object_rsname,    ! Result string buffer
;  191        P 0427  1                 RSS = nam$c_maxrss,
;  192          0428  1                 RLF = object_rel_nam),  ! Related NAM block
;  193          0429  1
;  194          0430  1     record_buffer:
;  195          0431  1         VECTOR [buffer_size,BYTE],      ! Input record buffer
;  196          0432  1
;  197          0433  1     sdl_exname:
;  198          0434  1         VECTOR [nam$c_maxrss,BYTE],     ! Expanded file name buffer
;  199          0435  1     sdl_relname:                        ! Related name buffer.
;  200          0436  1         VECTOR [nam$c_maxrss, BYTE],
;  201          0437  1     sdl_rel_nam:                        ! Name block for related input file.
;  202        P 0438  1                 $NAM(
;  203          0439  1                 RSA = sdl_relname),
;  204          0440  1     sdl_rsname:
;  205          0441  1         VECTOR [nam$c_maxrss,BYTE],     ! Result file name buffer
;  206        P 0442  1     sdl_nam: $NAM(
;  207        P 0443  1                 ESA = sdl_exname,       ! Expanded string buffer
;  208        P 0444  1                 ESS = nam$c_maxrss,
;  209        P 0445  1                 RSA = sdl_rsname,       ! Result string buffer
;  210        P 0446  1                 RSS = nam$c_maxrss,
;  211          0447  1                 RLF = sdl_rel_nam);     ! Related NAM block
;  212          0448  1
;  213          0449  1 GLOBAL
;  214          0450  1     cli_flags           : BITVECTOR[32] ! Qualifier presence bitmap
;  215          0451  1         INITIAL(0),                     ! Initially none present
;  216          0452  1
;  217          0453  1     command_line_desc:                  ! Descriptor for entire command line.
```

```
;  218          0454  1          BLOCK [dsc$k_s_bln, BYTE]
;  219          0455  1          PRESET ([dsc$w_length] = 0,
;  220          0456  1                  [dsc$b_class] = dsc$k_class_d,
;  221          0457  1                  [dsc$b_dtype] = dsc$k_dtype_t,
;  222          0458  1                  [dsc$a_pointer] = 0),
;  223          0459  1
;  224          0460  1      file_queue_hdr:                     ! Header for queue of input file names.
;  225          0461  1          BBLOCK[8] FIELD (info_queue_flds)
;  226          0462  1          PRESET ([queue_flink] = file_queue_hdr, [queue_blink] = file_queue_hdr),
;  227          0463  1
;  228          0464  1      filename_desc:                      ! Descriptor of /FILE value
;  229          0465  1          BLOCK [dsc$k_s_bln, BYTE]
;  230          0466  1          PRESET ([dsc$w_length] = 0,
;  231          0467  1                  [dsc$b_class] = dsc$k_class_d,
;  232          0468  1                  [dsc$b_dtype] = dsc$k_dtype_t,
;  233          0469  1                  [dsc$a_pointer] = 0),
;  234          0470  1
;  235        P 0471  1      input_fab:  $FAB(                   ! Input file FAB
;  236        P 0472  1                  DNA = UPLIT('.MSG'),    ! Default file name
;  237        P 0473  1                  DNS = 4,
;  238        P 0474  1                  NAM = input_nam,        ! NAM block
;  239          0475  1                  XAB = input_xabdat),    ! XABDAT block
;  240        P 0476  1      input_rab:  $RAB(                   ! Input file RAB
;  241        P 0477  1                  FAB = input_fab,
;  242        P 0478  1                  UBF = record_buffer,    ! Record buffer
;  243          0479  1                  USZ = buffer_size),
;  244          0480  1
;  245        P 0481  1      listing_fab: $FAB(                  ! Listing file FAB
;  246        P 0482  1                  DNA = UPLIT('.LIS'),    ! Default file name
;  247        P 0483  1                  DNS = 4,
;  248        P 0484  1                  FOP = OFP,
;  249        P 0485  1                  NAM = listing_nam,      ! Address of NAM block
;  250        P 0486  1                  RAT = CR,               ! Carriage return format
;  251          0487  1                  RFM = VAR),             ! Variable length records
;  252        P 0488  1      listing_rab: $RAB(                  ! Listing file RAB
;  253        P 0489  1                  FAB = listing_fab,
;  254          0490  1                  ROP = WBH),             ! Overlap I/O if possible
;  255          0491  1
;  256        P 0492  1      mdl_fab:    $FAB(                   ! MDL file FAB
;  257        P 0493  1                  DNA = UPLIT('.MDL'),    ! Default file name
;  258        P 0494  1                  DNS = 4,
;  259        P 0495  1                  FOP = OFP,
;  260        P 0496  1                  NAM = mdl_nam,          ! Address of NAM block
;  261        P 0497  1                  RAT = CR,               ! Carriage return format
;  262          0498  1                  RFM = VAR),             ! Variable length records
;  263        P 0499  1      mdl_rab:    $RAB(                   ! MDL file RAB
;  264        P 0500  1                  FAB = mdl_fab,
;  265          0501  1                  ROP = WBH),             ! Overlap I/O if possible
;  266          0502  1
;  267          0503  1      module_name:        VECTOR  [2],    ! Descriptor of module name
;  268          0504  1
;  269        P 0505  1      object_fab: $FAB(                   ! Object module file FAB
;  270        P 0506  1                  DNA = UPLIT('.OBJ'),
;  271        P 0507  1                  DNS = 4,
;  272        P 0508  1                  FOP = OFP,
;  273          0509  1                  NAM = object_nam),
;  274        P 0510  1      object_rab: $RAB(                   ! Object module RAB
```

```
;  275          0511  1                   FAB = object_fab),
;  276          0512  1
;  277        P 0513  1      output_fab: $FAB(                    ! Output file FAB
;  278        P 0514  1                   FNM='SYS$OUTPUT:',      ! File name
;  279        P 0515  1                   RAT = CR,               ! Carriage return format
;  280          0516  1                   RFM = VAR),             ! Variable length records
;  281        P 0517  1      output_rab: $RAB(                    ! Output file RAB
;  282          0518  1                   FAB = output_fab),
;  283          0519  1
;  284        P 0520  1      sdl_fab:    $FAB(                    ! SDL file FAB
;  285        P 0521  1                   DNA = UPLIT('.SDL'),    ! Default file name
;  286        P 0522  1                   DNS = 4,
;  287        P 0523  1                   FOP = OFP,
;  288        P 0524  1                   NAM = sdl_nam,          ! Address of name block
;  289        P 0525  1                   RAT = CR,               ! Carriage return format
;  290          0526  1                   RFM = VAR),             ! Variable length records
;  291        P 0527  1      sdl_rab:    $RAB(                    ! SDL file RAB
;  292        P 0528  1                   FAB = sdl_fab,
;  293          0529  1                   ROP = WBH),             ! Overlap I/O if possible
;  294          0530  1
;  295          0531  1      worst_error: INITIAL(ss$_normal)     ! Worst error encountered
;  296          0532  1                   BBLOCK[4];
;  297          0533  1
;  298          0534  1 !
;  299          0535  1 ! External routines
;  300          0536  1 !
;  301          0537  1
;  302          0538  1 EXTERNAL ROUTINE
;  303          0539  1      parse_file,                          ! Parse input file
;  304          0540  1      output_object,                       ! Output object module
;  305          0541  1      end_listing,                         ! Cleanup listing file
;  306          0542  1      error_summary,                       ! Give error summary
;  307          0543  1      lib$file_scan:      ADDRESSING_MODE(GENERAL),       ! Search wildcard sequence
;  308          0544  1                                                       ! for all files.
;  309          0545  1      lib$find_file:      ADDRESSING_MODE(GENERAL),       ! Search wildcard sequence
;  310          0546  1                                                       ! for next file.
;  311          0547  1      lib$free_vm:        ADDRESSING_MODE(GENERAL),       ! Dynamic memory deallocation.
;  312          0548  1      lib$get_vm:         ADDRESSING_MODE(GENERAL),       ! Allocate dynamic memory.
;  313          0549  1      cli$get_value:      ADDRESSING_MODE(GENERAL),       ! Get command line values.
;  314          0550  1      cli$present:        ADDRESSING_MODE(GENERAL);       ! Test command line for
;  315          0551  1                                                       ! presence of qualifiers.
;  316          0552  1
;  317          0553  1 !
;  318          0554  1 ! External literals (for command line interpreter codes):
;  319          0555  1 !
;  320          0556  1
;  321          0557  1 EXTERNAL LITERAL
;  322          0558  1      cli$_absent,
;  323          0559  1      cli$_defaulted,
;  324          0560  1      cli$_locneg,
;  325          0561  1      cli$_locpres,
;  326          0562  1      cli$_negated,
;  327          0563  1      cli$_present;
;  328          0564  1
;  329          0565  1 ROUTINE null: NOVALUE = ;
```

```
                                                                .TITLE  MAIN MESSAGE FILE COMPILER
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                45  4E  49  4C  24  00000 P.AAB:  .ASCII  \$LINE\                          ;
                                                    00005       .BLKB   3
                                  00000005          00008 P.AAA:  .LONG   5                               ;
                                  00000000'         0000C       .ADDRESS P.AAB                          ;
                        45  4D  41  4E  5F  45  4C  49  46  00010 P.AAD:  .ASCII  \FILE_NAME\             ;
                                                    00019       .BLKB   3
                                  00000009          0001C P.AAC:  .LONG   9                               ;
                                  00000000'         00020       .ADDRESS P.AAD                          ;
                                    54  53  49  4C  00024 P.AAF:  .ASCII  \LIST\                        ;
                                  00000004          00028 P.AAE:  .LONG   4                               ;
                                  00000000'         0002C       .ADDRESS P.AAF                          ;
                                    4C  44  4D      00030 P.AAH:  .ASCII  \MDL\                         ;
                                                    00033       .BLKB   1
                                  00000003          00034 P.AAG:  .LONG   3                               ;
                                  00000000'         00038       .ADDRESS P.AAH                          ;
                            54  43  45  4A  42  4F  0003C P.AAJ:  .ASCII  \OBJECT\                       ;
                                                    00042       .BLKB   2
                                  00000006          00044 P.AAI:  .LONG   6                               ;
                                  00000000'         00048       .ADDRESS P.AAJ                          ;
                                    31  50          0004C P.AAL:  .ASCII  \P1\                          ;
                                                    0004E       .BLKB   2
                                  00000002          00050 P.AAK:  .LONG   2                               ;
                                  00000000'         00054       .ADDRESS P.AAL                          ;
                                    4C  44  53      00058 P.AAN:  .ASCII  \SDL\                         ;
                                                    0005B       .BLKB   1
                                  00000003          0005C P.AAM:  .LONG   3                               ;
                                  00000000'         00060       .ADDRESS P.AAN                          ;
                        53  4C  4F  42  4D  59  53  00064 P.AAP:  .ASCII  \SYMBOLS\                      ;
                                                    0006B       .BLKB   1
                                  00000007          0006C P.AAO:  .LONG   7                               ;
                                  00000000'         00070       .ADDRESS P.AAP                          ;
                                    54  58  45  54  00074 P.AAR:  .ASCII  \TEXT\                        ;
                                  00000004          00078 P.AAQ:  .LONG   4                               ;
                                  00000000'         0007C       .ADDRESS P.AAR                          ;
                                    47  53  4D  2E  00080 P.AAS:  .ASCII  \.MSG\                        ;
                                    53  49  4C  2E  00084 P.AAT:  .ASCII  \.LIS\                        ;
                                    4C  44  4D  2E  00088 P.AAU:  .ASCII  \.MDL\                        ;
                                    4A  42  4F  2E  0008C P.AAV:  .ASCII  \.OBJ\                        ;
3A  54  55  50  54  55  4F  24  53  59  53  00090 P.AAW:  .ASCII  \SYS$OUTPUT:\                         ;
                                                    0009B       .BLKB   1                               ;
                                    4C  44  53  2E  0009C P.AAX:  .ASCII  \.SDL\                        ;

                                                                .PSECT  $OWN$,NOEXE,2

                                                    00000 INPUT_EXNAME:
                                                                .BLKB   255
                                                    000FF       .BLKB   1
                                                    00100 INPUT_RSNAME:
                                                                .BLKB   255
                                                    001FF       .BLKB   1
                                                      02  00200 INPUT_NAM:
                                                                .BYTE   2                               ;
```

```
                60  00201          .BYTE   96
                FF  00202          .BYTE   -1
                00  00203          .BYTE   0
        00000000'   00204          .ADDRESS INPUT_RSNAME
                00  00208          .BYTE   0
                00  00209          .BYTE   0
                FF  0020A          .BYTE   -1
                00  0020B          .BYTE   0
        00000000'   0020C          .ADDRESS INPUT_EXNAME
        00000000    00210          .LONG   0
            0000#   00214          .WORD   0[8]
            0000#   00224          .WORD   0[3]
            0000#   0022A          .WORD   0[3]
        00000000    00230          .LONG   0
        00000000    00234          .LONG   0
                00  00238          .BYTE   0
                00  00239          .BYTE   0
                00  0023A          .BYTE   0
                00  0023B          .BYTE   0
                00  0023C          .BYTE   0
                00  0023D          .BYTE   0
                00# 0023E          .BYTE   0[2]
        00000000    00240          .LONG   0
        00000000    00244          .LONG   0
        00000000    00248          .LONG   0
        00000000    0024C          .LONG   0
        00000000    00250          .LONG   0
        00000000    00254          .LONG   0
        00000000#   00258          .LONG   0[2]
                12  00260 INPUT_XABDAT:
                                   .BYTE   18
                2C  00261          .BYTE   44
              0000  00262          .WORD   0
        00000000    00264          .LONG   0
              0000  00268          .WORD   0
              0000  0026A          .WORD   0
        00000000#   0026C          .LONG   0[2]
        00000000#   00274          .LONG   0[2]
        00000000    0027C          .LONG   0
        00000000    00280          .LONG   0
        00000000#   00284          .LONG   0[2]
                    0028C LISTING_EXNAME:
                                   .BLKB   255
                    0038B          .BLKB   1
                    0038C LISTING_RELNAME:
                                   .BLKB   255
                    0048B          .BLKB   1
                02  0048C LISTING_REL_NAM:
                                   .BYTE   2
                60  0048D          .BYTE   96
                00  0048E          .BYTE   0
                00  0048F          .BYTE   0
        00000000'   00490          .ADDRESS LISTING_RELNAME
                00  00494          .BYTE   0
                00  00495          .BYTE   0
                00  00496          .BYTE   0
                00  00497          .BYTE   0
```

```
                                00000000  00498          .LONG   0                                                       ;
                                00000000  0049C          .LONG   0                                                       ;
                                    0000# 004A0          .WORD   0[8]                                                    ;
                                    0000# 004B0          .WORD   0[3]                                                    ;
                                    0000# 004B6          .WORD   0[3]                                                    ;
                                00000000  004BC          .LONG   0                                                       ;
                                00000000  004C0          .LONG   0                                                       ;
                                      00  004C4          .BYTE   0                                                       ;
                                      00  004C5          .BYTE   0                                                       ;
                                      00  004C6          .BYTE   0                                                       ;
                                      00  004C7          .BYTE   0                                                       ;
                                      00  004C8          .BYTE   0                                                       ;
                                      00  004C9          .BYTE   0                                                       ;
                                      00# 004CA          .BYTE   0[2]                                                    ;
                                00000000  004CC          .LONG   0                                                       ;
                                00000000  004D0          .LONG   0                                                       ;
                                00000000  004D4          .LONG   0                                                       ;
                                00000000  004D8          .LONG   0                                                       ;
                                00000000  004DC          .LONG   0                                                       ;
                                00000000  004E0          .LONG   0                                                       ;
                                00000000# 004E4          .LONG   0[2]                                                    ;
                                          004EC LISTING_RSNAME:
                                                         .BLKB   255
                                          005EB          .BLKB   1
                                      02  005EC LISTING_NAM:
                                                         .BYTE   2                                                       ;
                                      60  005ED          .BYTE   96                                                      ;
                                      FF  005EE          .BYTE   -1                                                      ;
                                      00  005EF          .BYTE   0                                                       ;
                                00000000' 005F0          .ADDRESS LISTING_RSNAME                                         ;
                                      00  005F4          .BYTE   0                                                       ;
                                      00  005F5          .BYTE   0                                                       ;
                                      FF  005F6          .BYTE   -1                                                      ;
                                      00  005F7          .BYTE   0                                                       ;
                                00000000' 005F8          .ADDRESS LISTING_EXNAME                                         ;
                                00000000' 005FC          .ADDRESS LISTING_REL_NAM                                        ;
                                    0000# 00600          .WORD   0[8]                                                    ;
                                    0000# 00610          .WORD   0[3]                                                    ;
                                    0000# 00616          .WORD   0[3]                                                    ;
                                00000000  0061C          .LONG   0                                                       ;
                                00000000  00620          .LONG   0                                                       ;
                                      00  00624          .BYTE   0                                                       ;
                                      00  00625          .BYTE   0                                                       ;
                                      00  00626          .BYTE   0                                                       ;
                                      00  00627          .BYTE   0                                                       ;
                                      00  00628          .BYTE   0                                                       ;
                                      00  00629          .BYTE   0                                                       ;
                                      00# 0062A          .BYTE   0[2]                                                    ;
                                00000000  0062C          .LONG   0                                                       ;
                                00000000  00630          .LONG   0                                                       ;
                                00000000  00634          .LONG   0                                                       ;
                                00000000  00638          .LONG   0                                                       ;
                                00000000  0063C          .LONG   0                                                       ;
                                00000000  00640          .LONG   0                                                       ;
                                00000000# 00644          .LONG   0[2]                                                    ;
                                          0064C MDL_EXNAME:
                                                         .BLKB   255
```

```
                        0074B           .BLKB   1
                        0074C MDL_RELNAME:
                                        .BLKB   255
                        0084B           .BLKB   1
                    02  0084C MDL_REL_NAM:
                                        .BYTE   2
                    60  0084D           .BYTE   96
                    00  0084E           .BYTE   0
                    00  0084F           .BYTE   0
            00000000'   00850           .ADDRESS MDL_RELNAME
                    00  00854           .BYTE   0
                    00  008?            .BYTE   0
                    00  00856           .BYTE   0
                    00  00857           .BYTE   0
            00000000    00858           .LONG   0
            00000000    0085C           .LONG   0
                0000#   00860           .WORD   0[8]
                0000#   00870           .WORD   0[3]
                0000#   00876           .WORD   0[3]
            00000000    0087C           .LONG   0
            00000000    00880           .LONG   0
                    00  00884           .BYTE   0
                    00  00885           .BYTE   0
                    00  00886           .BYTE   0
                    00  00887           .BYTE   0
                    00  00888           .BYTE   0
                    00  00889           .BYTE   0
                    00# 0088A           .BYTE   0[2]
            00000000    0088C           .LONG   0
            00000000    00890           .LONG   0
            00000000    00894           .LONG   0
            00000000    00898           .LONG   0
            00000000    0089C           .LONG   0
            00000000    008A0           .LONG   0
            00000000#   008A4           .LONG   0[2]
                        008AC MDL_RSNAME:
                                        .BLKB   255
                        009AB           .BLKB   1
                    02  009AC MDL_NAM:  .BYTE   2
                    60  009AD           .BYTE   96
                    FF  009AE           .BYTE   -1
                    00  009AF           .BYTE   0
            00000000'   009B0           .ADDRESS MDL_RSNAME
                    00  009B4           .BYTE   0
                    00  009B5           .BYTE   0
                    FF  009B6           .BYTE   -1
                    00  009B7           .BYTE   0
            00000000'   009B8           .ADDRESS MDL_EXNAME
            00000000'   009BC           .ADDRESS MDL_REL_NAM
                0000#   009C0           .WORD   0[8]
                0000#   009D0           .WORD   0[3]
                0000#   009D6           .WORD   0[3]
            00000000    009DC           .LONG   0
            00000000    009E0           .LONG   0
                    00  009E4           .BYTE   0
                    00  009E5           .BYTE   0
                    00  009E6           .BYTE   0
```

```
                                00  009E7          .BYTE   0                                                       ;
                                00  009E8          .BYTE   0                                                       ;
                                00  009E9          .BYTE   0                                                       ;
                                00# 009EA          .BYTE   0[2]                                                    ;
                            00000000  009EC          .LONG   0                                                      ;
                            00000000  009F0          .LONG   0                                                      ;
                            00000000  009F4          .LONG   0                                                      ;
                            00000000  009F8          .LONG   0                                                      ;
                            00000000  009FC          .LONG   0                                                      ;
                            00000000  00A00          .LONG   0                                                      ;
                            00000000# 00A04          .LONG   0[2]                                                   ;
                                      00A0C OBJECT_EXNAME:
                                                     .BLKB   255
                                      00B0B          .BLKB   1
                                      00B0C OBJECT_RELNAME:
                                                     .BLKB   255
                                      00C0B          .BLKB   1
                                02    00C0C OBJECT_RE_NAM:
                                                     .BYTE   2
                                60    00C0D          .BYTE   96                                                      ;
                                00    00C0E          .BYTE   0                                                       ;
                                00    00C0F          .BYTE   0                                                       ;
                            00000000' 00C10          .ADDRESS OBJECT_RELNAME                                         ;
                                00    00C14          .BYTE   0                                                       ;
                                00    00C15          .BYTE   0                                                       ;
                                00    00C16          .BYTE   0                                                       ;
                                00    00C17          .BYTE   0                                                       ;
                            00000000  00C18          .LONG   0                                                      ;
                            00000000  00C1C          .LONG   0                                                      ;
                               0000# 00C20          .WORD   0[8]                                                   ;
                               0000# 00C30          .WORD   0[3]                                                   ;
                               0000# 00C36          .WORD   0[3]                                                   ;
                            00000000  00C3C          .LONG   0                                                      ;
                            00000000  00C40          .LONG   0                                                      ;
                                00    00C44          .BYTE   0                                                       ;
                                00    00C45          .BYTE   0                                                       ;
                                00    00C46          .BYTE   0                                                       ;
                                00    00C47          .BYTE   0                                                       ;
                                00    00C48          .BYTE   0                                                       ;
                                00    00C49          .BYTE   0                                                       ;
                                00# 00C4A          .BYTE   0[2]                                                    ;
                            00000000  00C4C          .LONG   0                                                      ;
                            00000000  00C50          .LONG   0                                                      ;
                            00000000  00C54          .LONG   0                                                      ;
                            00000000  00C58          .LONG   0                                                      ;
                            00000000  00C5C          .LONG   0                                                      ;
                            00000000  00C60          .LONG   0                                                      ;
                            00000000# 00C64          .LONG   0[2]                                                   ;
                                      00C6C OBJECT_RSNAME:
                                                     .BLKB   255
                                      00D6B          .BLKB   1
                                02    00D6C OBJECT_NAM:
                                                     .BYTE   2
                                60    00D6D          .BYTE   96                                                      ;
                                FF    00D6E          .BYTE   -1                                                      ;
                                00    00D6F          .BYTE   0                                                       ;
                            00000000' 00D70          .ADDRESS OBJECT_RSNAME                                          ;
```

```
                      00  00D74          .BYTE   0
                      00  00D75          .BYTE   0
                      FF  00D76          .BYTE   -1
                      00  00D77          .BYTE   0
                00000000' 00D78          .ADDRESS OBJECT_EXNAME
                00000000' 00D7C          .ADDRESS OBJECT_REL_NAM
                    0000# 00D80          .WORD   0[8]
                    0000# 00D90          .WORD   0[3]
                    0000# 00D96          .WORD   0[3]
                00000000  00D9C          .LONG   0
                00000000  00DA0          .LONG   0
                      00  00DA4          .BYTE   0
                      00  00DA5          .BYTE   0
                      00  00DA6          .BYTE   0
                      00  00DA7          .BYTE   0
                      00  00DA8          .BYTE   0
                      00  00DA9          .BYTE   0
                      00# 00DAA          .BYTE   0[2]
                00000000  00DAC          .LONG   0
                00000000  00DB0          .LONG   0
                00000000  00DB4          .LONG   0
                00000000  00DB8          .LONG   0
                00000000  00DBC          .LONG   0
                00000000  00DC0          .LONG   0
                00000000# 00DC4          .LONG   0[2]
                          00DCC RECORD_BUFFER:
                                         .BLKB   512
                          00FCC SDL_EXNAME:
                                         .BLKB   255
                          010CB          .BLKB   1
                          010CC SDL_RELNAME:
                                         .BLKB   255
                          011CB          .BLKB   1
                      02  011CC SDL_REL_NAM:
                                         .BYTE   2
                      60  011CD          .BYTE   96
                      00  011CE          .BYTE   0
                      00  011CF          .BYTE   0
                00000000' 011D0          .ADDRESS SDL_RELNAME
                      00  011D4          .BYTE   0
                      00  011D5          .BYTE   0
                      00  011D6          .BYTE   0
                      00  011D7          .BYTE   0
                00000000  011D8          .LONG   0
                00000000  011DC          .LONG   0
                    0000# 011E0          .WORD   0[8]
                    0000# 011F0          .WORD   0[3]
                    0000# 011F6          .WORD   0[3]
                00000000  011FC          .LONG   0
                00000000  01200          .LONG   0
                      00  01204          .BYTE   0
                      00  01205          .BYTE   0
                      00  01206          .BYTE   0
                      00  01207          .BYTE   0
                      00  01208          .BYTE   0
                      00  01209          .BYTE   0
                      00# 0120A          .BYTE   0[2]
```

```
                00000000  0120C          .LONG   0
                00000000  01210          .LONG   0
                00000000  01214          .LONG   0
                00000000  01218          .LONG   0
                00000000  0121C          .LONG   0
                00000000  01220          .LONG   0
                00000000# 01224          .LONG   0[2]
                          0122C SDL_RSNAME:
                                         .BLKB   255
                          0132B          .BLKB   1
                      02  0132C SDL_NAM: .BYTE   2
                      60  0132D          .BYTE   96
                      FF  0132E          .BYTE   -1
                      00  0132F          .BYTE   0
                00000000' 01330          .ADDRESS SDL_RSNAME
                      00  01334          .BYTE   0
                      00  01335          .BYTE   0
                      FF  01336          .BYTE   -1
                      00  01337          .BYTE   0
                00000000' 01338          .ADDRESS SDL_EXNAME
                00000000' 0133C          .ADDRESS SDL_REL_NAM
                    0000# 01340          .WORD   0[8]
                    0000# 01350          .WORD   0[3]
                    0000# 01356          .WORD   0[3]
                00000000  0135C          .LONG   0
                00000000  01360          .LONG   0
                      00  01364          .BYTE   0
                      00  01365          .BYTE   0
                      00  01366          .BYTE   0
                      00  01367          .BYTE   0
                      00  01368          .BYTE   0
                      00  01369          .BYTE   0
                      00# 0136A          .BYTE   0[2]
                00000000  0136C          .LONG   0
                00000000  01370          .LONG   0
                00000000  01374          .LONG   0
                00000000  01378          .LONG   0
                00000000  0137C          .LONG   0
                00000000  01380          .LONG   0
                00000000# 01384          .LONG   0[2]

                                         .PSECT  $GLOBAL$,NOEXE,2

                00000000  00000 CLI_FLAGS::
                                         .LONG   0                                      ;
                    0000  00004 COMMAND_LINE_DESC::
                                         .WORD   0                                      ;
                  02  0E  00006          .BYTE   14, 2                                  ;
                00000000  00008          .LONG   0                                      ;
00000000'       00000000' 0000C FILE_QUEUE_HDR::
                                         .ADDRESS FILE_QUEUE_HDR, FILE_QUEUE_HDR        ;
                    0000  00014 FILENAME_DESC::
                                         .WORD   0                                      ;
                  02  0E  00016          .BYTE   14, 2                                  ;
                00000000  00018          .LONG   0                                      ;
                      03  0001C INPUT_FAB::
                                         .BYTE   3                                      ;
```

```
                50  0001D          .BYTE   80
              0000  0001E          .WORD   0
          00000000  00020          .LONG   0
          00000000  00024          .LONG   0
          00000000  00028          .LONG   0
          00000000  0002C          .LONG   0
              0000  00030          .WORD   0
                02  00032          .BYTE   2
                00  00033          .BYTE   0
          00000000  00034          .LONG   0
                00  00038          .BYTE   0
                00  00039          .BYTE   0
                00  0003A          .BYTE   0
                02  0003B          .BYTE   2
          00000000  0003C          .LONG   0
          00000000' 00040          .ADDRESS INPUT_XABDAT
          00000000' 00044          .ADDRESS INPUT_NAM
          00000000  00048          .LONG   0
          00000000' 0004C          .ADDRESS P.AAS
                00  00050          .BYTE   0
                04  00051          .BYTE   4
              0000  00052          .WORD   0
          00000000  00054          .LONG   0
              0000  00058          .WORD   0
                00  0005A          .BYTE   0
                00  0005B          .BYTE   0
          00000000  0005C          .LONG   0
          00000000  00060          .LONG   0
              0000  00064          .WORD   0
                00  00066          .BYTE   0
                00  00067          .BYTE   0
          00000000  00068          .LONG   0
                01  0006C INPUT_RAB::
                                   .BYTE   1
                44  0006D          .BYTE   68
              0000  0006E          .WORD   0
          00000000  00070          .LONG   0
          00000000  00074          .LONG   0
          00000000  00078          .LONG   0
              0000# 0007C          .WORD   0[3]
              0000  00082          .WORD   0
          00000000  00084          .LONG   0
              0000  00088          .WORD   0
                00  0008A          .BYTE   0
                00  0008B          .BYTE   0
              0200  0008C          .WORD   512
              0000  0008E          .WORD   0
          00000000' 00090          .ADDRESS RECORD_BUFFER
          00000000  00094          .LONG   0
          00000000  00098          .LONG   0
          00000000  0009C          .LONG   0
                00  000A0          .BYTE   0
                00  000A1          .BYTE   0
                00  000A2          .BYTE   0
                00  000A3          .BYTE   0
          00000000  000A4          .LONG   0
          00000000' 000A8          .ADDRESS INPUT_FAB
```

```
                                00000000  000AC          .LONG   0                                                       ;
                                      03  000B0 LISTING_FAB::                                                          ;
                                                         .BYTE   3                                                       ;
                                      50  000B1          .BYTE   80                                                      ;
                                    0000  000B2          .WORD   0                                                       ;
                                20000000  000B4          .LONG   536870912                                               ;
                                00000000  000B8          .LONG   0                                                       ;
                                00000000  000BC          .LONG   0                                                       ;
                                00000000  000C0          .LONG   0                                                       ;
                                    0000  000C4          .WORD   0                                                       ;
                                      02  000C6          .BYTE   2                                                       ;
                                      00  000C7          .BYTE   0                                                       ;
                                00000000  000C8          .LONG   0                                                       ;
                                      00  000CC          .BYTE   0                                                       ;
                                      00  000CD          .BYTE   0                                                       ;
                                      02  000CE          .BYTE   2                                                       ;
                                      02  000CF          .BYTE   2                                                       ;
                                00000000  000D0          .LONG   0                                                       ;
                                00000000  000D4          .LONG   0                                                       ;
                                00000000' 000D8          .ADDRESS LISTING_NAM                                            ;
                                00000000  000DC          .LONG   0                                                       ;
                                00000000' 000E0          .ADDRESS P.AAT                                                  ;
                                      00  000E4          .BYTE   0                                                       ;
                                      04  000E5          .BYTE   4                                                       ;
                                    0000  000E6          .WORD   0                                                       ;
                                00000000  000E8          .LONG   0                                                       ;
                                    0000  000EC          .WORD   0                                                       ;
                                      00  000EE          .BYTE   0                                                       ;
                                      00  000EF          .BYTE   0                                                       ;
                                00000000  000F0          .LONG   0                                                       ;
                                00000000  000F4          .LONG   0                                                       ;
                                    0000  000F8          .WORD   0                                                       ;
                                      00  000FA          .BYTE   0                                                       ;
                                      00  000FB          .BYTE   0                                                       ;
                                00000000  000FC          .LONG   0                                                       ;
                                      01  00100 LISTING_RAB::                                                          ;
                                                         .BYTE   1                                                       ;
                                      44  00101          .BYTE   68                                                      ;
                                    0000  00102          .WORD   0                                                       ;
                                00000400  00104          .LONG   1024                                                    ;
                                00000000  00108          .LONG   0                                                       ;
                                00000000  0010C          .LONG   0                                                       ;
                                    0000# 00110          .WORD   0[3]                                                    ;
                                    0000  00116          .WORD   0                                                       ;
                                00000000  00118          .LONG   0                                                       ;
                                    0000  0011C          .WORD   0                                                       ;
                                      00  0011E          .BYTE   0                                                       ;
                                      00  0011F          .BYTE   0                                                       ;
                                    0000  00120          .WORD   0                                                       ;
                                    0000  00122          .WORD   0                                                       ;
                                00000000  00124          .LONG   0                                                       ;
                                00000000  00128          .LONG   0                                                       ;
                                00000000  0012C          .LONG   0                                                       ;
                                00000000  00130          .LONG   0                                                       ;
                                      00  00134          .BYTE   0                                                       ;
                                      00  00135          .BYTE   0                                                       ;
                                      00  00136          .BYTE   0                                                       ;
```

```
                                      00  00137          .BYTE   0
                                00000000  00138          .LONG   0
                                00000000' 0013C          .ADDRESS LISTING_FAB
                                00000000  00140          .LONG   0
                                      03  00144 MDL_FAB::
                                                         .BYTE   3
                                      50  00145          .BYTE   80
                                    0000  00146          .WORD   0
                                20000000  00148          .LONG   536870912
                                00000000  0014C          .LONG   0
                                00000000  00150          .LONG   0
                                00000000  00154          .LONG   0
                                    0000  00158          .WORD   0
                                      02  0015A          .BYTE   2
                                      00  0015B          .BYTE   0
                                00000000  0015C          .LONG   0
                                      00  00160          .BYTE   0
                                      00  00161          .BYTE   0
                                      02  00162          .BYTE   2
                                      02  00163          .BYTE   2
                                00000000  00164          .LONG   0
                                00000000  00168          .LONG   0
                                00000000' 0016C          .ADDRESS MDL_NAM
                                00000000  00170          .LONG   0
                                00000000' 00174          .ADDRESS P.AAU
                                      00  00178          .BYTE   0
                                      04  00179          .BYTE   4
                                    0000  0017A          .WORD   0
                                00000000  0017C          .LONG   0
                                    0000  00180          .WORD   0
                                      00  00182          .BYTE   0
                                      00  00183          .BYTE   0
                                00000000  00184          .LONG   0
                                00000000  00188          .LONG   0
                                    0000  0018C          .WORD   0
                                      00  0018E          .BYTE   0
                                      00  0018F          .BYTE   0
                                00000000  00190          .LONG   0
                                      01  00194 MDL_RAB::
                                                         .BYTE   1
                                      44  00195          .BYTE   68
                                    0000  00196          .WORD   0
                                00000400  00198          .LONG   1024
                                00000000  0019C          .LONG   0
                                00000000  001A0          .LONG   0
                                    0000# 001A4          .WORD   0[3]
                                    0000  001AA          .WORD   0
                                00000000  001AC          .LONG   0
                                    0000  001B0          .WORD   0
                                      00  001B2          .BYTE   0
                                      00  001B3          .BYTE   0
                                    0000  001B4          .WORD   0
                                    0000  001B6          .WORD   0
                                00000000  001B8          .LONG   0
                                00000000  001BC          .LONG   0
                                00000000  001C0          .LONG   0
                                00000000  001C4          .LONG   0
```

```
                    00  001C8          .BYTE   0                                                       ;
                    00  001C9          .BYTE   0                                                       ;
                    00  001CA          .BYTE   0                                                       ;
                    00  001CB          .BYTE   0                                                       ;
              00000000  001CC          .LONG   0                                                       ;
              00000000' 001D0          .ADDRESS MDL_FAB                                                ;
              00000000  001D4          .LONG   0                                                       ;
                        001D8 MODULE_NAME::
                                       .BLKB   8
                    03  001E0 OBJECT_FAB::
                                       .BYTE   3
                    50  001E1          .BYTE   80                                                      ;
                  0000  001E2          .WORD   0                                                       ;
              20000000  001E4          .LONG   536870912                                               ;
              00000000  001E8          .LONG   0                                                       ;
              00000000  001EC          .LONG   0                                                       ;
              00000000  001F0          .LONG   0                                                       ;
                  0000  001F4          .WORD   0                                                       ;
                    02  001F6          .BYTE   2                                                       ;
                    00  001F7          .BYTE   0                                                       ;
              00000000  001F8          .LONG   0                                                       ;
                    00  001FC          .BYTE   0                                                       ;
                    00  001FD          .BYTE   0                                                       ;
                    00  001FE          .BYTE   0                                                       ;
                    02  001FF          .BYTE   2                                                       ;
              00000000  00200          .LONG   0                                                       ;
              00000000  00204          .LONG   0                                                       ;
              00000000' 00208          .ADDRESS OBJECT_NAM                                             ;
              00000000  0020C          .LONG   0                                                       ;
              00000000' 00210          .ADDRESS P.AAV                                                  ;
                    00  00214          .BYTE   0                                                       ;
                    04  00215          .BYTE   4                                                       ;
                  0000  00216          .WORD   0                                                       ;
              00000000  00218          .LONG   0                                                       ;
                  0000  0021C          .WORD   0                                                       ;
                    00  0021E          .BYTE   0                                                       ;
                    00  0021F          .BYTE   0                                                       ;
              00000000  00220          .LONG   0                                                       ;
              00000000  00224          .LONG   0                                                       ;
                  0000  00228          .WORD   0                                                       ;
                    00  0022A          .BYTE   0                                                       ;
                    00  0022B          .BYTE   0                                                       ;
              00000000  0022C          .LONG   0                                                       ;
                    01  00230 OBJECT_RAB::
                                       .BYTE   1
                    44  00231          .BYTE   68                                                      ;
                  0000  00232          .WORD   0                                                       ;
              00000000  00234          .LONG   0                                                       ;
              00000000  00238          .LONG   0                                                       ;
              00000000  0023C          .LONG   0                                                       ;
                  0000# 00240          .WORD   0[3]                                                    ;
                  0000  00246          .WORD   0                                                       ;
              00000000  00248          .LONG   0                                                       ;
                  0000  0024C          .WORD   0                                                       ;
                    00  0024E          .BYTE   0                                                       ;
                    00  0024F          .BYTE   0                                                       ;
                  0000  00250          .WORD   0                                                       ;
```

```
    0000  00252             .WORD    0
00000000  00254             .LONG    0
00000000  00258             .LONG    0
00000000  0025C             .LONG    0
00000000  00260             .LONG    0
      00  00264             .BYTE    0
      00  00265             .BYTE    0
      00  00266             .BYTE    0
      00  00267             .BYTE    0
00000000  00268             .LONG    0
00000000' 0026C             .ADDRESS OBJECT_FAB
00000000  00270             .LONG    0
      03  00274 OUTPUT_FAB::
                            .BYTE    3
      50  00275             .BYTE    80
    0000  00276             .WORD    0
00000000  00278             .LONG    0
00000000  0027C             .LONG    0
00000000  00280             .LONG    0
00000000  00284             .LONG    0
    0000  00288             .WORD    0
      02  0028A             .BYTE    2
      00  0028B             .BYTE    0
00000000  0028C             .LONG    0
      00  00290             .BYTE    0
      00  00291             .BYTE    0
      02  00292             .BYTE    2
      02  00293             .BYTE    2
00000000  00294             .LONG    0
00000000  00298             .LONG    0
00000000  0029C             .LONG    0
00000000' 002A0             .ADDRESS P.AAW
00000000  002A4             .LONG    0
      0B  002A8             .BYTE    11
      00  002A9             .BYTE    0
    0000  002AA             .WORD    0
00000000  002AC             .LONG    0
    0000  002B0             .WORD    0
      00  002B2             .BYTE    0
      00  002B3             .BYTE    0
00000000  002B4             .LONG    0
00000000  002B8             .LONG    0
    0000  002BC             .WORD    0
      00  002BE             .BYTE    0
      00  002BF             .BYTE    0
00000000  002C0             .LONG    0
      01  002C4 OUTPUT_RAB::
                            .BYTE    1
      44  002C5             .BYTE    68
    0000  002C6             .WORD    0
00000000  002C8             .LONG    0
00000000  002CC             .LONG    0
00000000  002D0             .LONG    0
   0000#  002D4             .WORD    0[3]
    0000  002DA             .WORD    0
00000000  002DC             .LONG    0
    0000  002E0             .WORD    0
```

```
                                      00  002E2          .BYTE   0
                                      00  002E3          .BYTE   0
                                    0000  002E4          .WORD   0
                                    0000  002E6          .WORD   0
                                00000000  002E8          .LONG   0
                                00000000  002EC          .LONG   0
                                00000000  002F0          .LONG   0
                                00000000  002F4          .LONG   0
                                      00  002F8          .BYTE   0
                                      00  002F9          .BYTE   0
                                      00  002FA          .BYTE   0
                                      00  002FB          .BYTE   0
                                00000000  002FC          .LONG   0
                                00000000' 00300          .ADDRESS OUTPUT_FAB
                                00000000  00304          .LONG   0
                                      03  00308 SDL_FAB::
                                                         .BYTE   3
                                      50  00309          .BYTE   80
                                    0000  0030A          .WORD   0
                                20000000  0030C          .LONG   536870912
                                00000000  00310          .LONG   0
                                00000000  00314          .LONG   0
                                00000000  00318          .LONG   0
                                    0000  0031C          .WORD   0
                                      02  0031E          .BYTE   2
                                      00  0031F          .BYTE   0
                                00000000  00320          .LONG   0
                                      00  00324          .BYTE   0
                                      00  00325          .BYTE   0
                                      02  00326          .BYTE   2
                                      02  00327          .BYTE   2
                                00000000  00328          .LONG   0
                                00000000  0032C          .LONG   0
                                00000000' 00330          .ADDRESS SDL_NAM
                                00000000  00334          .LONG   0
                                00000000' 00338          .ADDRESS P.AAX
                                      00  0033C          .BYTE   0
                                      04  0033D          .BYTE   4
                                    0000  0033E          .WORD   0
                                00000000  00340          .LONG   0
                                    0000  00344          .WORD   0
                                      00  00346          .BYTE   0
                                      00  00347          .BYTE   0
                                00000000  00348          .LONG   0
                                00000000  0034C          .LONG   0
                                    0000  00350          .WORD   0
                                      00  00352          .BYTE   0
                                      00  00353          .BYTE   0
                                00000000  00354          .LONG   0
                                      01  00358 SDL_RAB::
                                                         .BYTE   1
                                      44  00359          .BYTE   68
                                    0000  0035A          .WORD   0
                                00000400  0035C          .LONG   1024
                                00000000  00360          .LONG   0
                                00000000  00364          .LONG   0
                                    0000# 00368          .WORD   0[3]
```

```
                                0000  0036E           .WORD   0                                                       ;
                          00000000  00370           .LONG   0                                                       ;
                                0000  00374           .WORD   0                                                       ;
                                  00  00376           .BYTE   0                                                       ;
                                  00  00377           .BYTE   0                                                       ;
                                0000  00378           .WORD   0                                                       ;
                                0000  0037A           .WORD   0                                                       ;
                          00000000  0037C           .LONG   0                                                       ;
                          00000000  00380           .LONG   0                                                       ;
                          00000000  00384           .LONG   0                                                       ;
                          00000000  00388           .LONG   0                                                       ;
                                  00  0038C           .BYTE   0                                                       ;
                                  00  0038D           .BYTE   0                                                       ;
                                  00  0038E           .BYTE   0                                                       ;
                                  00  0038F           .BYTE   0                                                       ;
                          00000000  00390           .LONG   0                                                       ;
                          00000000' 00394           .ADDRESS SDL_FAB                                                ;
                          00000000  00398           .LONG   0                                                       ;
                          00000001  0039C WORST_ERROR::
                                                    .LONG   1                                                       ;

                                                SD_$LINE=           P.AAA
                                                SD_FILE_NAME=       P.AAC
                                                SD_LIST=            P.AAE
                                                SD_MDL=             P.AAG
                                                SD_OBJECT=          P.AAI
                                                SD_P1=              P.AAK
                                                SD_SDL=             P.AAM
                                                SD_SYMBOLS=         P.AAO
                                                SD_TEXT=            P.AAQ
                                                        .EXTRN  PARSE_FILE, OUTPUT_OBJECT
                                                        .EXTRN  END_LISTING, ERROR_SUMMARY
                                                        .EXTRN  LIB$FILE_SCAN, LIB$FIND_FILE
                                                        .EXTRN  LIB$FREE_VM, LIB$GET_VM
                                                        .EXTRN  CLI$GET_VALUE, CLI$PRESENT
                                                        .EXTRN  CLI$_ABSENT, CLI$_DEFAULTED
                                                        .EXTRN  CLI$_LOCNEG, CLI$_LOCPRES
                                                        .EXTRN  CLI$_NEGATED, CLI$_PRESENT

                                                        .PSECT  $CODE$,NOWRT,2

                                0000 00000 NULL:    .WORD   Save nothing                                            ; 0565
                                  04 00002          RET                                                             ;

; Routine Size:  3 bytes,    Routine Base:  $CODE$ + 0000
```

```
;   331         0566  1 %SBTTL 'START Routine'
;   332         0567  1 ROUTINE start =
;   333         0568  1
;   334         0569  1 !---
;   335         0570  1 !
;   336         0571  1 !       This is the main routine for the message compiler.  This
;   337         0572  1 !       program accepts the message definition language and generates
;   338         0573  1 !       object modules which may be linked into any image.  $GETMSG
;   339         0574  1 !       will locate the message definitions in the message sections.
;   340         0575  1 !
;   341         0576  1 ! Inputs:
;   342         0577  1 !
;   343         0578  1 !       None
;   344         0579  1 !
;   345         0580  1 ! Outputs:
;   346         0581  1 !
;   347         0582  1 !       Object module specified by /OBJECT
;   348         0583  1 !       Listing file specified by /LIST
;   349         0584  1 !       MDL file specified by /MDL
;   350         0585  1 !       SDL file specified by /SDL
;   351         0586  1 !
;   352         0587  1 !---
;   353         0588  1
;   354         0589  2 BEGIN
;   355         0590  2
;   356         0591  2 BUILTIN
;   357         0592  2     FP,                                 ! Reference FP register
;   358         0593  2     REMQUE;
;   359         0594  2
;   360         0595  2
;   361         0596  2 LOCAL
;   362         0597  2     status;
;   363         0598  2
;   364         0599  2 !
;   365         0600  2 !       Set signal handler
;   366         0601  2 !
;   367         0602  2
;   368         0603  2 .fp = handler;                          ! Set handler address
;   369         0604  2
;   370         0605  2
;   371         0606  2
;   372         0607  2 !
;   373         0608  2 !       Parse the command line
;   374         0609  2 !
;   375         0610  2
;   376         0611  2 get_quals();                            ! Get command qualifiers and parameters.
;   377         0612  2
;   378         0613  2 !
;   379         0614  2 !       Open the output file for error reporting only
;   380         0615  2 !
;   381         0616  2
;   382         0617  2 status = $CREATE (FAB = output_fab);    ! Always create output file
;   383         0618  2 IF NOT .status THEN rms_error (emsg(openout),output_fab);
;   384         0619  2 status = $CONNECT (RAB = output_rab);
;   385         0620  2 IF NOT .status THEN rms_error (emsg(openout) output_fab,output_rab);
;   386         0621  2
;   387         0622  2 !
```

```
;  388     0623  2 !       Process each file in the input list
;  389     0624  2 !
;  390     0625  2
;  391     0626  3 BEGIN
;  392     0627  3 LOCAL
;  393     0628  3     sticky_context:     INITIAL (0),    ! Argument for lib$file_scan.
;  394     0629  3     file_info: REF BBLOCK               ! Address for input file info
;  395     0630  3         FIELD (info_queue_flds);        !  block.
;  396     0631  3
;  397     0632  3
;  398     0633  3 UNTIL REMQUE(.file_queue_hdr[queue_flink], file_info) DO
;  399     0634  4     BEGIN
;  400     0635  4     input_fab[fab$l_fna] = .file_info[name_addr];
;  401     0636  4     input_fab[fab$b_fns] = .file_info[name_length];
;  402     0637  4     lib$file_scan(input_fab,            ! Search all wildcard files
;  403     0638  4                   do_file,              !  calling here if file is found
;  404     0639  4                   search_error,         !  calling here if error occurs
;  405     0640  4                   sticky_context);      !  maintain sticky-context.
;  406     0641  4     lib$free_vm(%REF(queue_element_size), file_info);
;  407     0642  3     END;
;  408     0643  2 END;
;  409     0644  2
;  410     0645  2 !
;  411     0646  2 !       Generate the object module file
;  412     0647  2 !
;  413     0648  2
;  414     0649  2 IF .cli_flags [qual_object]             ! If /OBJECT specified,
;  415     0650  2     AND .worst_error                    ! and no errors so far,
;  416     0651  2 THEN
;  417     0652  2     output_object();                    ! Output object module
;  418     0653  2
;  419     0654  2 !
;  420     0655  2 !       Close the output and listing files
;  421     0656  2 !
;  422     0657  2
;  423     0658  2 IF NOT .worst_error                     ! If error detcted,
;  424     0659  2 THEN
;  425     0660  2     error_summary(output_rab);          ! Give summary
;  426     0661  2
;  427     0662  2 status = $CLOSE (FAB = output_fab);     ! Close output file
;  428     0663  2 IF NOT .status THEN rms_error (emsg(closedel), output_fab);
;  429     0664  2
;  430     0665  2 IF .cli_flags [qual_listing]            ! If /LIST specified,
;  431     0666  2     AND .listing_fab [fab$w_ifi] NEQ 0  ! and it was opened
;  432     0667  2 THEN
;  433     0668  3     BEGIN
;  434     0669  3     end_listing(listing_rab);           ! Cleanup listing file
;  435     0670  3     status = $CLOSE (FAB = listing_fab);        ! Close the listing file
;  436     0671  3     IF NOT .status
;  437     0672  3     THEN
;  438     0673  3         rms_error (emsg(closedel), listing_fab);
;  439     0674  2     END;
;  440     0675  2
;  441     0676  2 IF .cli_flags [qual_mdl]                ! If /MDL specified,
;  442     0677  2    AND .mdl_fab [fab$w_ifi] NEQ 0       ! and it was opened
;  443     0678  2 THEN
;  444     0679  3     BEGIN
```

```
;  445     0680  3      status = $CLOSE (FAB = mdl_fab);     . Close the MDL file
;  446     0681  3      IF NOT .status
;  447     0682  3      THEN
;  448     0683  3          rms_error (emsg(closedel), mdl_fab);
;  449     0684  2      END;
;  450     0685  2
;  451     0686  2 IF .cli_flags [qual_sdl]                  ! If /SDL specified,
;  452     0687  2    AND .sdl_fab [fab$w_ifi] NEQ 0         ! and it was opened
;  453     0688  2 THEN
;  454     0689  3      BEGIN
;  455     0690  3      status = $CLOSE (FAB = sdl_fab);     ! Close the SDL file
;  456     0691  3      IF NOT .status
;  457     0692  3      THEN
;  458     0693  3          rms_error (emsg(closedel), sdl_fab);
;  459     0694  2      END;
;  460     0695  2
;  461     0696  2 RETURN .worst_error;                      ! Return worst error encountered
;  462     0697  2
;  463     0698  1 END;


                                                .EXTRN  SYS$CREATE, SYS$CONNECT
                                                .EXTRN  SYS$CLOSE

                                 003C 00000 START:  .WORD   Save R2,R3,R4,R5                          ; 0567
                     55      0000V   CF  9E 00002       MOVAB   RMS_ERROR, R5                        ;
                     54 00000000G    00  9E 00007       MOVAB   SYS$CLOSE, R4                        ;
                     53      0000'   CF  9E 0000E       MOVAB   OUTPUT_FAB, R3                       ;
                     5E              0C  C2 00013       SUBL2   #12, SP                              ;
                     6D      0000V   CF  9E 00016       MOVAB   HANDLER, (FP)                        ; 0603
                0000V CF                 00  FB 0001B       CALLS   #0, GET_QUALS                    ; 0611
                                         53  DD 00020       PUSHL   R3                               ; 0617
           00000000G 00                  01  FB 00022       CALLS   #1, SYS$CREATE                   ;
                     52                  50  D0 00029       MOVL    R0, STATUS                       ;
                     0B                  52  E8 0002C       BLBS    STATUS, 1$                       ; 0618
                                         53  DD 0002F       PUSHL   R3                               ;
                         009710A4        8F  DD 00031       PUSHL   #9900196                         ;
                     65                  02  FB 00037       CALLS   #2, RMS_ERROR                    ;
                             50          A3  9F 0003A 1$:   PUSHAB  OUTPUT_RAB                       ; 0619
           00000000G 00                  01  FB 0003D       CALLS   #1, SYS$CONNECT                  ;
                     52                  50  D0 00044       MOVL    R0, STATUS                       ;
                     0E                  52  E8 00047       BLBS    STATUS, 2$                       ; 0620
                             50          A3  9F 0004A       PUSHAB  OUTPUT_RAB                       ;
                                         53  DD 0004D       PUSHL   R3                               ;
                         009710A4        8F  DD 0004F       PUSHL   #9900196                         ;
                     65                  03  FB 00055       CALLS   #3, RMS_ERROR                    ;
                              04         AE  D4 00058 2$:   CLRL    STICKY_CONTEXT                   ; 0626
                08   AE       FD98      D3  0F 0005B 3$:   REMQUE  @FILE_QUEUE_HDR, FILE_INFO       ; 0633
                                         39  1D 00061       BVS     4$                               ;
                     50          08      AE  D0 00063       MOVL    FILE_INFO, R0                    ; 0635
                FDD4 C3          0C      A0  D0 00067       MOVL    12(R0), INPUT_FAB+44             ;
                FDDC C3          08      A0  90 0006D       MOVB    8(R0), INPUT_FAB+52              ; 0636
                              04         AE  9F 00073       PUSHAB  STICKY_CONTEXT                   ; 0637
                         0000V   CF  9F 00076       PUSHAB  SEARCH_ERROR                             ;
                         0000V   CF  9F 0007A       PUSHAB  DO_FILE                                  ;
                         FDA8    C3  9F 0007E       PUSHAB  INPUT_FAB                                ;
```

```
                00000000G  00          04  FB 00082          CALLS   #4, LIB$FILE_SCAN                        ;
                                  08    AE  9F 00089          PUSHAB  FILE_INFO                               ; 0641
                             04   AE    10  D0 0008C          MOVL    #16, 4(SP)                              ;
                                  04    AF  9F 00090          PUSHAB  4(SP)                                   ;
                00000000G  00          02  FB 00093          CALLS   #2, LIB$FREE_VM                         ;
                                       BF  11 0009A          BRB     3$                                      ; 0633
            0A      FD8C   C3          01  E1 0009C  4$:     BBC     #1, CLI_FLAGS, 5$                       ; 0649
                       0A      0128    C3  E9 000A2          BLBC    WORST_ERROR, 6$                         ; 0650
                    0000G  CF          00  FB 000A7          CALLS   #0, OUTPUT_OBJECT                       ; 0652
                           08      0128 C3  E8 000AC  5$:     BLBS    WORST_ERROR, 7$                         ; 0658
                                     50    A3  9F 000B1  6$:     PUSHAB  OUTPUT_RAB                          ; 0660
                    0000G  CF          01  FB 000B4          CALLS   #1, ERROR_SUMMARY                       ;
                                       53  DD 000B9  7$:     PUSHL   R3                                      ; 0662
                           64          01  FB 000BB          CALLS   #1, SYS$CLOSE                           ;
                           52          50  D0 000BE          MOVL    R0, STATUS                              ;
                           0B          52  E8 000C1          BLBS    STATUS, 8$                              ; 0663
                                       53  DD 000C4          PUSHL   R3                                      ;
                              0097121A 8F  DD 000C6          PUSHL   #9900570                                ;
                           65          02  FB 000CC          CALLS   #2, RMS_ERROR                           ;
                           29      FD8C C3  E9 000CF  8$:     BLBC    CLI_FLAGS, 9$                           ; 0665
                               FE3E    C3  B5 000D4          TSTW    LISTING_FAB+2                           ; 0666
                                       23  13 000D8          BEQL    9$                                      ;
                               FE8C    C3  9F 000DA          PUSHAB  LISTING_RAB                             ; 0669
                    0000G  CF          01  FB 000DE          CALLS   #1, END_LISTING                         ;
                               FE3C    C3  9F 000E3          PUSHAB  LISTING_FAB                             ; 0670
                           64          01  FB 000E7          CALLS   #1, SYS$CLOSE                           ;
                           52          50  D0 000EA          MOVL    R0, STATUS                              ;
                           0D          52  E8 000ED          BLBS    STATUS, 9$                              ; 0671
                               FE3C    C3  9F 000F0          PUSHAB  LISTING_FAB                             ; 0673
                              0097121A 8F  DD 000F4          PUSHL   #9900570                                ;
                           65          02  FB 000FA          CALLS   #2, RMS_ERROR                           ;
                        20     FD8C   C3          03  E1 000FD  9$:     BBC     #3, CLI_FLAGS, 10$          ; 0676
                                   FED2    C3  B5 00103          TSTW    MDL_FAB+2                           ; 0677
                                           1A  13 00107          BEQL    10$                                 ;
                                   FED0    C3  9F 00109          PUSHAB  MDL_FAB                             ; 0680
                               64          01  FB 0010D          CALLS   #1, SYS$CLOSE                       ;
                               52          50  D0 00110          MOVL    R0, STATUS                          ;
                               0D          52  E8 00113          BLBS    STATUS, 10$                         ; 0681
                                   FED0    C3  9F 00116          PUSHAB  MDL_FAB                             ; 0683
                              0097121A 8F  DD 0011A          PUSHL   #9900570                                ;
                           65          02  FB 00120          CALLS   #2, RMS_ERROR                           ;
                        20     FD8C   C3          04  E1 00123  10$:    BBC     #4, CLI_FLAGS, 11$          ; 0686
                                   0096    C3  B5 00129          TSTW    SDL_FAB+2                           ; 0687
                                           1A  13 0012D          BEQL    11$                                 ;
                                   0094    C3  9F 0012F          PUSHAB  SDL_FAB                             ; 0690
                               64          01  FB 00133          CALLS   #1, SYS$CLOSE                       ;
                               52          50  D0 00136          MOVL    R0, STATUS                          ;
                               0D          52  E8 00139          BLBS    STATUS, 11$                         ; 0691
                                   0094    C3  9F 0013C          PUSHAB  SDL_FAB                             ; 0693
                              0C97121A 8F  DD 00140          PUSHL   #9900570                                ;
                           65          02  FB 00146          CALLS   #2, RMS_ERROR                           ;
                           50      0128 C3  D0 00149  11$:     MOVL    WORST_ERROR, R0                         ; 0696
                                       04  00 0014E          RET                                             ; 0698

; Routine Size:  335 bytes,    Routine Base:  $CODE$ + 0003
```

```
;  465     0699  1 %SBTTL 'GET_QUALS - Parse command line.'
;  466     0700  1 ROUTINE get_quals : NOVALUE =
;  467     0701  1
;  468     0702  1 !----
;  469     0703  1 !
;  470     0704  1 ! Functional description
;  471     0705  1 !
;  472     0706  1 !       This routine uses cli$present and cli$get_value to set a flag word
;  473     0707  1 !       indicating which qualifiers are present in the comand line and to get
;  474     0708  1 !       the values for qualifiers which have them.  It also gets the input
;  475     0709  1 !       filenames and puts them into a queue and determines related filenames
;  476     0710  1 !       for output files.                                                        ; R
;  477     0711  1 !
;  478     0712  1 ! Input parameters
;  479     0713  1 !
;  480     0714  1 !       None
;  481     0715  1 !
;  482     0716  1 ! Implicit inputs
;  483     0717  1 !
;  484     0718  1 !       The command line and the associated rms data structures for specified
;  485     0719  1 !       files are implicit inputs.
;  486     0720  1 !
;  487     0721  1 ! Output parameters
;  488     0722  1 !
;  489     0723  1 !       cli_flags       = Bitmap indicating which qualifiers are present
;  490     0724  1 !
;  491     0725  1 ! Implicit outputs
;  492     0726  1 !
;  493     0727  1 !       A queue of file input file information and filling in of RMS NAM blocks
;  494     0728  1 !       for ouput files.
;  495     0729  1 !
;  496     0730  1 !----
;  497     0731  1
;  498     0732  1 ! AUTHOR:  Tim Halvorsen, Nov 1979
;  499     0733  1 !
;  500     0734  1 ! Modified by:
;  501     0735  1 !
;  502     0736  1 !       V03-001 GJA0049         Greg Awdziewicz         8-Sep-1983
;  503     0737  1 !               Change to new CLI interface.
;  504     0738  1 !               Merge into MAIN module.
;  505     0739  1 !               Change qualifiers (/LISTING -> /LIST, /FILE -> /FILE_NAME) for
;  506     0740  1 !               consistency with other utilities and with documentation.
;  507     0741  1 !
;  508     0742  1 !       V02-002 JWT0002         Jim Teague              06-NOV-1981
;  509     0743  1 !               Added /SDL capability
;  510     0744  1 !
;  511     0745  1 !       V02-001 PCG0001         Peter George            29-Dec-1980
;  512     0746  1 !               Add /MDL functionality
;  513     0747  1 !
;  514     0748  1 !--
;  515     0749  1
;  516     0750  2 BEGIN
;  517     0751  2 BUILTIN
;  518     0752  2     INSQUE;
;  519     0753  2 LOCAL
;  520     0754  2     first_parm_flag:                    ! First parameter gets special treatment.
;  521     0755  2         INITIAL (true),
```

```
;  522     0756  2     value:                                  ! Dynamic descriptor for cli$get_value
;  523     0757  2         BLOCK [dsc$k_s_bln, BYTE]           ! to use to return a value.
;  524     0758  2         PRESET ([dsc$w_length] = 0,
;  525     0759  2                 [dsc$b_class] = dsc$k_class_d,
;  526     0760  2                 [dsc$b_dtype] = dsc$k_dtype_t,
;  527     0761  2                 [dsc$a_pointer] = 0),
;  528     0762  2     result:                                 ! Dynamic descriptor for lib$find_file
;  529     0763  2         BLOCK [dsc$k_s_bln, BYTE]           ! to use to return the file spec.
;  530     0764  2         PRESET ([dsc$w_length] = 0,
;  531     0765  2                 [dsc$b_class] = dsc$k_class_d,
;  532     0766  2                 [dsc$b_dtype] = dsc$k_dtype_t,
;  533     0767  2                 [dsc$a_pointer] = 0),
;  534     0768  2     context:    INITIAL (0),                ! Context pointer for lib$find_file.
;  535     0769  2     default:                                ! Descriptor for lib$find_file
;  536     0770  2         BLOCK [dsc$k_s_bln, BYTE]           ! to use for defaults.
;  537     0771  2         PRESET ([dsc$b_class] = dsc$k_class_s,
;  538     0772  2                 [dsc$b_dtype] = dsc$k_dtype_t),
;  539     0773  2     stv,                                    ! RMS error status value.
;  540     0774  2     fflgs:      BITVECTOR [32]              ! Lib$find_file user flags:
;  541     0775  2         PRESET ([1] = true),                !  multiple input file stickyness.
;  542     0776  2
;  543     0777  2     status:     INITIAL (true);
;  544     0778  2
;  545     0779  2
;  546     0780  2 ! Set the default descriptor from the input file fab:
;  547     0781  2 !
;  548     0782  2 default[dsc$w_length] = .input_fab[fab$b_dns];
;  549     0783  2 default[dsc$a_pointer] = .input_fab[fab$l_dna];
;  550     0784  2                                                                                 ; R
;  551     0785  2
;  552     0786  2 ! Get a copy of the whole command line so that we can echo it in the listing:
;  553     0787  2 !
;  554     0788  2 cli$get_value (sd_$LINE, command_line_desc);
;  555     0789  2
;  556     0790  2
;  557     0791  2 !+
;  558     0792  2 !  Check first for the nonpositional global qualifiers (/File_name, /Symbols,
;  559     0793  2 !  and /Text).
;  560     0794  2 !-
;  561     0795  2
;  562     0796  2
;  563     0797  2 ! Check if /FILE_NAME qualifier is present and get filename to point to:
;  564     0798  2 !
;  565     0799  2 IF cli$present (sd_FILE_NAME)
;  566     0800  2 THEN
;  567     0801  3     BEGIN
;  568     0802  3     cli_flags[qual_file] = true;
;  569     0803  3     cli$get_value (sd_FILE_NAME, filename_desc);
;  570     0804  3     IF .filename_desc[dsc$w_length] GTRU nam$c_maxrss
;  571     0805  3     THEN
;  572     0806  3         SIGNAL (emsg(syntax), 1, filename_desc);
;  573     0807  2     END;
;  574     0808  2
;  575     0809  2
;  576     0810  2 ! Check if /SYMBOLS qualifier is present:
;  577     0811  2 !
;  578     0812  2 IF cli$present (sd_SYMBOLS)
```

```
;  579          0813  2 THEN
;  580          0814  2     cli_flags[qual_symbols] = true;
;  581          0815  2
;  582          0816  2
;  583          0817  2 ! Check if /TEXT qualifier is present:
;  584          0818  2 !
;  585          0819  2 IF cli$present (sd_TEXT)
;  586          0820  2 THEN
;  587          0821  2     cli_flags[qual_text] = true;
;  588          0822  2
;  589          0823  2
;  590          0824  2 !+
;  591          0825  2 !  Check next for the positional qualifiers (/List, /Mdl, /Object, and /Sdl)
;  592          0826  2 !  to see if any are used in a global sense (that is, to see if they were used
;  593          0827  2 !  as qualifiers on the command rather than on a file specification).
;  594          0828  2 !-
;  595          0829  2
;  596          0830  2
;  597          0831  2 ! Check if /LIST qualifier is present and get filename for listing output:
;  598          0832  2 !
;  599          0833  2
;  600          0834  2 IF cli$present (sd_LIST) THEN
;  601          0835  3     BEGIN
;  602          0836  3     cli_flags[qual_listing] = true;
;  603          0837  3     IF cli$get_value (sd_LIST, value) THEN
;  604          0838  4         BEGIN
;  605          0839  4         listing_fab [fab$b_fns] = .value[dsc$w_length];
;  606          0840  4         listing_fab [fab$l_fna] = .value[dsc$a_pointer];
;  607          0841  4         value[dsc$w_length] = 0;
;  608          0842  4         value[dsc$a_pointer] = 0;
;  609          0843  3         END;
;  610          0844  2     END;
;  611          0845  2
;  612          0846  2
;  613          0847  2 ! Check if /MDL qualifier is present and get filename for MDL output:
;  614          0848  2 !
;  615          0849  2 IF cli$present (sd_MDL) THEN
;  616          0850  3     BEGIN
;  617          0851  3     cli_flags[qual_MDL] = true;
;  618          0852  3     IF cli$get_value (sd_MDL, value) THEN
;  619          0853  4         BEGIN
;  620          0854  4         MDL_fab [fab$b_fns] = .value[dsc$w_length];
;  621          0855  4         MDL_fab [fab$l_fna] = .value[dsc$a_pointer];
;  622          0856  4         value[dsc$w_length] = 0;
;  623          0857  4         value[dsc$a_pointer] = 0;
;  624          0858  3         END;
;  625          0859  2     END;
;  626          0860  2
;  627          0861  2
;  628          0862  2 ! Check if /OBJECT qualifier is present and get filename for object output:
;  629          0863  2 !
;  630          0864  2 IF cli$present (sd_OBJECT) THEN
;  631          0865  3     BEGIN
;  632          0866  3     cli_flags[qual_object] = true;
;  633          0867  3     IF cli$get_value (sd_OBJECT, value) THEN
;  634          0868  4         BEGIN
;  635          0869  4         object_fab [fab$b_fns] = .value[dsc$w_length];
```

```
;  636     0870  4          object_fab [fab$l_fna] = .value[dsc$a_pointer];
;  637     0871  4          value[dsc$w_length] = 0;
;  638     0872  4          value[dsc$a_pointer] = 0;
;  639     0873  3          END;
;  640     0874  2      END;
;  641     0875  2
;  642     0876  2
;  643     0877  2 ! Check if /SDL qualifier is present and get filename for SDL output:
;  644     0878  2 !
;  645     0879  2 IF cli$present (sd_SDL) THEN
;  646     0880  3      BEGIN
;  647     0881  3      cli_flags[qual_SDL] = true;
;  648     0882  3      IF cli$get_value (sd_SDL, value) THEN
;  649     0883  4          BEGIN
;  650     0884  4          SDL_fab [fab$b_fns] = .value[dsc$w_length];
;  651     0885  4          SDL_fab [fab$l_fna] = .value[dsc$a_pointer];
;  652     0886  4          value[dsc$w_length] = 0;
;  653     0887  4          value[dsc$a_pointer] = 0;
;  654     0888  3          END;
;  655     0889  2      END;
;  656     0890  2
;  657     0891  2
;  658     0892  2 !+
;  659     0893  2 !  Finally, we go through the input filenames, to establish a queue of
;  660     0894  2 !  them for further processing and to check for local use of the
;  661     0895  2 !  positional qualifiers (/List, /Mdl, /Object, and /Sdl) to establish
;  662     0896  2 !  appropriate output file names.
;  663     0897  2 !  If a positional qualifier is used locally instead of globally we want
;  664     0898  2 !  to use as much of the local context as possible, so we use the
;  665     0899  2 !  resultant filename from Lib$Find_file as the related filename for the
;  666     0900  2 !  output file.  We also clear the OFP bit so that the output file will
;  667     0901  2 !  be put in the same directory as the input file.
;  668     0902  2 !  However, if the qualifier has a value we want to set the OFP back
;  669     0903  2 !  again, so that if the value specified does not include a device
;  670     0904  2 !  and/or a directory we will use the current default device and/or
;  671     0905  2 !  directory.
;  672     0906  2 !-
;  673     0907  2
;  674     0908  2 WHILE cli$get_value(sd_P1, value) AND .status DO
;  675     0909  3      BEGIN
;  676     0910  3      LOCAL
;  677     0911  3          file_info:
;  678     0912  3              REF BBLOCK FIELD (info_queue_flds);
;  679     0913  3
;  680     0914  3
;  681     0915  3      status = lib$find_file (value, result, context, default, 0, stv, ffflgs);
;  682     0916  3
;  683     0917  3      IF NOT .status THEN         ! Failure to find any input file is fatal.
;  684     0918  3          SIGNAL_STOP (emsg(searchfail), 1, value, .status, .stv)
;  685     0919  3      ELSE
;  686     0920  4          BEGIN
;  687     0921  4          lib$get_vm(%REF(queue_element_size), file_info);
;  688     0922  4          CH$MOVE (dsc$c_s_bln, value, file_info[name_desc]);
;  689     0923  4          value[dsc$w_length] = 0;
;  690     0924  4          value[dsc$a_pointer] = 0;
;  691     0925  4          INSQUE (.file_info, .file_queue_hdr[queue_blink]);
;  692     0926  4
```

```
;  693     0927  4
;  694     0928  4          ! We use the first file specification to provide the related name for
;  695     0929  4          ! any output files which were indicated with global qualifiers:
;  696     0930  4          !
;  697     0931  4          IF .first_parm_flag THEN
;  698     0932  5              BEGIN
;  699     0933  5              first_parm_flag = false;
;  700     0934  5              SELECT 1 OF
;  701     0935  5              SET
;  702     0936  5              [.cli_flags[qual_listing]]:
;  703     0937  5                  BEGIN
;  704     0938  6                  listing_rel_nam[nam$b_rsl] = .result[dsc$w_length];
;  705     0939  6                  CH$MOVE (.result[dsc$w_length], .result[dsc$a_pointer], listing_relname);
;  706     0940  5                  END;
;  707     0941  5              [.cli_flags[qual_MDL]]:
;  708     0942  6                  BEGIN
;  709     0943  6                  MDL_rel_nam[nam$b_rsl] = .result[dsc$w_length];
;  710     0944  6                  CH$MOVE (.result[dsc$w_length], .result[dsc$a_pointer], MDL_relname);
;  711     0945  5                  END;
;  712     0946  5              [.cli_flags[qual_object]]:
;  713     0947  6                  BEGIN
;  714     0948  6                  object_rel_nam[nam$b_rsl] = .result[dsc$w_length];
;  715     0949  6                  CH$MOVE (.result[dsc$w_length], .result[dsc$a_pointer], object_relname);
;  716     0950  5                  END;
;  717     0951  5              [.cli_flags[qual_SDL]]:
;  718     0952  6                  BEGIN
;  719     0953  6                  SDL_rel_nam[nam$b_rsl] = .result[dsc$w_length];
;  720     0954  6                  CH$MOVE (.result[dsc$w_length], .result[dsc$a_pointer], SDL_relname);
;  721     0955  5                  END;
;  722     0956  5              TES;
;  723     0957  4              END;
;  724     0958  4
;  725     0959  4          ! Check if /LIST qualifier is present or negated and, if present,  get
;  726     0960  4          ! filename for listing output:
;  727     0961  4          !
;  728     0962  4          SELECT cli$present (sd_LIST) OF
;  729     0963  4          SET
;  730     0964  4          [cli$_locneg]:
;  731     0965  4              cli_flags[qual_listing] = false;
;  732     0966  4          [cli$_locpres]:
;  733     0967  5              BEGIN
;  734     0968  5              cli_flags[qual_listing] = true;
;  735     0969  5              listing_fab[fab$v_ofp] = false;
;  736     0970  5              listing_rel_nam[nam$b_rsl] = .result[dsc$w_length];
;  737     0971  5              CH$MOVE (.result[dsc$w_length], .result[dsc$a_pointer], listing_relname);
;  738     0972  5              IF cli$get_value (sd_LIST, value) THEN
;  739     0973  6                  BEGIN
;  740     0974  6                  listing_fab[fab$v_ofp] = true;
;  741     0975  6                  listing_fab [fab$b_fns] = .value[dsc$w_length];
;  742     0976  6                  listing_fab [fab$l_fna] = .value[dsc$a_pointer];
;  743     0977  6                  value[dsc$w_length] = 0;
;  744     0978  6                  value[dsc$a_pointer] = 0;
;  745     0979  6                  END
;  746     0980  5              ELSE
;  747     0981  5                  listing_fab [fab$b_fns] = 0;
;  748     0982  4              END;
;  749     0983  4          TES;
```

```
;  750     0984  4
;  751     0985  4
;  752     0986  4          ! Check if /MDL qualifier is present or negated and, if present, get
;  753     0987  4          ! filename for MDL output:
;  754     0988  4          !
;  755     0989  4          SELECT cli$present (sd_MDL) OF
;  756     0990  4          SET
;  757     0991  4          [cli$_locneg]:
;  758     0992  4              cli_flags[qual_MDL] = false;
;  759     0993  4          [cli$_locpres]:
;  760     0994  5              BEGIN
;  761     0995  5              cli_flags[qual_MDL] = true;
;  762     0996  5              MDL_fab[fab$v_ofp] = false;
;  763     0997  5              MDL_rel_nam[nam$b_rsl] = .result[dsc$w_length];
;  764     0998  5              CH$MOVE (.result[dsc$w_length], .result[dsc$a_pointer], MDL_relname);
;  765     0999  5              IF cli$get_value (sd_MDL, value) THEN
;  766     1000  6                  BEGIN
;  767     1001  6                  MDL_fab[fab$v_ofp] = true;
;  768     1002  6                  MDL_fab [fab$b_fns] = .value[dsc$w_length];
;  769     1003  6                  MDL_fab [fab$l_fna] = .value[dsc$a_pointer];                    ; R
;  770     1004  6                  value[dsc$w_length] = 0;
;  771     1005  6                  value[dsc$a_pointer] = 0;
;  772     1006  6                  END
;  773     1007  5              ELSE
;  774     1008  5                  MDL_fab [fab$b_fns] = 0;
;  775     1009  4              END;
;  776     1010  4          TES;
;  777     1011  4
;  778     1012  4
;  779     1013  4          ! Check if /OBJECT qualifier is present or negated and, if present, get
;  780     1014  4          ! filename for object output:
;  781     1015  4          !
;  782     1016  4          SELECT cli$present (sd_OBJECT) OF
;  783     1017  4          SET
;  784     1018  4          [cli$_locneg]:
;  785     1019  4              cli_flags[qual_object] = false;
;  786     1020  4          [cli$_locpres]:
;  787     1021  5              BEGIN
;  788     1022  5              cli_flags[qual_object] = true;
;  789     1023  5              object_fab[fab$v_ofp] = false;
;  790     1024  5              object_rel_nam[nam$b_rsl] = .result[dsc$w_length];
;  791     1025  5              CH$MOVE (.result[dsc$w_length], .result[dsc$a_pointer], object_relname);
;  792     1026  5              IF cli$get_value (sd_OBJECT, value) THEN
;  793     1027  6                  BEGIN
;  794     1028  6                  object_fab[fab$v_ofp] = true;
;  795     1029  6                  object_fab [fab$b_fns] = .value[dsc$w_length];
;  796     1030  6                  object_fab [fab$l_fna] = .value[dsc$a_pointer];
;  797     1031  6                  value[dsc$w_length] = 0;
;  798     1032  6                  value[dsc$a_pointer] = 0;
;  799     1033  6                  END
;  800     1034  5              ELSE
;  801     1035  5                  object_fab [fab$b_fns] = 0;
;  802     1036  4              END;
;  803     1037  4          TES;
;  804     1038  4
;  805     1039  4
;  806     1040  4          ! Check if /SDL qualifier is present or negated and, if present, get
```

```
;  807          1041  4          ! filename for SDL output:
;  808          1042  4          !
;  809          1043  4          SELECT cli$present (sd_SDL) OF
;  810          1044  4          SET
;  811          1045  4          [cli$_locneg]:
;  812          1046  4              cli_flags[qual_SDL] = false;
;  813          1047  4          [cli$_locpres]:
;  814          1048  5              BEGIN
;  815          1049  5              cli_flags[qual_SDL] = true;
;  816          1050  5              SDL_fab[fab$v_ofp] = false;
;  817          1051  5              SDL_rel_nam[nam$b_rsl] = .result[dsc$w_length];
;  818          1052  5              CH$MOVE (.result[dsc$w_length], .result[dsc$a_pointer], SDL_relname);
;  819          1053  5              IF cli$get_value (sd_SDL, value) THEN
;  820          1054  6                  BEGIN
;  821          1055  6                  SDL_fab[fab$v_ofp] = true;
;  822          1056  6                  SDL_fab [fab$b_fns] = .value[dsc$w_length];
;  823          1057  6                  SDL_fab [fab$l_fna] = .value[dsc$a_pointer];
;  824          1058  6                  value[dsc$w_length] = 0;
;  825          1059  6                  value[dsc$a_pointer] = 0;
;  826          1060  6                  END
;  827          1061  5              ELSE
;  828          1062  5                  SDL_fab [fab$b_fns] = 0;
;  829          1063  4              END;
;  830          1064  4          TES;
;  831          1065  3          END;
;  832          1066  2      END;
;  833          1067  2
;  834          1068  1 END;


                                OFFC 00000 GET_QUALS:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 0700
                5B 00000000G    00   9E 00002          MOVAB   CLI$GET_VALUE, R11               ;
                5A      0000'   CF   9E 00009          MOVAB   SD_LIST, R10                     ;
                59      0000'   CF   9E 0000E          MOVAB   CLI_FLAGS, R9                    ;
                5E              2C   C2 00013          SUBL2   #44, SP                          ;
                58              01   D0 00016          MOVL    #1, FIRST_PARM_FLAG              ; 0750
          24    AE  020E0000    8F   D0 00019          MOVL    #34471936, VALUE                 ; 0761
                        28      AE   D4 00021          CLRL    VALUE+4                          ;
          1C    AE  020E0000    8F   D0 00024          MOVL    #34471936, RESULT                ; 0767
                        20      AE   D4 0002C          CLRL    RESULT+4                         ;
                        0C      AE   D4 0002F          CLRL    CONTEXT                          ;
          14    AE  010E0000    8F   D0 00032          MOVL    #17694720, DEFAULT               ; 0772
                        18      AE   D4 0003A          CLRL    DEFAULT+4                        ;
          04    AE              02   D0 0003D          MOVL    #2, FFFLGS                       ; 0775
                57              01   D0 00041          MOVL    #1, STATUS                       ;
          14    AE        51    A9   9B 00044          MOVZBW  INPUT_FAB+53, DEFAULT            ; 0782
          18    AE        4C    A9   D0 00049          MOVL    INPUT_FAB+48, DEFAULT+4          ; 0783
                        04      A9   9F 0004E          PUSHAB  COMMAND_LINE_DESC                ; 0788
                        E0      AA   9F 00051          PUSHAB  SD_$LINF                         ;
                6B              02   FB 00054          CALLS   #2, CLI$GET_VALUE                ;
                        F4      AA   9F 00057          PUSHAB  SD_FILE_NAME                     ; 0799
00000000G       00              01   FB 0005A          CALLS   #1, CLI$PRESENT                  ;
                26              50   E9 00061          BLBC    R0, 1$                           ;
```

```
                    69              04  88 00064          BISB2   #4, CLI_FLAGS                           ; 0802
                              14    A9  9F 00067          PUSHAB  FILENAME_DESC                           ; 0803
                              F4    AA  9F 0006A          PUSHAB  SD_FILE_NAME                            ;
                    6B              02  FB 0006D          CALLS   #2, CLI$GET_VALUE                       ;
                00FF  8F            14  A9  P1 00070      CMPW    FILENAME_DESC, #255                     ; 0804
                                    12  1B 00076          BLEQU   1$                                      ;
                                    14  A9  9F 00078      PUSHAB  FILENAME_DESC                           ; 0806
                                    01  DD 0007B          PUSHL   #1                                      ;
                          009710FC  8F  DD 0007D          PUSHL   #9900284                                ;
                00000000G   00          03  FB 00083      CALLS   #3, LIB$SIGNAL                          ;
                                    44  AA  9F 0008A 1$:   PUSHAB  SD_SYMBOLS                            ; 0812
                00000000G   00          01  FB 0008D      CALLS   #1, CLI$PRESENT                         ;
                            04          50  E9 00094      BLBC    R0, 2$                                  ;
                            69      40  8F  88 00097      BISB2   #64, CLI_FLAGS                          ; 0814
                                    50  AA  9F 0009B 2$:   PUSHAB  SD_TEXT                               ; 0819
                00000000G   00          01  FB 0009E      CALLS   #1, CLI$PRESENT                         ;
                            04          50  E9 000A5      BLBC    R0, 3$                                  ;
                            69      80  8F  88 000A8      BISB2   #128, CLI_FLAGS                         ; 0821
                                    5A  DD 000AC 3$:      PUSHL   R10                                     ; 0834
                00000000C   00          01  FB 000AE      CALLS   #1, CLI$PRESENT                         ;
                            20          50  E9 000B5      BLBC    R0, 4$                                  ;
                            69          01  88 000B8      BISB2   #1, CLI_FLAGS                           ; 0836
                                    24  AE  9F 000BB      PUSHAB  VALUE                                   ; 0837
                                    5A  DD 000BE          PUSHL   R10                                     ;
                    6B              02  FB 000C0          CALLS   #2, CLI$GET_VALUE                       ;
                    12              50  E9 000C3          BLBC    R0, 4$                                  ;
                00E4  C9        24  AE  90 000C6          MOVB    VALUE, LISTING_FAB+52                   ; 0839
                00DC  C9        28  AE  D0 000CC          MOVL    VALUE+4, LISTING_FAB+44                 ; 0840
                                24  AE  B4 000D2          CLRW    VALUE                                   ; 0841
                                28  AE  D4 000D5          CLRL    VALUE+4                                 ; 0842
                                0C  AA  9F 000D8 4$:      PUSHAB  SD_MDL                                  ; 0849
                00000000G   00          01  FB 000DB      CALLS   #1, CLI$PRESENT                         ;
                            21          50  E9 000E2      BLBC    R0, 5$                                  ;
                            69          08  88 000E5      BISB2   #8, CLI_FLAGS                           ; 0851
                                    24  AE  9F 000E8      PUSHAB  VALUE                                   ; 0852
                                    0C  AA  9F 000EB      PUSHAB  SD_MDL                                  ;
                    6B              02  FB 000EE          CALLS   #2, CLI$GET_VALUE                       ;
                    12              50  E9 000F1          BLBC    R0, 5$                                  ;
                0178  C9        24  AE  90 000F4          MOVB    VALUE, MDL_FAB+52                       ; 0854
                0170  C9        28  AE  D0 000FA          MOVL    VALUE+4, MDL_FAB+44                     ; 0855
                                24  AE  B4 00100          CLRW    VALUE                                   ; 0856
                                28  AE  D4 00103          CLRL    VALUE+4                                 ; 0857
                                1C  AA  9F 00106 5$:      PUSHAB  SD_OBJECT                               ; 0864
                00000000G   00          01  FB 00109      CALLS   #1, CLI$PRESENT                         ;
                            21          50  E9 00110      BLBC    R0, 6$                                  ;
                            69          02  88 00113      BISB2   #2, CLI_FLAGS                           ; 0866
                                    24  AE  9F 00116      PUSHAB  VALUE                                   ; 0867
                                    1C  AA  9F 00119      PUSHAB  SD_OBJECT                               ;
                    6B              02  FB 0011C          CALLS   #2, CLI$GET_VALUE                       ;
                    12              50  E9 0011F          BLBC    R0, 6$                                  ;
                0214  C9        24  AE  90 00122          MOVB    VALUE, OBJECT_FAB+52                    ; 0869
                020C  C9        28  AE  D0 00128          MOVL    VALUE+4, OBJECT_FAB+44                  ; 0870
                                24  AE  B4 0012E          CLRW    VALUE                                   ; 0871
                                28  AE  D4 00131          CLRL    VALUE+4                                 ; 0872
                                34  AA  9F 00134 6$:      PUSHAB  SD_SDL                                  ; 0879
                00000000G   00          01  FB 00137      CALLS   #1, CLI$PRESENT                         ;
                            21          50  E9 0013E      BLBC    R0, 8$                                  ;
```

```
                                69          10  88 00141          BISB2   #16, CLI_FLAGS                          ; 0881
                                        24  AE  9F 00144          PUSHAB  VALUE                                   ; 0882
                                        34  AA  9F 00147          PUSHAB  SD_SDL                                  ;
                                6B          02  FB 0014A          CALLS   #2, CLI$GET_VALUE                       ;
                                12          50  E9 0014D          BLBC    R0, 8$                                  ;
                         033C   C9      24  AE  90 00150 7$:      MOVB    VALUE, SDL_FAB+52                       ; 0884
                         0334   C9      28  AE  D0 00156          MOVL    VALUE+4, SDL_FAB+44                     ; 0885
                                        24  AE  B4 0015C          CLRW    VALUE                                   ; 0886
                                        28  AE  D4 0015F          CLRL    VALUE+4                                 ; 0887
                                        24  AE  9F 00162 8$:      PUSHAB  VALUE                                   ; 0908
                                        28  AA  9F 00165          PUSHAB  SD_P1                                   ;
                                6B          02  FB 00168          CALLS   #2, CLI$GET_VALUE                       ;
                                01          50  E8 0016B          BLBS    R0, 9$                                  ;
                                            04  00016E           RET                                             ;
                                01          57  E8 0016F 9$:      BLBS    STATUS, 10$                             ;
                                            04  00172           RET                                             ;
                                        04  AE  9F 00173 10$:     PUSHAB  FFFLGS                                  ; 0915
                                        0C  AE  9F 00176          PUSHAB  STV                                     ;
                                        7E  D4 00179          CLRL    -(SP)                                   ;
                                        20  AE  9F 0017B          PUSHAB  DEFAULT                                 ;
                                        1C  AE  9F 0017E          PUSHAB  CONTEXT                                 ;
                                        30  AE  9F 00181          PUSHAB  RESULT                                  ;
                                        3C  AE  9F 00184          PUSHAB  VALUE                                   ;
                     00000000G  00          07  FB 00187          CALLS   #7, LIB$FIND_FILE                       ;
                                57          50  D0 0018E          MOVL    R0, STATUS                              ;
                                19          57  E8 00191          BLBS    STATUS, 11$                             ; 0917
                                        08  AE  DD 00194          PUSHL   STV                                     ; 0918
                                        57  DD 00197          PUSHL   STATUS                                  ;
                                        2C  AE  9F 00199          PUSHAB  VALUE                                   ;
                                        01  DD 0019C          PUSHL   #1                                      ;
                                 0097123A   8F  DD 0019E          PUSHL   #9900602                                ;
                     00000000G  00          05  FB 001A4          CALLS   #5, LIB$STOP                            ;
                                            B5  11 001AB          BRB     8$                                      ;
                                        10  AE  9F 001AD 11$:     PUSHAB  FILE_INFO                               ; 0921
                                04  AE          10  D0 001B0          MOVL    #16, 4(SP)                              ;
                                        04  AE  9F 001B4          PUSHAB  4(SP)                                   ;
                     00000000G  00          02  FB 001B7          CALLS   #2, LIB$GET_VM                          ;
                                56          10  AE  D0 001BE          MOVL    FILE_INFO, R6                           ; 0922
                08  A6          24  AE          08  28 001C2          MOVC3   #8, VALUE, 8(R6)                        ;
                                        24  AE  B4 001C8          CLRW    VALUE                                   ; 0923
                                        28  AE  D4 001CB          CLRL    VALUE+4                                 ; 0924
                                10  B9          66  0E 001CE          INSQUE  (R6), @FILE_QUEUE_HDR+4                 ; 0925
                                49          58  E9 001D2          BLBC    FIRST_PARM_FLAG, T5$                    ; 0931
                                            58  D4 001D5          CLRL    FIRST_PARM_FLAG                         ; 0933
                                0E          69  E9 001D7          BLBC    CLI_FLAGS, 12$                          ; 0936
                          0000' CF          1C  AE  90 001DA          MOVB    RESULT, LISTING_REL_NAM+3               ; 0938
0000'  CF          20  BE                  1C  AE  28 001E0          MOVC3   RESULT, @RESULT+4, LISTING_RELNAME      ; 0939
       0E                  69              03  E1 001E8 12$:     BBC     #3, CLI_FLAGS, 13$                      ; 0941
                          0000' CF          1C  AE  90 001EC          MOVB    RESULT, MDL_REL_NAM+3                   ; 0943
0000'  CF          20  BE                  1C  AE  28 001F2          MOVC3   RESULT, @RESULT+4, MDL_RELNAME          ; 0944
       0E                  69              01  E1 001FA 13$:     BBC     #1, CLI_FLAGS, 14$                      ; 0946
                          0000' CF          1C  AE  90 001FE          MOVB    RESULT, OBJECT_REL_NAM+3                ; 0948
0000'  CF          20  BE                  1C  AE  28 00204          MOVC3   RESULT, @RESULT+4, OBJECT_RELNAME       ; 0949
       0E                  69              04  E1 0020C 14$:     BBC     #4, CLI_FLAGS, 15$                      ; 0951
                          0000' CF          1C  AE  90 00210          MOVB    RESULT, SDL_REL_NAM+3                   ; 0953
0000'  CF          20  BE                  1C  AE  28 00216          MOVC3   RESULT, @RESULT+4, SDL_RELNAME          ; 0954
                                            5A  DD 0021E 15$:     PUSHL   R10                                     ; 0962
```

```
                00000000G  00              01  FB 00220          CALLS   #1, CLI$PRESENT                                  ;
                00000000G  8F              50  D1 00227          CMPL    R0, #CLI$_LOCNEG                                ; 0964
                                           03  12 0022E          BNEQ    16$                                             ;
                           69              01  8A 00230          BICB2   #1, CLI_FLAGS                                   ; 0965
                00000000G  8F              50  D1 00233 16$:     CMPL    R0, #CLI$_LOCPRES                               ; 0966
                                           3E  12 0023A          BNEQ    18$                                             ;
                           69              01  88 0023C          BISB2   #1, CLI_FLAGS                                   ; 0968
                    00B7   C9              20  8A 0023F          BICB2   #32, LISTING_FAB+7                              ; 0969
                    0000'  CF          1C  AE  90 00244          MOVB    RESULT, LISTING_REL_NAM+3                       ; 0970
          0000'  CF   20   BE          1C  AE  28 0024A          MOVC3   RESULT, @RESULT+4, LISTING_RELNAME              ; 0971
                                       24  AE  9F 00252          PUSHAB  VALUE                                           ; 0972
                                           5A  DD 00255          PUSHL   R10                                             ;
                           6B              02  FB 00257          CALLS   #2, CLI$GET_VALUE                               ;
                           19              50  E9 0025A          BLBC    R0, 17$                                         ;
                    00B7   C9              20  88 0025D          BISB2   #32, LISTING_FAB+7                              ; 0974
                    00E4   C9          24  AE  90 00262          MOVB    VALUE, LISTING_FAB+52                           ; 0975
                    00DC   C9          28  AE  D0 00268          MOVL    VALUE+4, LISTING_FAB+44                         ; 0976
                                       24  AE  B4 0026E          CLRW    VALUE                                           ; 0977
                                       28  AE  D4 00271          CLRL    VALUE+4                                         ; 0978
                                           04  11 00274          BRB     18$                                             ; 0972
                                    00E4   C9  94 00276 17$:     CLRB    LISTING_FAB+52                                  ; 0981
                                       0C  AA  9F 0027A 18$:     PUSHAB  SD_MDL                                          ; 0989
                00000000G  00              01  FB 0027D          CALLS   #1, CLI$PRESENT                                 ;
                00000000G  8F              50  D1 00284          CMPL    R0, #CLI$_LOCNEG                                ; 0991
                                           03  12 0028B          BNEQ    19$                                             ;
                           69              03  8A 0028D          BICB2   #8, CLI_FLAGS                                   ; 0992
                00000000G  8F              50  D1 00290 19$:     CMPL    R0, #CLI$_LOCPRES                               ; 0993
                                           3F  12 00297          BNEQ    21$                                             ;
                           69              08  88 00299          BISB2   #8, CLI_FLAGS                                   ; 0995
                    014B   C9              20  8A 0029C          BICB2   #32, MDL_FAB+7                                  ; 0996
                    0000'  CF          1C  AE  90 002A1          MOVB    RESULT, MDL_REL_NAM+3                           ; 0997
          0000'  CF   20   BE          1C  AE  28 002A7          MOVC3   RESULT, @RESULT+4, MDL_RELNAME                  ; 0998
                                       24  AE  9F 002AF          PUSHAB  VALUE                                           ; 0999
                                       0C  AA  9F 002B2          PUSHAB  SD_MDL                                          ;
                           6B              02  FB 002B5          CALLS   #2, CLI$GET_VALUE                               ;
                           19              50  E9 002B8          BLBC    R0, 20$                                         ;
                    014B   C9              20  88 002BB          BISB2   #32, MDL_FAB+7                                  ; 1001
                    0178   C9          24  AE  90 002C0          MOVB    VALUE, MDL_FAB+52                               ; 1002
                    0170   C9          28  AE  D0 002C6          MOVL    VALUE+4, MDL_FAB+44                             ; 1003
                                       24  AE  B4 002CC          CLRW    VALUE                                           ; 1004
                                       28  AE  D4 002CF          CLRL    VALUE+4                                         ; 1005
                                           04  11 002D2          BRB     21$                                             ; 0999
                                    0178   C9  94 002D4 20$:     CLRB    MDL_FAB+52                                      ; 1008
                                       1C  AA  9F 002D8 21$:     PUSHAB  SD_OBJECT                                       ; 1016
                00000000G  00              01  FB 002DB          CALLS   #1, CLI$PRESENT                                 ;
                00000000G  8F              50  D1 002E2          CMPL    R0, #CLI$_LOCNEG                                ; 1018
                                           03  12 002E9          BNEQ    22$                                             ;
                           69              02  8A 002EB          BICB2   #2, CLI_FLAGS                                   ; 1019
                00000000G  8F              50  D1 002EE 22$:     CMPL    R0, #CLI$_LOCPRES                               ; 1020
                                           3F  12 002F5          BNEQ    24$                                             ;
                           69              02  88 002F7          BISB2   #2, CLI_FLAGS                                   ; 1022
                    01E7   C9              20  8A 002FA          BICB2   #32, OBJECT_FAB+7                               ; 1023
                    0000'  CF          1C  AE  90 002FF          MOVB    RESULT, OBJECT_REL_NAM+3                        ; 1024
          0000'  CF   20   BE          1C  AE  28 00305          MOVC3   RESULT, @RESULT+4, OBJECT_RELNAME               ; 1025
                                       24  AE  9F 0030D          PUSHAB  VALUE                                           ; 1026
                                       1C  AA  9F 00310          PUSHAB  SD_OBJECT                                       ;
                           6B              02  FB 00313          CALLS   #2, CLI$GET_VALUE                               ;
```

```
                        19            50  E9 00316         BLBC    R0, 23$                              ;
                  01E7  C9            20  88 00319         BISB2   #32, OBJECT_FAB+7                    ; 1028
                  0214  C9        24  AE  90 0031E         MOVB    VALUE, OBJECT_FAB+52                 ; 1029
                  020C  C9        28  AE  D0 00324         MOVL    VALUE+4, OBJECT_FAB+44               ; 1030
                                  24  AE  B4 0032A         CLRW    VALUE                                ; 1031
                                  28  AE  D4 0032D         CLRL    VALUE+4                              ; 1032
                                      04  11 0033C         BRB     24$                                  ; 1026
                              0214    C9  94 00332 23$:    CLRB    OBJECT_FAB+52                        ; 1035
                                  34  AA  9F 00336 24$:    PUSHAB  SD_SDL                               ; 1043
              00000000G  00           01  FB 00339         CALLS   #1, CLI$PRESENT                      ;
              00000000G  8F           50  D1 00340         CMPL    R0, #CLI$_LOCNEG                     ; 1045
                                      03  12 00347         BNEQ    25$                                  ;
                        69            10  8A 00349         BICB2   #16, CLI_FLAGS                       ; 1046
              00000000G  8F           50  D1 0034C 25$:    CMPL    R0, #CLI$_LOCPRES                    ; 1047
                                      2E  12 00353         BNEQ    27$                                  ;
                        69            10  88 00355         BISB2   #16, CLI_FLAGS                       ; 1049
                  030F  C9            20  8A 00358         BICB2   #32, SDL_FAB+7                       ; 1050
                  0000'  CF       1C  AE  90 0035D         MOVB    RESULT, SDL_REL_NAM+3                ; 1051
      0000'  CF     20  BE        1C  AE  28 00363         MOVC3   RESULT, @RESULT+4, SDL_RELNAME       ; 1052
                                  24  AE  9F 0036B         PUSHAB  VALUE                                ; 1053
                                  34  AA  9F 0036E         PUSHAB  SD_SDL                               ;
                        6B            02  FB 00371         CALLS   #2, CLI$GET_VALUE                    ;
                        08            50  E9 00374         BLBC    R0, 26$                              ;
                  030F  C9            20  88 00377         BISB2   #32, SDL_FAB+7                       ; 1055
                                    FDD1  31 0037C         BRW     7$                                   ; 1056
                              033C    C9  94 0037F 26$:    CLRB    SDL_FAB+52                           ; 1062
                                    FDDC  31 00383 27$:    BRW     8$                                   ; 0908
                                          04 00386         RET                                          ; 1068

; Routine Size:  903 bytes,    Routine Base:  $CODE$ + 0152
```

```
;  836     1069  1 %SBTTL 'RMS_ERROR - Report an RMS error.'
;  837     1070  1 GLOBAL ROUTINE rms_error (message, fab_block, rab_block) =
;  838     1071  1
;  839     1072  1 !---
;  840     1073  1 !
;  841     1074  1 !       This routine issues a multi-line error associated
;  842     1075  1 !       with an RMS operation.  The first parameter is the
;  843     1076  1 !       first message to be issued.  The second and third
;  844     1077  1 !       parameter is the address of the FAB and/or the RAB block
;  845     1078  1 !       which contains the error codes and the file name.
;  846     1079  1 !       If the third parameter is present, the RAB is assumed
;  847     1080  1 !       to contain the error status.
;  848     1081  1 !
;  849     1082  1 ! Inputs:
;  850     1083  1 !
;  851     1084  1 !       message = First message code to be output
;  852     1085  1 !       fab_block = Address of FAB block
;  853     1086  1 !       rab_block = Address of RAB block
;  854     1087  1 !
;  855     1088  1 ! Outputs:
;  856     1089  1 !
;  857     1090  1 !       The message is signaled.
;  858     1091  1 !---
;  859     1092  1
;  860     1093  2 BEGIN
;  861     1094  2
;  862     1095  2 BUILTIN
;  863     1096  2     NULLPARAMETER;                  ! True if parameter unspecified
;  864     1097  2
;  865     1098  2 MAP
;  866     1099  2     fab_block:  REF BBLOCK,         ! Define FAB structure
;  867     1100  2     rab_block:  REF BBLOCK;         ! Define RAB structure
;  868     1101  2
;  869     1102  2 LOCAL
;  870     1103  2     nam_block:  REF BBLOCK,         ! Define NAM structure
;  871     1104  2     desc:       VECTOR[2];          ! Temp. string descriptor
;  872     1105  2
;  873     1106  2 nam_block = .fab_block [fab$l_nam]; ! Get address of NAM block
;  874     1107  2 IF .nam_block EQL 0                 ! If no NAM block present,
;  875     1108  2 THEN
;  876     1109  3     BEGIN
;  877     1110  3     desc [0] = .fab_block [fab$b_fns],  ! then use original file string
;  878     1111  3     desc [1] = .fab_block [fab$l_fna];
;  879     1112  3     END
;  880     1113  2 ELSE IF .nam_block [nam$b_rsl] NEQ 0    ! If result string nonblank,
;  881     1114  2 THEN
;  882     1115  3     BEGIN
;  883     1116  3     desc [0] = .nam_block [nam$b_rsl];  ! then use it
;  884     1117  3     desc [1] = .nam_block [nam$l_rsa];
;  885     1118  3     END
;  886     1119  2 ELSE IF .nam_block [nam$b_esl] NEQ 0    ! If expanded string nonblank,
;  887     1120  2 THEN
;  888     1121  3     BEGIN
;  889     1122  3     desc [0] = .nam_block [nam$b_esl];  ! then use it
;  890     1123  3     desc [1] = .nam_block [nam$l_esa];
;  891     1124  3     END
;  892     1125  2 ELSE
```

```
;  893          1126  3     BEGIN
;  894          1127  3     desc [0] = .fab_block [fab$b_fns];  ! Else, use original file string
;  895          1128  3     desc [1] = .fab_block [fab$l_fna];
;  896          1129  2     END;
;  897          1130  2
;  898          1131  2 IF NOT NULLPARAMETER(3)                 ! If RAB argument specified,
;  899          1132  2 THEN
;  900          1133  2     SIGNAL(.message,1,desc,              ! Signal error
;  901          1134  2                 .rab_block [rab$l_sts], ! with RMS error code from RAB
;  902          1135  2                 .rab_block [rab$l_stv]) ! and secondary code
;  903          1136  2 ELSE
;  904          1137  3     IF .message EQL emsg(searchfail)     ! Failure to find an input file
;  905          1138  2     THEN SIGNAL_STOP(.message,1,desc,    ! is FATAL, just like other compilers
;  906          1139  2                 .fab_block [fab$l_sts], ! so SIGNAL and _STOP
;  907          1140  2                 .fab_block [fab$l_stv])
;  908          1141  2     ELSE
;  909          1142  2     SIGNAL(.message,1,desc,              ! Signal error
;  910          1143  2                 .fab_block [fab$l_sts], ! with RMS error code from FAB
;  911          1144  2                 .fab_block [fab$l_stv]);! and secondary code
;  912          1145  2
;  913          1146  2 RETURN true;
;  914          1147  2
;  915          1148  1 END;


                                0000 00000          .ENTRY  RMS_ERROR, Save nothing                 ; 1070
                             5E 08   C2 00002       SUBL2   #8, SP                                  ;
                             51 08   AC D0 00005       MOVL    FAB_BLOCK, R1                        ; 1106
                             50 28   A1 D0 00009       MOVL    40(R1), NAM_BLOCK                    ;
                                20   13 0000D       BEQL    2$                                   ; 1107
                                03   A0 95 0000F       TSTB    3(NAM_BLOCK)                         ; 1113
                                0B   13 00012       BEQL    1$                                   ;
                             6E 03   A0 9A 00014       MOVZBL  3(NAM_BLOCK), DESC                   ; 1116
                          04 AE 04   A0 D0 00018       MOVL    4(NAM_BLOCK), DESC+4                 ; 1117
                                19   11 0001D       BRB     3$                                   ; 1113
                                0B   A0 95 0001F 1$:    TSTB    11(NAM_BLOCK)                        ; 1119
                                0B   13 00022       BEQL    2$                                   ;
                             6E 0B   A0 9A 00024       MOVZBL  11(NAM_BLOCK), DESC                  ; 1122
                          04 AE 0C   A0 D0 00028       MOVL    12(NAM_BLOCK), DESC+4                ; 1123
                                09   11 0002D       BRB     3$                                   ; 1119
                             6E 54   A1 9A 0002F 2$:    MOVZBL  52(R1), DESC                         ; 1127
                          04 AE 2C   A1 D0 00033       MOVL    44(R1), DESC+4                       ; 1128
                             03      6C 91 00038 3$:    CMPB    (AP), #3                             ; 1131
                                0F   1F 0003B       BLSSU   4$                                   ;
                                0C   AC D5 0003D       TSTL    12(AP)                               ;
                                0A   13 00040       BEQL    4$                                   ;
                             50 0C   AC D0 00042       MOVL    RAB_BLOCK, R0                        ; 1135
                             7E 08   A0 7D 00046       MOVQ    8(R0), -(SP)                         ; 1134
                                23   11 0004A       BRB     6$                                   ; 1133
                0097123A     8F 04   AC D1 0004C 4$:    CMPL    MESSAGE, #9900602                    ; 1137
                                15   12 00054       BNEQ    5$                                   ;
                             7E 08   A1 7D 00056       MOVQ    8(R1), -(SP)                         ; 1139
                                08   AE 9F 0005A       PUSHAB  DESC                                 ; 1138
                                01   DD 0005D       PUSHL   #1                                   ;
```

```
                                 04     AC  DD 0005F          PUSHL   MESSAGE                                          ;
                    00000000G    00           05  FB 00062          CALLS   #5, LIB$STOP                             ;
                                              13  11 00069          BRB     7$                                       ;
                                 7E       08  A1  7D 00068 5$:      MOVQ    8(R1), -(SP)                             ; 1143
                                          08  AE  9F 0006F 6$:      PUSHAB  DESC                                     ; 1142
                                              01  DD 00072          PUSHL   #1                                       ;
                                          04  AC  DD 00074          PUSHL   MESSAGE                                  ;
                    00000000G    00           05  FB 00077          CALLS   #5, LIB$SIGNAL                           ;
                                 50           01  D0 0007E 7$:      MOVL    #1, R0                                   ; 1146
                                                  04 00081          RET                                              ; 1148

; Routine Size:  130 bytes,    Routine Base:  $CODE$ + 04D9
```

```
;  917          1149  1 %SBTTL 'SEARCH_ERROR - Error searching for input file.'
;  918          1150  1 ROUTINE search_error (fab) =
;  919          1151  1
;  920          1152  1 !---
;  921          1153  1 !
;  922          1154  1 !       Signal an error searching for an input file.
;  923          1155  1 !
;  924          1156  1 ! Inputs:
;  925          1157  1 !
;  926          1158  1 !       fab = Address of FAB used during searching
;  927          1159  1 !
;  928          1160  1 ! Outputs:
;  929          1161  1 !
;  930          1162  1 !       None
;  931          1163  1 !---
;  932          1164  1 RETURN rms_error (emsg(searchfail),,fab);



                                0000 00000 SEARCH_ERROR:
                                                .WORD   Save nothing                    ; 1150
                                04   AC  DD 00002         PUSHL   FAB                    ; 1164
                           0097123A  8F  DD 00005         PUSHL   #9900602               ;
                      FF6E   CF       02  FB 0000B         CALLS   #2, RMS_ERROR          ;
                                      04 00010         RET                            ;

; Routine Size:  17 bytes,    Routine Base:  $CODE$ + 055B
```

```
;  934          1165  1 %SBTTL 'DO_FILE - Processing associated with input file.'
;  935          1166  1 ROUTINE do_file =
;  936          1167  1
;  937          1168  1 !---
;  938          1169  1 !
;  939          1170  1 !       This routine is called once for each file specified
;  940          1171  1 !       in the input file list.  The file is opened and parsed
;  941          1172  1 !       and all message definitions are saved in virtual memory.
;  942          1173  1 !
;  943          1174  1 ! Inputs:
;  944          1175  1 !
;  945          1176  1 !       input_fab = Input FAB
;  946          1177  1 !       input_rab = Input RAB
;  947          1178  1 !
;  948          1179  1 ! Outputs:
;  949          1180  1 !
;  950          1181  1 !       Message definition blocks
;  951          1182  1 !---
;  952          1183  1
;  953          1184  2 BEGIN
;  954          1185  2
;  955          1186  2 LOCAL
;  956          1187  2     status,parse_status;                ! Status codes
;  957          1188  2
;  958          1189  2 status = $OPEN (FAB = input_fab);       ! Open the input file
;  959          1190  2 IF NOT .status                          ! If error found,
;  960          1191  2 THEN
;  961          1192  3     BEGIN
;  962          1193  3     rms_error (emsg(openin), input_fab); ! then signal the error
;  963          1194  3     RETURN .status;
;  964          1195  2     END;
;  965          1196  2
;  966          1197  2 status = $CONNECT (RAB = input_rab);    ! and connect the input stream
;  967          1198  2 IF NOT .status                          ! If error found,
;  968          1199  2 THEN
;  969          1200  3     BEGIN
;  970          1201  3     rms_error (emsg(openin), input_fab, input_rab); ! then signal the error
;  971          1202  3     RETURN .status;
;  972          1203  2     END;
;  973          1204  2
;  974          1205  2 status = get_module_name (input_fab);   ! Get module name
;  975          1206  2 IF NOT .status                          ! If error detected,
;  976          1207  2 THEN
;  977          1208  2     RETURN .status;                     ! return with status
;  978          1209  2
;  979          1210  2 !
;  980          1211  2 !       If the listing file has not yet been opened, open
;  981          1212  2 !       it now.  This is done after the input file open so
;  982          1213  2 !       that the name will be sticky with the input file name.
;  983          1214  2 !
;  984          1215  2
;  985          1216  2 IF .cli_flags [qual_listing]     ! If /LIST specified,
;  986          1217  2     AND .listing_fab [fab$w_ifi] EQL 0  ! and not already opened
;  987          1218  2 THEN
;  988          1219  3     BEGIN
;  989          1220  3     status = $CREATE (FAB = listing_fab);       ! Create the listing file
;  990          1221  3     IF NOT .status THEN rms_error (emsg(openout),listing_fab);
```

```
;  991          1222  3     status = $CONNECT (RAB = listing_rab);
;  992          1223  3     IF NOT .status THEN rms_error (emsg(openout),listing_fab,listing_rab);
;  993          1224  2     END;
;  994          1225  2
;  995          1226  2 IF .cli_flags [qual_mdl]                ! If /MDL specified
;  996          1227  2     AND .mdl_fab [fab$w_ifi] EQL 0       ! and not already opened
;  997          1228  2 THEN
;  998          1229  3     BEGIN
;  999          1230  3     status = $CREATE (FAB = mdl_fab);            ! Create the mdl file
; 1000          1231  3     IF NOT .status THEN rms_error (emsg(openout),mdl_fab);
; 1001          1232  3     status = $CONNECT (RAB = mdl_rab);
; 1002          1233  3     IF NOT .status THEN rms_error (emsg(openout),mdl_fab,mdl_rab);
; 1003          1234  2     END;
; 1004          1235  2
; 1005          1236  2 IF .cli_flags [qual_sdl]                ! If /SDL specified
; 1006          1237  2     AND .sdl_fab [fab$w_ifi] EQL 0       ! and not already opened
; 1007          1238  2 THEN
; 1008          1239  3     BEGIN
; 1009          1240  3     status = $CREATE (FAB = sdl_fab);            ! Create the sdl file
; 1010          1241  3     IF NOT .status THEN rms_error (emsg(openout),sdl_fab);
; 1011          1242  3     status = $CONNECT (RAB = sdl_rab);
; 1012          1243  3     IF NOT .status THEN rms_error (emsg(openout),sdl_fab,sdl_rab);
; 1013          1244  2     END;
; 1014          1245  2
; 1015          1246  2 parse_status = parse_file();             ! Parse the input file
; 1016          1247  2
; 1017          1248  2 status = $CLOSE (FAB = input_fab);       ! Close the input file
; 1018          1249  2 IF NOT .status                           ! If error found,
; 1019          1250  2 THEN
; 1020          1251  2     rms_error (emsg(closedel), input_fab); ! then signal the error
; 1021          1252  2
; 1022          1253  2 RETURN .parse_status;                    ! Return with parse status
; 1023          1254  2
; 1024          1255  1 END;


                                                        .EXTRN  SYS$OPEN

                                          00FC 00000 DO_FILE:.WORD   Save R2,R3,R4,R5,R6,R7                       ; 1166
                    57 00000000G   00   9E 00002          MOVAB   SYS$CREATE, R7                                  ;
                    56 00000000G   00   9E 00009          MOVAB   SYS$CONNECT, R6                                 ;
                    55       FF59  CF   9E 00010          MOVAB   RMS_ERROR, R5                                   ;
                    54       0000' CF   9E 00015          MOVAB   INPUT_FAB, R4                                   ;
                                   54   DD 0001A          PUSHL   R4                                              ; 1189
          00000000G 00             01   FB 0001C          CALLS   #1, SYS$OPEN                                    ;
                    52             50   D0 00023          MOVL    R0, STATUS                                      ;
                    0D             52   E8 00026          BLBS    STATUS, 1$                                      ; 1190
                                   54   DD 00029          PUSHL   R4                                              ; 1193
                       0097109A    8F   DD 0002B          PUSHL   #9900186                                        ;
                    65             02   FB 00031          CALLS   #2, RMS_ERROR                                   ;
                                   29   11 00034          BRB     3$                                              ; 1194
                             50    A4   9F 00036 1$:      PUSHAB  INPUT_RAB                                       ; 1197
                    66             01   FB 00039          CALLS   #1, SYS$CONNECT                                 ;
                    52             50   D0 0003C          MOVL    R0, STATUS                                      ;
                    10             52   E8 0003F          BLBS    STATUS, 2$                                      ; 1198
                             50    A4   9F 00042          PUSHAB  INPUT_RAB                                       ; 1201
```

```
                                                54  DD 00045          PUSHL   R4                                      ;
                                    0097109A    8F  DD 00047          PUSHL   #9900186                                ;
                                65              03  FB 0004D          CALLS   #3, RMS_ERROR                           ;
                                                0D  11 00050          BRB     3$                                      ; 1202
                                                54  DD 00052 2$:      PUSHL   R4                                      ; 1205
                          0000V CF              01  FB 00054          CALLS   #1, GET_MODULE_NAME                     ;
                                52              50  D0 00059          MOVL    R0, STATUS                              ;
                                04              52  E8 0005C          BLBS    STATUS, 4$                              ; 1206
                                50              52  D0 0005F 3$:      MOVL    STATUS, R0                              ; 1208
                                                04  00062            RET                                             ;
                                3E        E4    A4  E9 00063 4$:      BLBC    CLI_FLAGS, 6$                           ; 1216
                                          0096  C4  B5 00067          TSTW    LISTING_FAB+2                           ; 1217
                                                38  12 0006B          BNEQ    6$                                      ;
                                          0094  C4  9F 0006D          PUSHAB  LISTING_FAB                             ; 1220
                                67              01  FB 00071          CALLS   #1, SYS$CREATE                          ;
                                52              50  D0 00074          MOVL    R0, STATUS                              ;
                                0D              52  E8 00077          BLBS    STATUS, 5$                              ; 1221
                                          0094  C4  9F 0007A          PUSHAB  LISTING_FAB                             ;
                                    009710A4    8F  DD 0007E          PUSHL   #9900196                                ;
                                65              02  FB 00084          CALLS   #2, RMS_ERROR                           ;
                                          00E4  C4  9F 00087 5$:      PUSHAB  LISTING_RAB                             ; 1222
                                66              01  FB 0008B          CALLS   #1, SYS$CONNECT                         ;
                                52              50  D0 0008E          MOVL    R0, STATUS                              ;
                                11              52  E8 00091          BLBS    STATUS, 6$                              ; 1223
                                          00E4  C4  9F 00094          PUSHAB  LISTING_RAB                             ;
                                          0094  C4  9F 00098          PUSHAB  LISTING_FAB                             ;
                                    009  0A4    8F  DD 0009C          PUSHL   #9900196                                ;
                                65              03  FB 000A2          CALLS   #3, RMS_ERROR                           ;
                          3E        E4    A4  03  E1 000A5 6$:      BBC     #3, CLI_FLAGS, 8$                       ; 1226
                                          012A  C4  B5 000AA          TSTW    MDL_FAB+2                               ; 1227
                                                38  12 000AE          BNEQ    8$                                      ;
                                          0128  C4  9F 000B0          PUSHAB  MDL_FAB                                 ; 1230
                                67              01  FB 000B4          CALLS   #1, SYS$CREATE                          ;
                                52              50  D0 000B7          MOVL    R0, STATUS                              ;
                                0D              52  E8 000BA          BLBS    STATUS, 7$                              ; 1231
                                          0128  C4  9F 000BD          PUSHAB  MDL_FAB                                 ;
                                    009710A4    8F  DD 000C1          PUSHL   #9900196                                ;
                                65              02  FB 000C7          CALLS   #2, RMS_ERROR                           ;
                                          0178  C4  9F 000CA 7$:      PUSHAB  MDL_RAB                                 ; 1232
                                66              01  FB 000CE          CALLS   #1, SYS$CONNECT                         ;
                                52              50  D0 000D1          MOVL    R0, STATUS                              ;
                                11              52  E8 000D4          BLBS    STATUS, 8$                              ; 1233
                                          0178  C4  9F 000D7          PUSHAB  MDL_RAB                                 ;
                                          0128  C4  9F 000DB          PUSHAB  MDL_FAB                                 ;
                                    009710A4    8F  DD 000DF          PUSHL   #9900196                                ;
                                65              03  FB 000E5          CALLS   #3, RMS_ERROR                           ;
                          3E        E4    A4  04  E1 000E8 8$:      BBC     #4, CLI_FLAGS, 10$                      ; 1236
                                          02EE  C4  B5 000ED          TSTW    SDL_FAB+2                               ; 1237
                                                38  12 000F1          BNEQ    10$                                     ;
                                          02EC  C4  9F 000F3          PUSHAB  SDL_FAB                                 ; 1240
                                67              01  FB 000F7          CALLS   #1, SYS$CREATE                          ;
                                52              50  D0 000FA          MOVL    R0, STATUS                              ;
                                0D              52  E8 000FD          BLBS    STATUS, 9$                              ; 1241
                                          02EC  C4  9F 00100          PUSHAB  SDL_FAB                                 ;
                                    009710A4    8F  DD 00104          PUSHL   #9900196                                ;
                                65              02  FB 0010A          CALLS   #2, RMS_ERROR                           ;
                                          033C  C4  9F 0010D 9$:      PUSHAB  SDL_RAB                                 ; 1242
```

```
                                66              01  FB 00111          CALLS   #1, SYS$CONNECT                                ;
                                52              50  D0 00114          MOVL    R0, STATUS                                    ;
                                11              52  E8 00117          BLBS    STATUS, 10$                                   ; 1243
                                        033C    C4  9F 0011A          PUSHAB  SDL_RAB                                       ;
                                        02EC    C4  9F 0011E          PUSHAB  SDL_FAB                                       ;
                                    009710A4    8F  DD 00122          PUSHL   #9900196                                      ;
                                65              03  FB 00128          CALLS   #3, RMS_ERROR                                 ;
                          0000G CF              00  FB 0012B 10$:     CALLS   #0, PARSE_FILE                                ; 1246
                                53              50  D0 00130          MOVL    R0, PARSE_STATUS                              ;
                                                54  DD 00133          PUSHL   R4                                            ; 1248
                    00000000G   00              01  FB 00135          CALLS   #1, SYS$CLOSE                                 ;
                                52              50  D0 0013C          MOVL    R0, STATUS                                    ;
                                0B              52  E8 0013F          BLBS    STATUS, 11$                                   ; 1249
                                                54  DD 00142          PUSHL   R4                                            ; 1251
                                    0097121A    8F  DD 00144          PUSHL   #9900570                                      ;
                                65              02  FB 0014A          CALLS   #2, RMS_ERROR                                 ;
                                50              53  D0 0014D 11$:     MOVL    PARSE_STATUS, R0                              ; 1253
                                                    04 00150          RET                                                   ; 1255

; Routine Size:  337 bytes,    Routine Base:  $CODE$ + 056C
```

```
;  1026     1256  1 %SBTTL 'GET_MODULE_NAME'
;  1027     1257  1 GLOBAL ROUTINE get_module_name (fab) =
;  1028     1258  1
;  1029     1259  1 !---
;  1030     1260  1 !
;  1031     1261  1 !       This routine figures out the module name to be
;  1032     1262  1 !       associated with the given file name.  The
;  1033     1263  1 !       module name used is the name of the file.
;  1034     1264  1 !
;  1035     1265  1 ! Inputs:
;  1036     1266  1 !
;  1037     1267  1 !       None
;  1038     1268  1 !
;  1039     1269  1 ! Outputs:
;  1040     1270  1 !
;  1041     1271  1 !       module_name = Descriptor of module name string
;  1042     1272  1 !       r0 = status (unsignaled)
;  1043     1273  1 !---
;  1044     1274  1
;  1045     1275  2 BEGIN
;  1046     1276  2
;  1047     1277  2 MAP
;  1048     1278  2     fab:        REF BBLOCK;             ! Address of FAB block
;  1049     1279  2
;  1050     1280  2 LOCAL
;  1051     1281  2     input_nam:  REF BBLOCK,             ! Address of NAM block
;  1052     1282  2     addr,size,                          ! Temporary descriptor
;  1053     1283  2     ptr;                                ! String pointer
;  1054     1284  2
;  1055     1285  2 input_nam = .fab [fab$l_nam];           ! Get NAM address
;  1056     1286  2 size = .input_nam [nam$b_rsl];          ! Get result name string
;  1057     1287  2 addr = .input_nam [nam$l_rsa];
;  1058     1288  2
;  1059     1289  2 ptr = CH$FIND_CH(.size, .addr, ']');    ! Find start of file name on input side
;  1060     1290  2 IF .ptr EQL 0                           ! If not found,
;  1061     1291  2 THEN
;  1062     1292  3     BEGIN
;  1063     1293  3     ptr = CH$FIND_CH(.size, .addr, '>'); ! Alternate syntax
;  1064     1294  3     IF .ptr EQL 0                       ! If still not found
;  1065     1295  4     THEN BEGIN                          ! Then make up a name
;  1066     1296  4          module_name [0] = %CHARCOUNT ('MESSAGE');
;  1067     1297  4          module_name [1] = UPLIT ('MESSAGE');
;  1068     1298  4          RETURN true;
;  1069     1299  3          END;
;  1070     1300  2     END;
;  1071     1301  2
;  1072     1302  2 size = .size - (.ptr + 1 - .addr);      ! Figure descriptor of file name
;  1073     1303  2 addr = .ptr + 1;
;  1074     1304  2
;  1075     1305  2 ptr = CH$FIND_CH(.size, .addr, '.');    ! Find where file name ends
;  1076     1306  2 IF .ptr EQL 0                           ! If not found,
;  1077     1307  2 THEN
;  1078     1308  2     RETURN rms$_rst;                    ! return invalid expanded string
;  1079     1309  2
;  1080     1310  2 module_name [0] = .ptr - .addr;         ! Figure descriptor of file name only
;  1081     1311  2 module_name [1] = .addr;
;  1082     1312  2
```

```
;  1083          1313  2 RETURN true;
;  1084          1314  2
;  1085          1315  1 END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2
                    00  45  47  41  53  53  45  4D  000A0 P.AAY:  .ASCII  \MESSAGE\<0>                                    ;


                                                                .PSECT  $CODE$,NOWRT,2

                                                  001C 00000    .ENTRY  GET_MODULE_NAME, Save R2,R3,R4                    ; 1257
                                  54      0000'   CF   9E 00002 MOVAB   MODULE_NAME, R4                                  ;
                                  50        04    AC   D0 00007 MOVL    FAB, R0                                         ; 1285
                                  50        28    A0   D0 0000B MOVL    40(R0), INPUT_NAM                               ;
                                  53        03    A0   9A 0000F MOVZBL  3(INPUT_NAM), SIZE                              ; 1286
                                  52        04    A0   D0 00013 MOVL    4(INPUT_NAM), ADDR                              ; 1287
                        62        53        5D    8F   3A 00017 LOCC    #93, SIZE, (ADDR)                               ; 1289
                                                    02   12 0001C BNEQ    1$                                              ;
                                                    51   D4 0001E CLRL    R1                                              ;
                                                    51   D5 00020 1$:    TSTL    PTR                                     ; 1290
                                                    17   12 00022 BNEQ    3$                                              ;
                        62        53              3E   3A 00024 LOCC    #62, SIZE, (ADDR)                               ; 1293
                                                    02   12 00028 BNEQ    2$                                              ;
                                                    51   D4 0002A CLRL    R1                                              ;
                                                    51   D5 0002C 2$:    TSTL    PTR                                     ; 1294
                                                    0B   12 0002E BNEQ    3$                                              ;
                                  64                07   D0 00030 MOVL    #7, MODULE_NAME                                 ; 1296
                              04  A4      0000'     CF   9E 00033 MOVAB   P.AAY, MODULE_NAME+4                            ; 1297
                                                    29   11 00039 BRB     6$                                              ; 1298
                        50        52                51   C3 0003B 3$:    SUBL3   PTR, ADDR, R0                           ; 1302
                                  53          FF  A043  9E 0003F MOVAB   -1(R0)[SIZE], SIZE                              ;
                                  52          01    A1  9E 00044 MOVAB   1(R1), ADDR                                    ; 1303
                        62        53                2E   3A 00048 LOCC    #46, SIZE, (ADDR)                               ; 1305
                                                    02   12 0004C BNEQ    4$                                              ;
                                                    51   D4 0004E CLRL    R1                                              ;
                                                    51   D5 00050 4$:    TSTL    PTR                                     ; 1306
                                                    08   12 00052 BNEQ    5$                                              ;
                                  50 0001869C       8F   D0 00054 MOVL    #99996, R0                                      ; 1308
                                                    04   00057 RET                                                         ;
                        64        51                52   C3 0005C 5$:    SUBL3   ADDR, PTR, MODULE_NAME                  ; 1310
                              04  A4                52   D0 00060 MOVL    ADDR, MODULE_NAME+4                             ; 1311
                                  50                01   D0 00064 6$:    MOVL    #1, R0                                  ; 1313
                                                    04   00067 RET                                                         ; 1315

; Routine Size:  104 bytes,    Routine Base:  $CODE$ + 06BD
```

```
;                1316  1 %SBTTL 'HANDLER - Save most severe error code.'
;                1317  1 ROUTINE handler (signal_args, mechanism_args) =
;                1318  1
;                1319  1 !---
;                1320  1 !
;                1321  1 !       This condition handler gets control on any signalled
;                1322  1 !       condition in order to save the highest severity error
;                1323  1 !       to be returned by exit from the image.
;                1324  1 !
;                1325  1 ! Inputs:
;                1326  1 !
;                1327  1 !       signal_args = Address of signal argument list
;                1328  1 !       mechanism_args = Address of mechanism argument list
;                1329  1 !
;                1330  1 ! Outputs:
;                1331  1 !
;                1332  1 !       worst_error is updated with highest severity error.                                             ; Rc
;                1333  1 !
;                1334  1 !---
;                1335  1
;                1336  2 BEGIN
;                1337  2
;                1338  2 MAP
;                1339  2     signal_args: REF BBLOCK,                   ! Address of signal arg list
;                1340  2     mechanism_args: REF BBLOCK;                ! Address of mechanism arg list
;                1341  2
;                1342  2 LOCAL
;                1343  2     code: BBLOCK [LONG];                       ! Condition code (longword)
;                1344  2
;                1345  2 code = .signal_args [chf$l_sig_name];          ! Get condition code
;                1346  2
;                1347  2 IF .code [sts$v_severity] GTR .worst_error [sts$v_severity]
;                1348  2 THEN
;                1349  2     worst_error = .code OR sts$m_inhib_msg;    ! Set new worst error
;                1350  2
;                1351  2 ss$_resignal                                  ! Continue signalling
;                1352  2
;                1353  1 END;


                                                  0000 00000 HANDLER:.WORD   Save nothing                                    ; 1317
                                    50      04    AC   D0 00002          MOVL    SIGNAL_ARGS, R0                             ; 1345
                                    50      04    A0   D0 00006          MOVL    4(R0), CODE                                 ;
          51    0000'  CF           03            00   EF 0000A          EXTZV   #0, #3, WORST_ERROR, R1                     ; 1347
          51           50           03            00   ED 00011          CMPZV   #0, #3, CODE, R1                            ;
                                                  0A   15 00016          BLEQ    1$                                          ;
                0000'  CF           50 10000000   8F   C9 00018          BISL3   #268435456, CODE, WORST_ERROR               ; 1349
                                    50      0918  8F   3C 00022 1$:      MOVZWL  #2328, R0                                   ; 1353
                                                       04 00027          RET                                                 ;

; Routine Size:  40 bytes,    Routine Base:  $CODE$ + 0725
```

```
; 1126          1354  1 %SBTTL ''
; 1127          1355  1 END
; 1128          1356  0 ELUDOM



                                                        .EXTRN  LIB$SIGNAL, LIB$STOP

;                                  PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $PLIT$                         168  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                         5004  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $GLOBAL$                       928  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        1869  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                               Library Statistics

;                                      -------- Symbols --------      Pages       Processing
;       File                           Total    Loaded   Percent      Mapped      Time

;  _$255$DUA28:[SYSLIB]STARLET.L32;1     9776        84         0         581        00:01.0



;                                        COMMAND QUALIFIERS

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

; Size:          1869 code + 6100 data bytes
; Run Time:         00:48.1
; Elapsed Time:     02:50.2
; Lines/CPU Min:     1691
; Lexemes/CPU-Min: 37678
; Memory Used:  325 pages
; Compilation Complete
```

MSGINT
MDL

SDLGENREQ
REQ

MSGDEF
SDL

MAIN
LIS

CVTMSGCOM
FOR

LISTING
LIS

CVTMSG
LIS

MSG
REQ

MDLGEN
LIS

SDLGEN
LIS

PARSE
LIS

MESSAGES
LIS

OBJECT
LIS