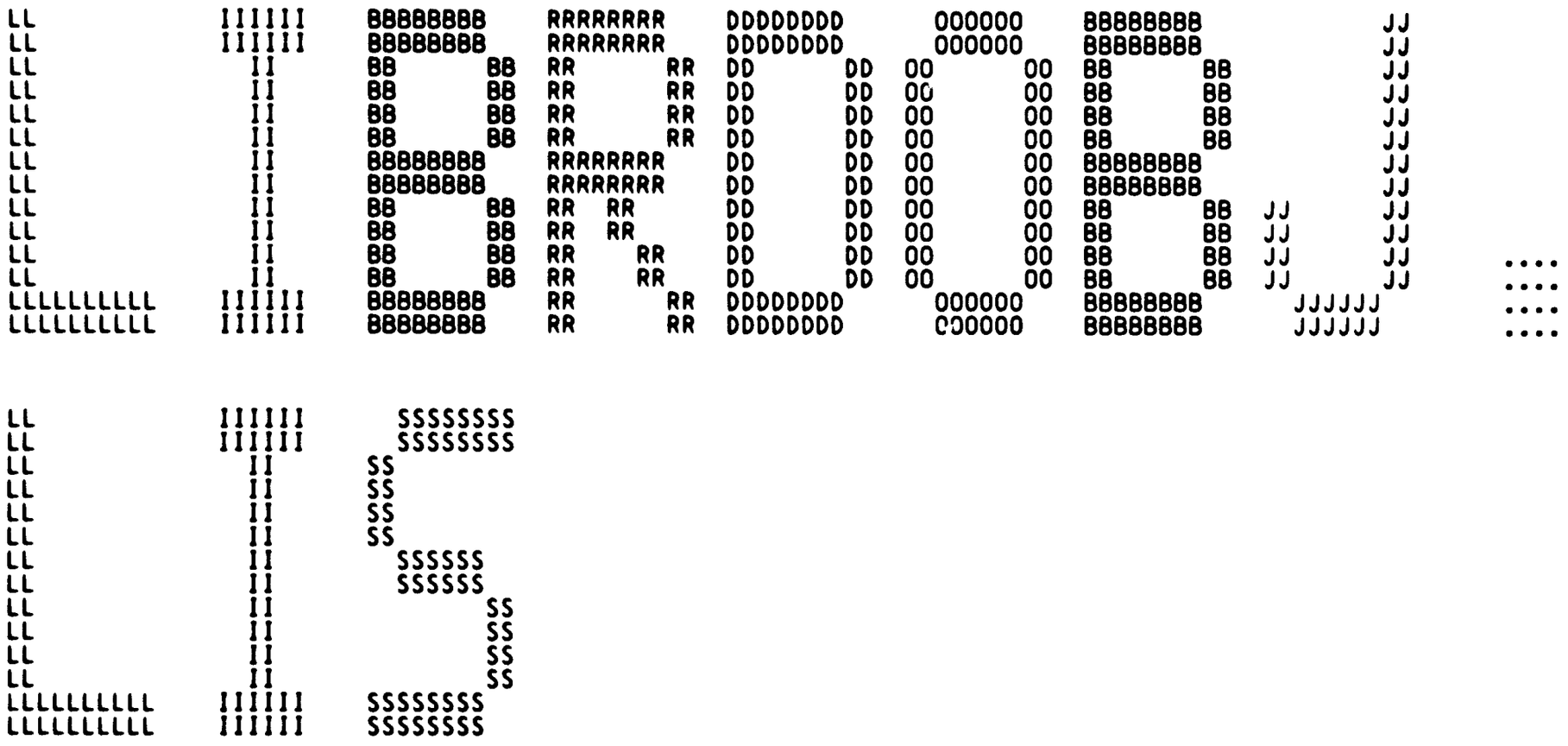
```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT       LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT       LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT       LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT       LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT       LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT       LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT       LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT       LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT       LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT       LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT       LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT       LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT       LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT       LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT       LLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT       LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT       LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT       LLLLLLLLLLLLLLL
```


LL IIIIII BBBBBBBB RRRRRRRR DDDDDDDD OOOOOO BBBBBBBB JJ
LL IIIIII BBBBBBBB RRRRRRRR DDDDDDDD OOOOOO BBBBBBBB JJ
LL II BB BB RR RR DD DD OO OO BB BB JJ
LL II BB BB RR RR DD DD OO OO BB BB JJ
LL II BB BB RR RR DD DD OO OO BB BB JJ
LL II BB BB RR RR DD DD OO OO BB BB JJ
LL II BBBBBBBB RRRRRRRR DD DD OO OO BBBBBBBB JJ
LL II BBBBBBBB RRRRRRRR DD DD OO OO BBBBBBBB JJ
LL II BB BB RR RR DD DD OO OO BB BB JJ JJ
LL II BB BB RR RR DD DD OO OO BB BB JJ JJ
LL II BB BB RR RR DD DD OO OO BB BB JJ JJ
LL II BB BB RR RR DD DD OO OO BB BB JJ JJ
LLLLLLLLLL IIIIII BBBBBBBB RR RR DDDDDDDD OOOOOO BBBBBBBB JJJJJJ
LLLLLLLLLL IIIIII BBBBBBBB RR RR DDDDDDDD OOOOOO BBBBBBBB JJJJJJ
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 lib$$read_object (                    ! File: LIBFNDIMG.B32 Edit: STAN3004
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 ADDRESSING_MODE(EXTERNAL=GENERAL),
;     4          0004  0                 IDENT = 'V03-004'
;     5          0005  0                 ) =
;     6          0006  1 BEGIN
;     7          0007  1 %TITLE  'Read and dissect object file';
;     8          0008  1 !
;     9          0009  1 !*****************************************************************************
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    12          0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    13          0013  1 !*  ALL RIGHTS RESERVED.                                                      *
;    14          0014  1 !*                                                                           *
;    15          0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    16          0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    17          0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    18          0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    19          0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    20          0020  1 !*  TRANSFERRED.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    23          0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    24          0024  1 !*  CORPORATION.                                                              *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    27          0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    28          0028  1 !*                                                                           *
;    29          0029  1 !*                                                                           *
;    30          0030  1 !*****************************************************************************
;    31          0031  1
;    32          0032  1 !++
;    33          0033  1 !
;    34          0034  1 ! FACILITY:  Run time library
;    35          0035  1 !
;    36          0036  1 ! ABSTRACT:
;    37          0037  1 !
;    38          0038  1 !       This procedure reads an object file and returns the symbols
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !
;    42          0042  1 !       VAX native, user mode.
;    43          0043  1 !
;    44          0044  1 !--
;    45          0045  1 !
;    46          0046  1 !
;    47          0047  1 ! AUTHOR:  Benn Schreiber
;    48          0048  1 !
;    49          0049  1 ! CREATION DATE: 23-Jan-1981
;    50          0050  1 !
;    51          0051  1 ! MODIFIED BY:
;    52          0052  1 !
;    53          0053  1 !       V03-004 STAN3004        Stanley Rabinowitz      24-Jul-1984
;    54          0054  1 !
;    55          0055  1 !       V03-003 BLS0277         Benn Schreiber           7-FEB-1984
;    56          0056  1 !               Convert to internal RTL routine.
;    57          0057  1 !
```

```
;    58          0058  1 !       V03-002 BLS0225         Benn Schreiber          16-Jun-1983
;    59          0059  1 !               Add flags argument and IMOD flag
;    60          0060  1 !
;    61          0061  1 !       V03-001 BLS0209         Benn Schreiber          27-Feb-1983
;    62          0062  1 !               Correct PSECT name for read/only OWN data
;    63          0063  1 !--
```

```
;    65          0064  1 %SBTTL  'Declarations';
;    66          0065  1 !
;    67          0066  1 ! BLISS Libraries
;    68          0067  1 !
;    69          0068  1 LIBRARY
;    70          0069  1     'RTLSTARLE';
;    71          0070  1 REQUIRE
;    72          0071  1     'RTLIN:RTLPSECT';
;    73          0166  1
;    74          0167  1 DECLARE_PSECTS (LIB);                   Declare psects for LIB facility
;    75          0168  1 !
;    76          0169  1 ! Data structure to describe object module
;    77          0170  1 !
;    78          0171  1 FIELD
;    79          0172  1     obc_fields =
;    80          0173  1         SET
;    81          0174  1         obc_l_gblrtn =  [0,0,32,0],         !Address of globals routine
;    82          0175  1         obc_l_pscrtn =  [4,0,32,0],         !Address of psect routine
;    83          0176  1         obc_l_eomrtn =  [8,0,32,0],         !Address of eom rec routine
;    84          0177  1         obc_l_ogsrtn =  [12,0,32,0],        !Address of other GSD routine
;    85          0178  1         obc_l_orcrtn =  [16,0,32,0],        !Address of other record routine
;    86          0179  1         obc_q_desc =    [20,0,0,0],         !Dynamic string descriptor
;    87          0180  1         obc_l_usrdata = [28,0,32,0],        !User data to pass to routines
;    88          0181  1         obc_w_maxreclng = [32,0,16,0],      !Max rec length allowed by caller
;    89          0182  1         obc_b_flags =   [34,0,8,0],         !Flags
;    90          0183  1         obc_v_mhdseen = [34,0,1,0],         ! module header seen
;    91          0184  1         obc_v_lnmseen = [34,1,1,0],         ! lang. name record seen
;    92          0185  1         obc_v_1mod =    [34,2,1,0],         ! only process one module
;    93          0186  1         obc_b_currectyp = [35,0,8,0],       !Current record type
;    94          0187  1         obc_b_lstrectyp = [36,0,8,0],       !Last record type
;    95          0188  1         obc_b_modnamlng = [37,0,8,0],       !Length of module name
;    96          0189  1         obc_t_modname = [38,0,0,0]          !Length 31
;    97          0190  1         TES;
;    98          0191  1
;    99          0192  1 LITERAL
;   100          0193  1         obc_c_size = 38+31;                 !Size of OBC structure
;   101          0194  1
;   102          0195  1 GLOBAL LITERAL
;   103          0196  1         lib$m_lnk_1mod = 1;                 !Bit mask for flags
;   104          0197  1
;   105          0198  1 LINKAGE
;   106          0199  1     context_11 = CALL : GLOBAL (context = 11);
;   107          0200  1
;   108          0201  1 FORWARD ROUTINE
;   109          0202  1     dealloc_context : context_11,           !Deallocate context block
;   110          0203  1     prohdr : context_11,                    !Process module header records
;   111          0204  1     progsd : context_11,                    !Process GSD records
;   112          0205  1     proeom : context_11,                    !Process end of module records
;   113          0206  1     sequence_check : context_11;            !Check sequence of object records
;   114          0207  1
;   115          0208  1 EXTERNAL ROUTINE
;   116          0209  1     lib$free_vm,                            !Deallocate virtual memory
;   117          0210  1     lib$get_vm,                             !Allocate virtual memory
;   118          0211  1     str$free1_dx;                           !Deallocate dynamic string
;   119          0212  1
;   120          0213  1 EXTERNAL LITERAL
;   121          0214  1     lib$_badccc,                            !Illegal compilation completion code
```

```
;  122          0215  1      lib$_eomerror,                  !Errors in eom compilation code
;  123          0216  1      lib$_eomfatal,                  !Fatal errors in eom compilation code
;  124          0217  1      lib$_eomwarn,                   !Warnings in eom compilation code
;  125          0218  1      lib$_gsdtyp,                    !Illegal gsd type
;  126          0219  1      lib$_illfmlcnt,                 !Illegal formals count
;  127          0220  1      lib$_illmodnam,                 !Illegal module name length
;  128          0221  1      lib$_illpsclen,                 !Illegal psect length
;  129          0222  1      lib$_illreclen,                 !Illegal record length
;  130          0223  1      lib$_illrecln2,                 !Illegal record length
;  131          0224  1      lib$_illrectyp,                 !Illegal record type
;  132          0225  1      lib$_illrecty2,                 !Illegal record type
;  133          0226  1      lib$_illsymlen,                 !Illegal symbol length
;  134          0227  1      lib$_noeom,                     !No end of module record in file
;  135          0228  1      lib$_rectoosml,                 !Record too small to hold data
;  136          0229  1      lib$_sequence,                  !Illegal record sequence
;  137          0230  1      lib$_sequence2,                 !Illegal record sequence
;  138          0231  1      lib$_strlvl;                    !Illegal structure level
;  139          0232  1
;  140          0233  1 LITERAL
;  141          0234  1      true = 1,
;  142          0235  1      false = 0;
;  143          0236  1
;  144          0237  1 GLOBAL
;  145          0238  1      LIB$$gl_objctx : REF $BBLOCK FIELD(obc_fields);!pointer to context block
;  146          0239  1
;  147          0240  1 PSECT OWN = _LIB$CODE;                !Read-only data
;  148          0241  1
;  149          0242  1 OWN
;  150          0243  1      compilecodes : VECTOR[3,LONG]   !Translate eom compile codes into messages
;  151          0244  1                   INITIAL (lib$_eomwarn,
;  152          0245  1                            lib$_eomerror,
;  153          0246  1                            lib$_eomfatal);
;  154          0247  1 PSECT OWN = _LIB$DATA;
```

```
;  156          0248  1 %SBTTL  'lib$$getfilename - Get descriptor of file spec from FAB';
;  157          0249  1 GLOBAL ROUTINE lib$$getfilename (fab) =
;  158          0250  2 BEGIN
;  159          0251  2 !++
;  160          0252  2 ! FUNCTIONAL DESCRIPTION:
;  161          0253  2 !
;  162          0254  2 !       This routine returns a string descriptor for a file.
;  163          0255  2 !
;  164          0256  2 ! Inputs:
;  165          0257  2 !
;  166          0258  2 !       fab             Address of the fab
;  167          0259  2 !
;  168          0260  2 ! Outputs:
;  169          0261  2 !
;  170          0262  2 !       Routine value is address of string descriptor for file name
;  171          0263  2 !
;  172          0264  2 !--
;  173          0265  2
;  174          0266  2 MAP
;  175          0267  2     fab : REF $BBLOCK;
;  176          0268  2
;  177          0269  2 LOCAL
;  178          0270  2     nam : REF $BBLOCK;
;  179          0271  2
;  180          0272  2 OWN
;  181          0273  2     filedesc : $BBLOCK[dsc$c_s_bln];
;  182          0274  2
;  183          0275  2 nam = .fab[fab$l_nam];
;  184          0276  3 IF (.nam EQL 0)
;  185          0277  3     OR (IF (filedesc [dsc$w_length] = .nam [nam$b_rsl]) NEQ 0
;  186          0278  3             THEN filedesc [dsc$a_pointer] = .nam [nam$l_rsa]
;  187          0279  3             ELSE IF (filedesc [dsc$w_length] = .nam [nam$b_esl]) NEQ 0
;  188          0280  3                 THEN filedesc [dsc$a_pointer] = .nam [nam$l_esa];
;  189          0281  3         .filedesc[dsc$w_length] EQL 0)
;  190          0282  3 THEN BEGIN
;  191          0283  3     filedesc [dsc$w_length] = .fab [fab$b_fns]; !Use filename string
;  192          0284  3     filedesc [dsc$a_pointer] = .fab [fab$l_fna];! if all else fails
;  193          0285  2     END;
;  194          0286  2
;  195          0287  2 RETURN filedesc
;  196          0288  1 END;                                    !Of lib$$getfilename


                                                        .TITLE  LIB$$READ_OBJECT Read and dissect object file
                                                        .IDENT  \V03-004\

                                                        .PSECT  _LIB$DATA,NOEXE,  PIC,2

                                                00000 LIB$$GL_OBJCTX::
                                                        .BLKB   4
                                                00004 FILEDESC:
                                                        .BLKB   8

                                                        .PSECT  _LIB$CODE,NOWRT,  SHR,  PIC,2

                    00000000G 00000000G 00000000G 00000 COMPILECODES:
                                                        .LONG   LIB$_EOMWARN, LIB$_EOMERROR, LIB$_EOMFATAL  ;
```

```
                                                LIB$M_LNK_1MOD==    1
                                                        .EXTRN  LIB$FREE_VM, LIB$GET_VM
                                                        .EXTRN  STR$FREET_DX, LIB$_BADCCC
                                                        .EXTRN  LIB$_EOMERROR, LIB$_EOMFATAL
                                                        .EXTRN  LIB$_EOMWARN, LIB$_GSDTYP
                                                        .EXTRN  LIB$_ILLFMLCNT, LIB$_ILLMODNAM
                                                        .EXTRN  LIB$_ILLPSCLEN, LIB$_ILLRECLEN
                                                        .EXTRN  LIB$_ILLRECLN2, LIB$_ILLRECTYP
                                                        .EXTRN  LIB$_ILLRECTY2, LIB$_ILLSYMLEN
                                                        .EXTRN  LIB$_NOEOM, LIB$_RECTOOSML
                                                        .EXTRN  LIB$_SEQUENCE, LIB$_SEQUENCE2
                                                        .EXTRN  LIB$_STRLVL

                                  0004 00000            .ENTRY  LIB$$GETFILENAME, Save R2                       ; 0249
                52 00000000'  EF   9E 00002            MOVAB   FILEDESC, R2
                51         04  AC   D0 00009            MOVL    FAB, R1                                         ; 0275
                50         28  A1   D0 0000D            MOVL    40(R1), NAM
                               1C   13 00011            BEQL    3$                                              ; 0276
                62         03  A0   9B 00013            MOVZBW  3(NAM), FILEDESC                                ; 0277
                               07   13 00017            BEQL    1$
             04 A2         04  A0   D0 00019            MOVL    4(NAM), FILEDESC+4                              ; 0278
                               0B   11 0001E            BRB     2$
                62         0B  A0   9B 00020 1$:       MOVZBW  11(NAM), FILEDESC                               ; 0279
                               05   13 00024            BEQL    2$
             04 A2         0C  A0   D0 00026            MOVL    12(NAM), FILEDESC+4                             ; 0280
                               62   B5 0002B 2$:       TSTW    FILEDESC                                        ; 0281
                               09   12 0002D            BNEQ    4$
                62         34  A1   9B 0002F 3$:       MOVZBW  52(R1), FILEDESC                                ; 0283
             04 A2         2C  A1   D0 00033            MOVL    44(R1), FILEDESC+4                              ; 0284
                50             62   9E 00038 4$:       MOVAB   FILEDESC, R0                                    ; 0287
                                    04 0003B            RET                                                     ; 0288

; Routine Size:  60 bytes,    Routine Base:  _LIB$CODE + 000C
```

```
;  198          0289  1 %SBTTL  'lib$$report_io_error - Report I/O error on FAB or RAB';
;  199          0290  1 GLOBAL ROUTINE lib$$report_io_error (frab) =
;  200          0291  2 BEGIN
;  201          0292  2 !+++
;  202          0293  2 ! FUNCTIONAL DESCRIPTION:
;  203          0294  2 !
;  204          0295  2 !       This routine signals an I/O error.
;  205          0296  2 !
;  206          0297  2 ! Inputs:
;  207          0298  2 !
;  208          0299  2 !       frab            The FAB or the RAB which got the error
;  209          0300  2 !                       The $L_CTX field of the FAB/RAB must contain the
;  210          0301  2 !                       error code to signal
;  211          0302  2 !                        (SHR$_OPENIN/OPENOUT/READERR/WRITEERR/CLOSEIN/CLOSEOUT)
;  212          0303  2 !                       If frab is a RAB, then RAB$L_FAB must point to the FAB
;  213          0304  2 !                       In either case, the FAB must point to a valid NAM block
;  214          0305  2 !                        with both the expanded and resultant name strings in
;  215          0306  2 !                        order for consistent error reporting
;  216          0307  2 !
;  217          0308  2 ! Outputs:
;  218          0309  2 !
;  219          0310  2 !       The error is signalled.  RMS$_EOF is not signalled
;  220          0311  2 !
;  221          0312  2 ! Routine value:
;  222          0313  2 !
;  223          0314  2 !       The $L_STS field of frab is returned
;  224          0315  2 !
;  225          0316  2 !---
;  226          0317  2
;  227          0318  2 MAP
;  228          0319  2     frab : REF $BBLOCK;
;  229          0320  2
;  230          0321  2 IF .frab[rab$l_sts] NEQ rms$_eof
;  231          0322  2     THEN SIGNAL(.frab[fab$l_ctx],1,
;  232          0323  3         lib$$getfilename((IF .frab[fab$b_bid] EQL fab$c_bid
;  233          0324  3                          THEN .frab
;  234          0325  2                          ELSE .frab[rab$l_fab])),
;  235          0326  2         .frab[fab$l_sts],.frab[fab$l_stv]);
;  236          0327  2
;  237          0328  2 RETURN .frab[fab$l_sts]
;  238          0329  1 END;


                                        0004 00000          .ENTRY  LIB$$REPORT_IO_ERROR, Save R2                   ; 0290
                    52          04      AC   D0 00002       MOVL    FRAB, R2                                        ; 0321
        0001827A    8F          08      A2   D1 00006       CMPL    8(R2), #98938
                                        22   13 0000E       BEQL    3$
                    7E          08      A2   7D 00010       MOVQ    8(R2), -(SP)                                    ; 0326
                    03                  62   91 00014       CMPB    (R2), #3                                        ; 0323
                                        04   12 00017       BNEQ    1$
                                        52   DD 00019       PUSHL   R2                                              ; 0324
                                        03   11 0001B       BRB     2$
                                3C      A2   DD 0001D 1$:   PUSHL   60(R2)                                          ; 0325
                      A0        AF          01   FB 00020 2$:   CALLS   #1, LIB$$GETFILENAME                        ; 0323
```

```
                                    50  DD 00024          PUSHL   R0                                      ;
                                    01  DD 00026          PUSHL   #1                                      ; 0322
                                18  A2  DD 00028          PUSHL   24(R2)                                  ;
                    00000000G   00      05  FB 0002B      CALLS   #5, LIB$SIGNAL                          ;
                                50  08  A2  D0 00032 3$:   MOVL    8(R2), R0                               ; 0328
                                        04 00036          RET                                             ; 0329

; Routine Size:  55 bytes,    Routine Base:  _LIB$CODE + 0048
```

```
;  240          0330  1 %SBTTL  'dealloc_context -- deallocate context block';
;  241          0331  1 ROUTINE dealloc_context : context_11 =
;  242          0332  2 BEGIN
;  243          0333  2 !
;  244          0334  2 ! This routine deallocates the context block
;  245          0335  2 !
;  246          0336  2 EXTERNAL REGISTER
;  247          0337  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;  248          0338  2
;  249          0339  2 LOCAL
;  250          0340  2     status;
;  251          0341  2
;  252          0342  2 IF .context NEQ 0
;  253          0343  3 THEN BEGIN
;  254          0344  3     str$free1_dx(lib$$gl_objctx[obc_q_desc]);
;  255          0345  3     status = lib$free_vm(%REF(obc_c_size),lib$$gl_objctx);
;  256          0346  3     lib$$gl_objctx = context = 0;
;  257          0347  3     RETURN .status
;  258          0348  3     END
;  259          0349  2 ELSE RETURN true
;  260          0350  2
;  261          0351  1 END;


                                0004 00000 DEALLOC_CONTEXT:
                                           .WORD   Save R2                                                  ; 0331
                 52 00000000'   EF  9E 00002        MOVAB   LIB$$GL_OBJCTX, R2                               ;
                 5E             04  C2 00009        SUBL2   #4, SP                                           ;
                                5B  D5 0000C        TSTL    CONTEXT                                          ; 0342
                                21  13 0000E        BEQL    1$
                      7E        62  14 C1 00010     ADDL3   #20, LIB$$GL_OBJCTX, -(SP)                       ; 0344
                 00000000G      00  01 FB 00014     CALLS   #1, STR$FREE1_DX
                                    52 DD 0001B     PUSHL   R2                                               ; 0345
                           04   AE  45  8F 9A 0001D MOVZBL  #69, 4(SP)
                                04  AE  9F 00022     PUSHAB  4(SP)
                 00000000G      00  02 FB 00025     CALLS   #2, LIB$FREE_VM
                                    5B D4 0002C     CLRL    CONTEXT                                          ; 0346
                                    62 D4 0002E     CLRL    LIB$$GL_OBJCTX
                                       04 00030     RET                                                      ; 0349
                                50      01  D0 00031 1$:    MOVL    #1, R0
                                       04 00034     RET                                                      ; 0351

; Routine Size:  53 bytes,    Routine Base:  _LIB$CODE + 007F
```

```
;  263          0352  1 %SBTTL  'sequence_check -- check record type sequence';
;  264          0353  1 ROUTINE sequence_check : context_11 =
;  265          0354  2 BEGIN
;  266          0355  2 !
;  267          0356  2 ! Check that the record sequence is correct
;  268          0357  2 !
;  269          0358  2 ROUTINE sequence_error : context_11 =
;  270          0359  3 BEGIN
;  271          0360  3 !
;  272          0361  3 ! Signal a record sequence error
;  273          0362  3 !
;  274          0363  3 EXTERNAL REGISTER
;  275          0364  3     context = 11 : REF $BBLOCK FIELD(obc_fields);
;  276          0365  3
;  277          0366  3 IF .context[obc_b_modnamlng] NEQ 0
;  278          0367  3     THEN SIGNAL(lib$_sequence,1,context[obc_b_modnamlng])
;  279          0368  3     ELSE SIGNAL(lib$_sequence2);
;  280          0369  3
;  281          0370  3 RETURN lib$_sequence
;  282          0371  2 END;


                                000C 00000 SEQUENCE_ERROR:
                                           .WORD   Save R2,R3                                          ; 0358
                    53 00000000G   8F  D0 00002    MOVL    #LIB$_SEQUENCE, R3                          :
                    52 00000000G   00  9E 00009    MOVAB   LIB$SIGNAL, R2                              :
                               25  AB  95 00010    TSTB    37(CONTEXT)                                 ; 0366
                                   0C  13 00013    BEQL    1$                                          :
                               25  AB  9F 00015    PUSHAB  37(CONTEXT)                                 ; 0367
                                   01  DD 00018    PUSHL   #1                                          :
                                   53  DD 0001A    PUSHL   R3                                          :
                    62                 03  FB 0001C    CALLS   #3, LIB$SIGNAL                          :
                                       09  11 0001F    BRB     2$                                      :
                        00000000G  8F  DD 00021 1$: PUSHL   #LIB$_SEQUENCE2                            ; 0368
                    62             01  FB 00027    CALLS   #1, LIB$SIGNAL                              :
                    50             53  D0 0002A 2$: MOVL    R3, R0                                     ; 0370
                                       04  0002D    RET                                                ; 0371

; Routine Size:  46 bytes,    Routine Base:  _LIB$CODE + 00B4

;  283          0372  2 !
;  284          0373  2 ! Main body of sequence_check
;  285          0374  2 !
;  286          0375  2 EXTERNAL REGISTER
;  287          0376  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;  288          0377  2
;  289          0378  2 BIND
;  290          0379  2     recdesc = context[obc_q_desc] : $BBLOCK,
;  291          0380  2     objrec = .recdesc[dsc$a_pointer] : $BBLOCK;
;  292          0381  2
;  293          0382  2 IF .context[obc_b_currectyp] EQL obj$c_hdr
;  294          0383  3     THEN BEGIN
;  295          0384  3         IF .objrec[obj$b_subtyp] EQL obj$c_hdr_mhd
```

```
;  296          0385  4          THEN BEGIN
;  297          0386  4              IF .context[obc_b_lstrectyp] EQL obj$c_eom
;  298          0387  5              THEN BEGIN
;  299          0388  5                  context[obc_v_mhdseen] = true;                  !Main mhd record has just followed eom recor
;  300          0389  5                  context[obc_v_lnmseen] = false;                 !Flag no lnm mhd seen
;  301          0390  5                  RETURN true
;  302          0391  5                  END
;  303          0392  4              ELSE RETURN sequence_error()                        !Last record was not eom, signal the error
;  304          0393  4              END
;  305          0394  3          ELSE IF .context[obc_v_mhdseen]
;  306          0395  4              THEN BEGIN
;  307          0396  4                  IF .objrec[obj$b_subtyp] EQL obj$c_hdr_lnm
;  308          0397  4                      THEN context[obc_v_lnmseen] = true;
;  309          0398  4                  RETURN true
;  310          0399  4                  END
;  311          0400  3              ELSE RETURN sequence_error()
;  312          0401  3              END
;  313          0402  2      ELSE IF .context[obc_v_mhdseen]                                                                  ; Re
;  314          0403  2                  AND .context[obc_v_lnmseen]
;  315          0404  3              THEN BEGIN
;  316          0405  3                  IF .context[obc_b_currectyp] EQL obj$c_eom      !If current record is end of module
;  317          0406  3                      THEN context[obc_v_mhdseen] = false;        ! then we have no mhd record
;  318          0407  3                  RETURN true
;  319          0408  3                  END
;  320          0409  2              ELSE RETURN sequence_error();
;  321          0410  2
;  322          0411  1 END;


                                0000 00000 SEQUENCE_CHECK:
                                           .WORD   Save nothing                                                         ; 0353
                     50    14   AB  9E 00002       MOVAB   20(CONTEXT), R0                                           ; 0379
                     50    04   A0  D0 00006       MOVL    4(R0), R0                                                 ; 0380
                           23   AB  95 0000A       TSTB    35(CONTEXT)                                               ; 0382
                                25  12 0000D       BNEQ    2$                                                        ;
                           01   A0  95 0000F       TSTB    1(R0)                                                     ; 0384
                                10  12 00012       BNEQ    1$                                                        ;
                     03    24   AB  91 00014       CMPB    36(CONTEXT), #3                                           ; 0386
                                31  12 00018       BNEQ    4$                                                        ;
                22   AB         01  88 0001A       BISB2   #1, 34(CONTEXT)                                           ; 0388
                22   AB         02  8A 0001E       BICB2   #2, 34(CONTEXT)                                           ; 0389
                                23  11 00022       BRB     3$                                                        ; 0390
                     23    22   AB  E9 00024 1$:    BLBC    34(CONTEXT), 4$                                           ; 0394
                     01    01   A0  91 00028       CMPB    1(R0), #1                                                 ; 0396
                                19  12 0002C       BNEQ    3$                                                        ;
                22   AB         02  88 0002E       BISB2   #2, 34(CONTEXT)                                           ; 0397
                                13  11 00032       BRB     3$                                                        ; 0398
                     13    22   AB  E9 00034 2$:    BLBC    34(CONTEXT), 4$                                           ; 0402
0E              22   AB         01  E1 00038       BBC     #1, 34(CONTEXT), 4$                                       ; 0403
                     03    23   AB  91 0003D       CMPB    35(CONTEXT), #3                                           ; 0405
                                04  12 00041       BNEQ    3$                                                        ;
                22   AB         01  8A 00043       BICB2   #1, 34(CONTEXT)                                           ; 0406
                     50         01  D0 00047 3$:    MOVL    #1, R0                                                    ; 0407
                                04  0004A         RET                                                               ;
```

```
                        83   AF          00  FB 0004B 48:    CALLS   #0, SEQUENCE_ERROR                    ; 0409
                                             04 0004F        RET                                          ; 0411

; Routine Size:  80 bytes,    Routine Base:  _LIB$CODE + 00E2
```

```
;  324          0412  1 %SBTTL  'prohdr -- process MHD records';
;  325          0413  1 ROUTINE prohdr : context_11 =
;  326          0414  2 BEGIN
;  327          0415  2 !
;  328          0416  2 ! This routine processes MHD records
;  329          0417  2 !
;  330          0418  2 ! Inputs:
;  331          0419  2 !
;  332          0420  2 !       recdesc         Address of string descriptor for mhd record
;  333          0421  2 !
;  334          0422  2
;  335          0423  2 EXTERNAL REGISTER
;  336          0424  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;  337          0425  2
;  338          0426  2 BIND
;  339          0427  2     recdesc = context[obc_q_desc] : $BBLOCK,
;  340          0428  2     objrec = .recdesc[dsc$a_pointer] : $BBLOCK;
;  341          0429  2
;  342          0430  2 LOCAL
;  343          0431  2     status;
;  344          0432  2
;  345          0433  2 !
;  346          0434  2 ! Check record sequence
;  347          0435  2 !
;  348          0436  3 IF NOT (status = sequence_check())
;  349          0437  2     THEN RETURN .status;
;  350          0438  2 !
;  351          0439  2 ! Skip all but main module header records
;  352          0440  2 !
;  353          0441  2 IF .objrec[obj$b_subtyp] NEQ obj$c_hdr_mhd
;  354          0442  2     THEN RETURN true;
;  355          0443  2 !
;  356          0444  2 ! Check for legal structure level
;  357          0445  2 !
;  358          0446  2 IF .objrec[mhd$b_strlvl] GTRU obj$c_strlvl
;  359          0447  3 THEN BEGIN
;  360          0448  3     SIGNAL(lib$_strlvl,1,objrec[mhd$b_namlng]);
;  361          0449  3     RETURN lib$_strlvl
;  362          0450  2     END;
;  363          0451  2 !
;  364          0452  2 ! Check max record length supplied
;  365          0453  2 !
;  366          0454  2 IF (context[obc_w_maxreclng] = .objrec[mhd$w_recsiz]) GTRU obj$c_maxrecsiz
;  367          0455  3 THEN BEGIN
;  368          0456  3     SIGNAL(lib$_illreclen,2,.objrec[mhd$w_recsiz],objrec[mhd$b_namlng]);
;  369          0457  3     RETURN lib$_illreclen
;  370          0458  2     END;
;  371          0459  2 !
;  372          0460  2 ! Check module name length
;  373          0461  2 !
;  374          0462  2 IF .objrec[mhd$b_namlng] GTRU obj$c_symsiz
;  375          0463  2     OR .objrec[mhd$b_namlng] EQL 0
;  376          0464  3 THEN BEGIN
;  377          0465  3     SIGNAL(lib$_illmodnam,.objrec[mhd$b_namlng],objrec[mhd$b_namlng]);
;  378          0466  3     RETURN lib$_illmodnam
;  379          0467  2     END;
;  380          0468  2 !
```

```
;  381          0469  2 ! Copy module name into context block for error messages
;  382          0470  2 !
;  383          0471  2 context[obc_b_modnamlng] = .objrec[mhd$b_namlng];
;  384          0472  2 CH$MOVE(.objrec[mhd$b_namlng],objrec[mhd$t_name],context[obc_t_modname]);
;  385          0473  2 !
;  386          0474  2 ! Call user action routine for "other records" if specified
;  387          0475  2 !
;  388          0476  2 IF .context[obc_l_orcrtn] NEQ 0
;  389          0477  2     THEN status = (.context[obc_l_orcrtn])(recdesc,.context[obc_l_usrdata])
;  390          0478  2     ELSE status = true;
;  391          0479  2
;  392          0480  2 RETURN .status
;  393          0481  1 END;


                                    07FC 00000 PROHDR: .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10          ; 0413
                5A 00000000G   8F   D0 00002          MOVL   #LIB$_ILLRECLEN, R10                       ;
                59 00000000G   8F   D0 00009          MOVL   #LIB$_STRLVL, R9                           ;
                58 00000000G   00   9E 00010          MOVAB  LIB$SIGNAL, R8                             ;
                56       14    AB   9E 00017          MOVAB  20(CONTEXT), R6                            ; 0427
                52       04    A6   D0 0001B          MOVL   4(R6), R2                                  ; 0428
             8D AF               00   FB 0001F          CALLS  #0, SEQUENCE_CHECK                         ; 0436
                57               50   D0 00023          MOVL   R0, STATUS                                 ;
                03               57   E8 00026          BLBS   STATUS, 1$                                 ;
                              0085   31 00029          BRW    8$                                         ;
                         01    A2   95 0002C 1$:      TSTB   1(R2)                                      ; 0441
                               04   13 0002F          BEQL   2$                                         ;
                50             01   D0 00031          MOVL   #1, R0                                     ; 0442
                               04   00034             RET                                               ;
                         02    A2   95 00035 2$:      TSTB   2(R2)                                      ; 0446
                               0E   13 00038          BEQL   3$                                         ;
                         05    A2   9F 0003A          PUSHAB 5(R2)                                      ; 0448
                               01   DD 0003D          PUSHL  #1                                         ;
                               59   DD 0003F          PUSHL  R9                                         ;
                68             03   FB 00041          CALLS  #3, LIB$SIGNAL                             ;
                50             59   D0 00044          MOVL   R9, R0                                     ; 0449
                               04   00047             RET                                               ;
                50       03    A2   3C 00048 3$:      MOVZWL 3(R2), R0                                  ; 0454
             20 AB               50   B0 0004C          MOVW   R0, 32(CONTEXT)                            ;
           0800 8F               50   B1 00050          CMPW   R0, #2048                                  ;
                               12   1B 00055          BLEQU  4$                                         ;
                         05    A2   9F 00057          PUSHAB 5(R2)                                      ; 0456
                7E       03    A2   3C 0005A          MOVZWL 3(R2), -(SP)                               ;
                               02   DD 0005E          PUSHL  #                                          ;
                               5A   DD 00060          PUSHL  R10                                        ;
                68             04   FB 00062          CALLS  #4, LIB$SIGNAL                             ;
                50             5A   D0 00065          MOVL   R10, R0                                    ; 0457
                               04   00068             RET                                               ;
                1F       05    A2   91 00069 4$:      CMPB   5(R2), #31                                 ; 0462
                               05   1A 0006D          BGTRU  5$                                         ;
                         05    A2   95 0006F          TSTB   5(R2)                                      ; 0463
                               18   12 00072          BNEQ   6$                                         ;
                         05    A2   9F 00074 5$:      PUSHAB 5(R2)                                      ; 0465
                7E       05    A2   9A 00077          MOVZBL 5(R2), -(SP)                               ;
```

```
                                00000000G   8F   DD 0007B          PUSHL   #LIB$_ILLMODNAM                          ;
                              68            03   FB 00081          CALLS   #3, LIB$SIGNAL                          ;
                              50 00000000G  8F   D0 00084          MOVL    #LIB$_ILLMODNAM, R0                     ; 0466
                                            04   04 0008B          RET                                             ;
                          25  AB      05    A2   90 0008C 6$:      MOVB    5(R2), 37(CONTEXT)                      ; 0471
                          50      05        A2   9A 00091          MOVZBL  5(R2), R0                               ; 0472
                      26  AB      06        A2   50 28 00095       MOVC3   R0, 6(R2), 38(CONTEXT)                  ;
                                        10  AB   D5 0009B          TSTL    16(CONTEXT)                             ; 0476
                                            0E   13 0009E          BEQL    7$                                      ;
                                        1C  AB   DD 000A0          PUSHL   28(CONTEXT)                             ; 0477
                                            56   DD 000A3          PUSHL   R6                                      ;
                          10  BB            02   FB 000A5          CALLS   #2, @16(CONTEXT)                        ;
                              57            50   D0 000A9          MOVL    R0, STATUS                              ;
                                            03   11 000AC          BRB     8$                                      ;
                              57            01   D0 000AE 7$:      MOVL    #1, STATUS                              ; 0478
                              50            57   D0 000B1 8$:      MOVL    STATUS, R0                              ; 0480
                                            04   04 000B4          RET                                             ; 0481

; Routine Size:  181 bytes,    Routine Base:  _LIB$CODE + 0132
```

```
;  395          0482  1 %SBTTL  'progsd -- process GSD records';
;  396          0483  1 ROUTINE progsd : context_11 =
;  397          0484  2 BEGIN
;  398          0485  2 !
;  399          0486  2 ! This routine processes GSD records
;  400          0487  2 !
;  401          0488  2 ! Inputs:
;  402          0489  2 !
;  403          0490  2 !       recdesc         Address of string descriptor for gsd record
;  404          0491  2 !
;  405          0492  2 !
;  406          0493  2 BUILTIN
;  407          0494  2     NULLPARAMETER;
;  408          0495  2
;  409          0496  2 EXTERNAL REGISTER
;  410          0497  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;  411          0498  2
;  412          0499  2 LOCAL
;  413          0500  2     symboldesc : $BBLOCK[dsc$c_s_bln],              !String descriptor for symbol name
;  414          0501  2     symbolvalue,                                    !Value of symbol
;  415          0502  2     symbolflags,                                    !Symbol flags
;  416          0503  2     gsd_desc : $BBLOCK[dsc$c_s_bln],                !String descriptor for gsd subrecord
;  417          0504  2     status,                                         !Status from processing entry point
;  418          0505  2     length,                                         !Length of def/ref
;  419          0506  2     gsdoffset,                                      !Offset into record
;  420          0507  2     objrec : REF $BBLOCK;                           !pointer to object record
;  421          0508  2
;  422          0509  2 BIND
;  423          0510  2     recdesc = context[obc_q_desc] : $BBLOCK,
;  424          0511  2     objvec = .recdesc[dsc$a_pointer] : VECTOR[,BYTE];   !Name record as byte vector
;  425          0512  2
;  426          0513  2 IF .context[obc_l_gblrtn] EQL 0                      !If no routine to process them
;  427          0514  2     THEN RETURN true;                               ! then don't bother with the record
;  428          0515  2
;  429          0516  2 gsd_desc[dsc$b_dtype] = gsd_desc[dsc$b_class] = 0;
;  430          0517  2 gsdoffset = obj$c_subtyp;                           !Init pointer into record
;  431          0518  2
;  432          0519  2 !
;  433          0520  2 ! Process the GSD record
;  434          0521  2 !
;  435          0522  2 WHILE .gsdoffset LSSU .recdesc[dsc$w_length]         !Loop through the record
;  436          0523  3 DO BEGIN
;  437          0524  3     LOCAL
;  438          0525  3         recordtype,
;  439          0526  3         wordpsectgsd;                               !Contains word of psect rather than byte
;  440          0527  3
;  441          0528  3     objrec = .recdesc[dsc$a_pointer] + .gsdoffset;  !Update record pointer
;  442          0529  5     wordpsectgsd = ((.objrec[gsd$b_gsdtyp] GEQU gsd$c_symw) !Test for word of psect number
;  443          0530  3                                 AND (.objrec[gsd$b_gsdtyp] LEQU gsd$c_prow));
;  444          0531  3
;  445          0532  4     CASE (recordtype = .objvec[.gsdoffset])         !Dispatch to process GSD
;  446          0533  3                          FROM gsd$c_psc TO gsd$c_maxrectyp OF
;  447          0534  3     SET
```

```
;  449     0535  3     [gsd$c_psc,gsd$c_spsc] :                        !Psect definition
;  450     0536  3     !
;  451     0537  3     ! PSECT definitions
;  452     0538  3     !
;  453     0539  4         BEGIN
;  454     0540  4         BIND
;  455     0541  4             psectdef = objvec[.gsdoffset] : $BBLOCK;        !Name the definition
;  456     0542  4
;  457     0543  4         LOCAL
;  458     0544  4             psectdesc : $BBLOCK[dsc$c_s_bln],
;  459     0545  4             psectalign,
;  460     0546  4             psectflags,
;  461     0547  4             psectalloc;
;  462     0548  4
;  463     0549  4         IF (.gsdoffset + gps$c_name + 1) GEQU .recdesc[dsc$w_length]
;  464     0550  5         THEN BEGIN
;  465     0551  5             SIGNAL(lib$_rectoosml,1,context[obc_b_modnamlng]);
;  466     0552  5             RETURN lib$_rectoosml
;  467     0553  4             END;
;  468     0554  4         psectdesc[dsc$b_dtype] = psectdesc[dsc$b_class] = 0;
;  469     0555  4         IF .recordtype EQL gsd$c_psc
;  470     0556  5         THEN BEGIN
;  471     0557  5             psectdesc[dsc$w_length] = .psectdef[gps$b_namlng];
;  472     0558  5             psectdesc[dsc$a_pointer] = psectdef[gps$t_name];
;  473     0559  5             length = gps$c_name + .psectdesc[dsc$w_length]; !Compute length of psect def.
;  474     0560  5             END
;  475     0561  5         ELSE BEGIN
;  476     0562  5             psectdesc[dsc$w_length] = .psectdef[sgps$b_namlng];
;  477     0563  5             psectdesc[dsc$a_pointer] = psectdef[sgps$t_name];
;  478     0564  5             length = sgps$c_name + .psectdesc[dsc$w_length];
;  479     0565  4             END;
;  480     0566  4         IF .psectdesc[dsc$w_length] EQL 0                   !Check length of psect name
;  481     0567  4             OR .psectdesc[dsc$w_length] GTRU obj$c_symsiz
;  482     0568  5         THEN BEGIN
;  483     0569  6             SIGNAL(lib$_illpsclen,3,(IF .recordtype EQL gsd$c_psc
;  484     0570  6                                      THEN psectdef[gps$b_namlng]
;  485     0571  5                                      ELSE psectdef[sgps$b_namlng]),
;  486     0572  5                    .psectdesc[dsc$w_length],context[obc_b_modnamlng]);
;  487     0573  5             RETURN lib$_illpsclen
;  488     0574  4             END;
;  489     0575  4         IF .context[obc_l_pscrtn] NEQ 0                     !If user psect routine supplied
;  490     0576  5         THEN BEGIN                                          ! then set up and call it now
;  491     0577  5             psectalign = .psectdef[gps$b_align];
;  492     0578  5             psectflags = .psectdef[gps$w_flags];
;  493     0579  5             psectalloc = .psectdef[gps$l_alloc];
;  494     0580  5             gsd_desc[dsc$w_length] = .length;               !Set up descriptor for psect def.
;  495     0581  5             gsd_desc[dsc$a_pointer] = .objrec;
;  496     0582  5             (.context[obc_l_pscrtn])(psectdesc,             !Call the user routine now
;  497     0583  5                                      psectalign,psectflags,psectalloc,
;  498     0584  5                                      .context[obc_l_usrdata],gsd_desc);
;  499     0585  4             END;
;  500     0586  4         gsdoffset = .gsdoffset + .length;                   !Update pointer into record
;  501     0587  3         END;
```

```
;  503     0588  3     !
;  504     0589  3     ! All types of symbols
;  505     0590  3     !
;  506     0591  3     [gsd$c_sym TO gsd$c_prow] :                                     !All symbols
;  507     0592  4         BEGIN
;  508     0593  4         BIND
;  509     0594  4             symbolrec = objvec[.gsdoffset] : $BBLOCK;               !Name the symbol gsd
;  510     0595  4
;  511     0596  4         LOCAL
;  512     0597  4             entrymask,
;  513     0598  4             symbolstring : REF VECTOR[,BYTE];                       !Pointer to symbol ascic name
;  514     0599  4
;  515     0600  4         IF .recordtype EQL gsd$c_epm                                !Process entry points and procedures
;  516     0601  4             OR .recordtype EQL gsd$c_epmw
;  517     0602  4             OR .recordtype EQL gsd$c_pro
;  518     0603  4             OR .recordtype EQL gsd$c_prow
;  519     0604  5         THEN BEGIN
;  520     0605  5              !
;  521     0606  5              ! Process entry points and procedure definitions
;  522     0607  5              !
;  523     0608  5              IF .wordpsectgsd
;  524     0609  6              THEN BEGIN
;  525     0610  6                   !
;  526     0611  6                   ! Entry point with word of psect
;  527     0612  6                   !
;  528     0613  6                   entrymask = .symbolrec[epmw$w_mask];
;  529     0614  6                   length = epmw$c_name + .symbolrec[epmw$b_namlng];
;  530     0615  6                   symbolvalue = .symbolrec[epmw$l_addrs];
;  531     0616  6                   symbolstring = symbolrec[epmw$b_namlng];
;  532     0617  6                   END
;  533     0618  6              ELSE BEGIN
;  534     0619  6                   !
;  535     0620  6                   ! Entry point with byte of psect
;  536     0621  6                   !
;  537     0622  6                   entrymask = .symbolrec[epm$w_mask];
;  538     0623  6                   length = epm$c_name + .symbolrec[epm$b_namlng];
;  539     0624  6                   symbolvalue = .symbolrec[epm$l_addrs];
;  540     0625  6                   symbolstring = symbolrec[epm$b_namlng];
;  541     0626  5                   END;
;  542     0627  5              !
;  543     0628  5              ! If this is procedure definition, then skip the argument
;  544     0629  5              !  descriptors
;  545     0630  5              !
;  546     0631  5              IF .recordtype EQL gsd$c_pro
;  547     0632  5                  OR .recordtype EQL gsd$c_prow
;  548     0633  6              THEN BEGIN
;  549     0634  6                   BIND
;  550     0635  6                       formals = objvec[.gsdoffset+.length] : $BBLOCK; !Name formal argument descriptors
;  551     0636  6
;  552     0637  6                   LOCAL
;  553     0638  6                       argcount;
;  554     0639  6
;  555     0640  6                   IF .formals[fml$b_minargs] GTRU .formals[fml$b_maxargs]
;  556     0641  7                   THEN BEGIN
;  557     0642  7                       SIGNAL(lib$_illfmlcnt,2,.symbolstring,context[obc_b_modnamlng]);
;  558     0643  7                       RETURN lib$_illfmlcnt
;  559     0644  6                       END;
```

```
;  560          0645  6                    IF (.gsdoffset + .length + fml$c_size) GEQU .recdesc[dsc$w_length]
;  561          0646  7                    THEN BEGIN
;  562          0647  7                        SIGNAL(lib$_rectoosml,1,context[obc_b_modnamlng]);
;  563          0648  7                        RETURN lib$_rectoosml
;  564          0649  6                        END;
;  565          0650  6                    length = .length + fml$c_size;                      !Skip fixed part of formals
;  566          0651  6                    IF (argcount = .formals[fml$b_maxargs]) NEQ 0       !If there are argument descriptors
;  567          0652  6                    THEN INCR i FROM 1 TO .argcount                     ! then process them
;  568          0653  7                    DO BEGIN
;  569          0654  7                        BIND
;  570          0655  7                            argdesc = objvec[.gsdoffset+.length] :      !Name the argument descriptor
;  571          0656  7                                                  $BBLOCK;
;  572          0657  7
;  573          0658  7                        length = .length + .argdesc[arg$b_bytecnt] + arg$c_size;
;  574          0659  6                        END;
;  575          0660  5                    END;
;  576          0661  4                END;
;  577          0662  4
;  578          0663  4            !
;  579          0664  4            ! Process ordinary symbol definitions and references
;  580          0665  4            !
;  581          0666  4            IF .recordtype EQL gsd$c_sym
;  582          0667  5                OR .recordtype EQL gsd$c_symw
;  583          0668  5            THEN BEGIN
;  584          0669  5                !
;  585          0670  5                ! Ordinary symbol definitions and references
;  586          0671  5                !
;  587          0672  5                entrymask = 0;                                          !No entry mask
;  588          0673  6                IF NOT .symbolrec[gsy$v_def]                            !If a reference
;  589          0674  6                THEN BEGIN
;  590          0675  6                    !
;  591          0676  6                    ! Symbol reference
;  592          0677  6                    !
;  593          0678  6                    length = srf$c_name + .symbolrec[srf$b_namlng];     !Simply compute length of ref
;  594          0679  6                    symbolvalue = 0;                                    !Value is 0 if a reference
;  595          0680  6                    symbolstring = symbolrec[srf$b_namlng];
;  596          0681  6                    END
;  597          0682  6                ELSE BEGIN
;  598          0683  6                    !
;  599          0684  6                    ! Symbol definition
;  600          0685  6                    !
;  601          0686  7                    IF .wordpsectgsd                                    !If a word of psect number
;  602          0687  7                    THEN BEGIN
;  603          0688  7                        !
;  604          0689  7                        ! ...with word of psect number
;  605          0690  7                        !
;  606          0691  7                        length = sdfw$c_name + .symbolrec[sdfw$b_namlng];
;  607          0692  7                        symbolvalue = .symbolrec[sdfw$l_value];         !Point to value
;  608          0693  7                        symbolstring = symbolrec[sdfw$b_namlng];        !Point to the symbol name
;  609          0694  7                        END
;  610          0695  7                    ELSE BEGIN
;  611          0696  7                        !
;  612          0697  7                        ! ...with byte of psect number
;  613          0698  7                        !
;  614          0699  7                        length = sdf$c_name + .symbolrec[sdf$b_namlng];
;  615          0700  7                        symbolvalue = .symbolrec[sdf$l_value];          !Point to symbol value
;  616          0701  6                        symbolstring = symbolrec[sdf$b_namlng];         !Point to the symbol name
                                                END;
```

```
;  617          0702  5                        END;
;  618          0703  4                    END;                                        !Symbol definition
;  619          0704  4                !
;  620          0705  4                ! Check length of symbol name
;  621          0706  4                !
;  622          0707  4                IF .symbolstring[0] EQL 0                      !Check validity of symbol name
;  623          0708  4                    OR .symbolstring[0] GTRU obj$c_symsiz
;  624          0709  5                THEN BEGIN
;  625          0710  5                    SIGNAL(lib$_illsymlen,3,.symbolstring,     !Signal illegal symbol name
;  626          0711  5                           .symbolstring[0],context[obc_b_modnamlng]);
;  627          0712  5                    RETURN lib$_illsymlen
;  628          0713  4                    END;
;  629          0714  4                !
;  630          0715  4                ! Create string descriptor for symbol name
;  631          0716  4                !
;  632          0717  4                symbolflags = .symbolrec[sdf$w_flags];         !Get the symbol flags
;  633          0718  4                symboldesc[dsc$w_length] = .symbolstring[0];
;  634          0719  4                symboldesc[dsc$b_dtype] = 0;
;  635          0720  4                symboldesc[dsc$b_class] = 0;
;  636          0721  4                symboldesc[dsc$a_pointer] = symbolstring[1];
;  637          0722  4                gsd_desc[dsc$w_length] = .length;
;  638          0723  4                gsd_desc[dsc$a_pointer] = .objrec;
;  639          0724  4
;  640          0725  5                (.context[obc_l_gblrtn])                       !Call the user global symbol routine
;  641          0726  4                    (symboldesc,symbolvalue,symbolflags,entrymask,
;  642          0727  4                                .context[obc_l_usrdata],gsd_desc);
;  643          0728  4                gsdoffset = .gsdoffset + .length;              !Update the pointer into the record
;  644          0729  3                END;
;  645          0730  3
;  646          0731  3            [gsd$c_idc] :                                      !Entity ident check
;  647          0732  4                BEGIN
;  648          0733  4 !                  BIND
;  649          0734  4 !                      entity_name = ,
;  650          0735  4 !                      entity_ident =,
;  651          0736  4 !                      object_name =;
;  652          0737  4
;  653          0738  4                    true
;  654          0739  4
;  655          0740  3                    END;
;  656          0741  3            [INRANGE] :
;  657          0742  4                BEGIN
;  658          0743  4                    true
;  659          0744  3                    END;
;  660          0745  3            TES;
;  661          0746  2            END;                                               !GSD record
;  662          0747  2
;  663          0748  2 RETURN true
;  664          0749  2
;  665          0750  1 END;                                                          !of progsd



                                        07FC 00000 PROGSD: .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10          ; 0483
                              5E        34   C2 00002      SUBL2  #52, SP                                    ;
                              55    14  AB   9E 00005      MOVAB  20(CONTEXT), R5                            ; 0510
```

```
                                  6B  D5 00009      TSTL    (CONTEXT)                        ; 0513
                                  0D  13 0000B      BEQL    2$                               ;
                              26  AE  B4 0000D      CLRW    GSD_DESC+2                       ; 0516
                     52           01  D0 00010      MOVL    #1, GSDOFFSET                    ; 0517
  52        65       10           00  ED 00013 1$:  CMPZV   #0, #16, (R5), GSDOFFSET         ; 0522
                                  03  1A 00018      BGTRU   3$                               ;
                                0251  31 0001A 2$:  BRW     35$                              ;
            5A       52       04  A5  C1 0001D 3$:  ADDL3   4(R5), GSDOFFSET, OBJREC         ; 0528
                                  51  D4 00022      CLRL    R1                               ; 0529
                     04       01  AA  91 00024      CMPB    1(OBJREC), #4                    ;
                                  02  1F 00028      BLSSU   4$                               ;
                                  51  D6 0002A      INCL    R1                               ;
                                  50  D4 0002C 4$:  CLRL    R0                               ; 0530
                     06       01  AA  91 0002E      CMPB    1(OBJREC), #6                    ;
                                  02  1A 00032      BGTRU   5$                               ;
                                  50  D6 00034      INCL    R0                               ;
                     53           51  D2 00036 5$:  MCOML   R1, R3                           ;
            6E       50           53  CB 00039      BICL3   R3, R0, WORDPSECTGSD             ;
            53       52       04  A5  C1 0003D      ADDL3   4(R5), GSDOFFSET, R3             ; 0532
                     58           63  9A 00042      MOVZBL  (R3), RECORDTYPE                 ;
            0C       00           58  CF 00045      CASEL   RECORDTYPE, #0, #12              ;
  00C2      00C2     00C2       001A     00049 6$:  .WORD   7$-6$,-                          ;
  FFCA      00C2     00C2       00C2     00051              16$-6$,-                         ;
  FFCA      FFCA     FFCA       FFCA     00059              16$-6$,-                         ;
                                001A     00061              16$-6$,-                         ;
                                                            16$-6$,-                         ;
                                                            16$-6$,-                         ;
                                                            16$-6$,-                         ;
                                                            1$-6$,-                          ;
                                                            1$-6$,-                          ;
                                                            1$-6$,-                          ;
                                                            1$-6$,-                          ;
                                                            1$-6$,-                          ;
                                                            7$-6$                            ;
                     50       0A  A2  9E 00063 7$:  MOVAB   10(R2), R0                       ; 0549
  50        65       10           00  ED 00067      CMPZV   #0, #16, (R5), R0                ;
                                  03  1A 0006C      BGTRU   8$                               ;
                                011D  31 0006E      BRW     23$                              ;
                              1E  AE  B4 00071 8$:  CLRW    PSECTDESC+2                      ; 0554
                                  51  D4 00074      CLRL    R1                               ; 0555
                                  58  D5 00076      TSTL    RECORDTYPE                       ;
                                  15  12 00078      BNEQ    9$                               ;
                                  51  D6 0007A      INCL    R1                               ;
                 1C  AE       08  A3  9B 0007C      MOVZBW  8(R3), PSECTDESC                 ; 0557
                 20  AE       09  A3  9E 00081      MOVAB   9(R3), PSECTDESC+4               ; 0558
                     57       1C  AE  3C 00086      MOVZWL  PSECTDESC, LENGTH                ; 0559
                     57           09  C0 0008A      ADDL2   #9, LENGTH                       ;
                                  11  11 0008D      BRB     10$                              ; 0555
                 1C  AE       0C  A3  9B 0008F 9$:  MOVZBW  12(R3), PSECTDESC                ; 0562
                 20  AE       0D  A3  9E 00094      MOVAB   13(R3), PSECTDESC+4              ; 0563
                     57       1C  AE  3C 00099      MOVZWL  PSECTDESC, LENGTH                ; 0564
                     57           0D  C0 0009D      ADDL2   #13, LENGTH                      ;
                     50       1C  AE  3C 000A0 10$: MOVZWL  PSECTDESC, R0                    ; 0566
                                  05  13 000A4      BEQL    11$                              ;
                     1F           50  B1 000A6      CMPW    R0, #31                          ; 0567
                                  2B  1B 000A9      BLEQU   14$                              ;
                              25  AB  9F 000AB 11$: PUSHAB  37(CONTEXT)                      ; 0572
```

```
                                                50  DD 000AE          PUSHL   R0
                                          06    51  E9 000B0          BLBC    R1, 12$
                                          50    08  A3 9E 000B3       MOVAB   8(R3), R0                       ;0570
                                                04  11 000B7          BRB     13$
                                          50    0C  A3 9E 000B9 12$:  MOVAB   12(R3), R0                      ;0571
                                                50  DD 000BD 13$:     PUSHL   R0
                                                03  DD 000BF          PUSHL   #3                              ;0572
                                      00000000G 8F  DD 000C1          PUSHL   #LIB$_ILLPSCLEN
                                00000000G 00    05  FB 000C7          CALLS   #5, LIB$SIGNAL
                                          50 C0000000G 8F DO 000CE    MOVL    #LIB$_ILLPSCLEN, R0             ;0573
                                                04  000D5          RET
                                          04    AB  D5 000D6 14$:     TSTL    4(CONTEXT)                      ;0575
                                                2D  13 000D9          BEQL    15$
                                    0C    AE    01  A3 9A 000DB       MOVZBL  1(R3), PSECTALIGN               ;0577
                                    08    AE    02  A3 3C 000E0       MOVZWL  2(R3), PSECTFLAGS               ;0578
                                    04    AE    04  A3 DO 000E5       MOVL    4(R3), PSECTALLOC               ;0579
                                    24    AE        57 B0 000EA       MOVW    LENGTH, GSD_DESC                ;0580
                                    28    AE        5A DO 000EE       MOVL    OBJREC, GSD_DESC+4              ;0581
                                          24    AE  9F 000F2          PUSHAB  GSD_DESC                        ;0582
                                          1C    AB  DD 000F5          PUSHL   28(CONTEXT)                     ;0584
                                          0C    AE  9F 000F8          PUSHAB  PSECTALLOC                      ;0582
                                          14    AE  9F 000FB          PUSHAB  PSECTFLAGS
                                          1C    AE  9F 000FE          PUSHAB  PSECTALIGN
                                          30    AE  9F 00101          PUSHAB  PSECTDESC
                                    04    BB        06  FB 00104      CALLS   #6, @4(CONTEXT)
                                              015D  31 00108 15$:     BRW     34$                             ;0586
                                          02        58  D1 0010B 16$: CMPL    RECORDTYPE, #2                  ;0600
                                                    0F  13 0010E      BEQL    17$
                                          05        58  D1 00110      CMPL    RECORDTYPE, #5                  ;0601
                                                    0A  13 00113      BEQL    17$
                                          03        58  D1 00115      CMPL    RECORDTYPE, #3                  ;0602
                                                    05  13 00118      BEQL    17$
                                          06        58  D1 0011A      CMPL    RECORDTYPE, #6                  ;0603
                                                    37  12 0011D      BNEQ    20$
                                          17        6E  E9 0011F 17$:  BLBC    WORDPSECTGSD, 18$               ;0608
                                    10    AE    0A  A3 3C 00122       MOVZWL  10(R3), ENTRYMASK               ;0613
                                          57    0C  A3 9A 00127       MOVZBL  12(R3), LENGTH                  ;0614
                                          57        0D  C0 0012B      ADDL2   #13, LENGTH
                                    18    AE    06  A3 DO 0012E       MOVL    6(R3), SYMBOLVALUE              ;0615
                                          56    0C  A3 9E 00133       MOVAB   12(R3), SYMBOLSTRING            ;0616
                                                15  11 00137          BRB     19$                             ;0608
                                    10    AE    09  A3 3C 00139 18$:  MOVZWL  9(R3), ENTRYMASK                ;0622
                                          57    0B  A3 9A 0013E       MOVZBL  11(R3), LENGTH                  ;0623
                                          57        0C  C0 00142      ADDL2   #12, LENGTH
                                    18    AE    05  A3 DO 00145       MOVL    5(R3), SYMBOLVALUE              ;0624
                                          56    0B  A3 9E 0014A       MOVAB   11(R3), SYMBOLSTRING            ;0625
                                          03        58  D1 0014E 19$:  CMPL    RECORDTYPE, #3                  ;0631
                                                    05  13 00151      BEQL    21$
                                          06        58  D1 00153      CMPL    RECORDTYPE, #6                  ;0632
                                                    72  12 00156 20$:  BNEQ    27$
59                                        52        57  C1 00158 21$:  ADDL3   LENGTH, GSDOFFSET, R9           ;0635
54                                        59    04  A5 C1 0015C       ADDL3   4(R5), R9, R4
                                    01    A4        64  91 00161      CMPB    (R4), 1(R4)                     ;0640
                                                1C  1B 00165          BLEQU   22$
                                          25    AB  9F 00167          PUSHAB  37(CONTEXT)                     ;0642
                                                56  DD 0016A          PUSHL   SYMBOLSTRING
                                                02  DD 0016C          PUSHL   #2
```

```
                                    00000000G  8F   DD 0016E          PUSHL   #LIB$_ILLFMLCNT                          ;
                          00000000G 00         04   FB 00174          CALLS   #4, LIB$SIGNAL                          ;
                                    50 00000000G 8F   D0 0017B          MOVL    #LIB$_ILLFMLCNT, R0                     ; 0643
                                               04 00182               RET                                             ;
                                    50      02   A9   9E 00183 22$:    MOVAB   2(R9), R0                               ; 0645
                50          65      10           00   ED 00187          CMPZV   #0, #16, (R5), R0                       ;
                                                 1A   1A 0018C          BGTRU   24$                                     ;
                                            25   AB   9F 0018E 23$:    PUSHAB  37(CONTEXT)                             ; 0647
                                                 01   DD 00191          PUSHL   #1                                      ;
                                    00000000G    8F   DD 00193          PUSHL   #LIB$_RECTOOSML                         ;
                          00000000G 00           03   FB 00199          CALLS   #3, LIB$SIGNAL                          ;
                                    50 00000000G 8F   D0 001A0          MOVL    #LIB$_RECTOOSML, R0                     ; 0648
                                               04 001A7               RET                                             ;
                                    57           02   C0 001A8 24$:    ADDL2   #2, LENGTH                              ; 0650
                                    59      01   A4   9A 001AB          MOVZBL  1(R4), ARGCOUNT                         ; 0651
                                                 19   13 001AF          BEQL    27$                                     ;
                                                 51   D4 001B1          CLRL    I                                       ; 0652
                                                 11   11 001B3          BRB     26$                                     ;
                50                  52           57   C1 001B5 25$:    ADDL3   LENGTH, GSDOFFSET, R0                   ; 0655
                                    50      04   A5   C0 001B9          ADDL2   4(R5), R0                               ;
                                    50      01   A0   9A 001BD          MOVZBL  1(R0), R0                               ; 0658
                                    57      02 A047   9E 001C1          MOVAB   2(R0)[LENGTH], LENGTH                   ;
                EB                  51           59   F3 001C6 26$:    AOBLEQ  ARGCOUNT, I, 25$                        ; 0652
                                    01           58   D1 001CA 27$:    CMPL    RECORDTYPE, #1                          ; 0665
                                                 05   13 001CD          BEQL    28$                                     ;
                                    04           58   D1 001CF          CMPL    RECORDTYPE, #4                          ; 0666
                                                 3D   12 001D2          BNEQ    31$                                     ;
                                            10   AE   D4 001D4 28$:    CLRL    ENTRYMASK                               ; 0671
                10          02      A3           01   E0 001D7          BBS     #1, 2(R3), 29$                          ; 0672
                                    57      04   A3   9A 001DC          MOVZBL  4(R3), LENGTH                           ; 0677
                                    57           05   C0 001E0          ADDL2   #5, LENGTH                              ;
                                            18   AE   D4 001E3          CLRL    SYMBOLVALUE                             ; 0678
                                    56      04   A3   9E 001E6          MOVAB   4(R3), SYMBOLSTRING                     ; 0679
                                                 25   11 001EA          BRB     31$                                     ; 0672
                                    12           6E   E9 001EC 29$:    BLBC    WORDPSECTGSD, 30$                       ; 0685
                                    57      0A   A3   9A 001EF          MOVZBL  10(R3), LENGTH                          ; 0690
                                    57           0B   C0 001F3          ADDL2   #11, LENGTH                             ;
                      18    AE          06   A3   D0 001F6          MOVL    6(R3), SYMBOLVALUE                      ; 0691
                                    56      0A   A3   9E 001FB          MOVAB   10(R3), SYMBOLSTRING                    ; 0692
                                                 10   11 001FF          BRB     31$                                     ; 0685
                                    57      09   A3   9A 00201 30$:    MOVZBL  9(R3), LENGTH                           ; 0698
                                    57           0A   C0 00205          ADDL2   #10, LENGTH                             ;
                      18    AE          05   A3   D0 00208          MOVL    5(R3), SYMBOLVALUE                      ; 0699
                                    56      09   A3   9E 0020D          MOVAB   9(R3), SYMBOLSTRING                     ; 0700
                                                 66   95 00211 31$:    TSTB    (SYMBOLSTRING)                          ; 0707
                                                 05   13 00213          BEQL    32$                                     ;
                                    1F           66   91 00215          CMPB    (SYMBOLSTRING), #31                     ; 0708
                                                 1F   1B 00218          BLEQU   33$                                     ;
                                            25   AB   9F 0021A 32$:    PUSHAB  37(CONTEXT)                             ; 0711
                                    7E           66   9A 0021D          MOVZBL  (SYMBOLSTRING), -(SP)                   ;
                                                 56   DD 00220          PUSHL   SYMBOLSTRING                            ;
                                                 03   DD 00222          PUSHL   #3                                      ;
                                    00000000G    8F   DD 00224          PUSHL   #LIB$_ILLSYMLEN                         ;
                00000000G           00           05   FB 0022A          CALLS   #5, LIB$SIGNAL                          ;
                                    50 00000000G 8F   D0 00231          MOVL    #LIB$_ILLSYMLEN, R0                     ; 0712
                                               04 00238               RET                                             ;
                      14    AE          02   A3   3C 00239 33$:    MOVZWL  2(R3), SYMBOLFLAGS                      ; 0717
```

```
                         2C   AE          66  9B 0023E          MOVZBW  (SYMBOLSTRING), SYMBOLDESC          ; 0718
                      2E AE   B4 00242                          CLRW    SYMBOLDESC+2                        ; 0719
                      30 AE   01  A6  9E 00245                  MOVAB   1(R6), SYMBOLDESC+4                 ; 0721
                      24 AE       57  B0 00249                  MOVW    LENGTH, GSD_DESC                    ; 0722
                      28 AE       5A  D0 0024E                  MOVL    OBJREC, GSD_DESC+4                  ; 0723
                                  24 AE  9F 00252               PUSHAB  GSD_DESC                            ; 0726
                                  1C AB  DD 00255               PUSHL   28(CONTEXT)                         ; 0727
                                  18 AE  9F 00258               PUSHAB  ENTRYMASK                           ; 0726
                                  20 AE  9F 0025B               PUSHAB  SYMBOLFLAGS                         ;
                                  28 AE  9F 0025E               PUSHAB  SYMBOLVALUE                         ;
                                  40 AE  9F 00261               PUSHAB  SYMBOLDESC                          ;
                         00   BB      06  FB 00264              CALLS   #6, @0(CONTEXT)                     ;
                              52      57  C0 00268 34$:         ADDL2   LENGTH, GSDOFFSET                   ; 0728
                                  FDA5  31 0026B                BRW     1$                                  ; 0532
                              50      01  D0 0026E 35$:         MOVL    #1, R0                              ; 0748
                                          04 00271              RET                                         ; 0750

; Routine Size:  626 bytes,    Routine Base:  _LIB$CODE + 01E7
```

```
;  667          0751  1 %SBTTL  'proeom -- process EOM records';
;  668          0752  1 ROUTINE proeom : context_11 =
;  669          0753  2 BEGIN
;  670          0754  2 !
;  671          0755  2 ! Process end of module records
;  672          0756  2 !
;  673          0757  2 EXTERNAL REGISTER
;  674          0758  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;  675          0759  2
;  676          0760  2 BIND
;  677          0761  2     recdesc = context[obc_q_desc] : $BBLOCK,
;  678          0762  2     objrec = .recdesc[dsc$a_pointer] : $BBLOCK;
;  679          0763  2
;  680          0764  2
;  681          0765  2 LOCAL
;  682          0766  2     eomflags,
;  683          0767  2     transfer_psect,
;  684          0768  2     transfer_address,
;  685          0769  2     comcode,
;  686          0770  2     wordpsecteom,
;  687          0771  2     status;
;  688          0772  2
;  689          0773  2 context[obc_w_maxreclng] = obj$c_maxrecsiz;                   !Reset to maximum allowed by language
;  690          0774  2 !
;  691          0775  2 ! Check record sequence
;  692          0776  2 !
;  693          0777  3 IF NOT (status = sequence_check())
;  694          0778  2     THEN RETURN .status;
;  695          0779  2
;  696          0780  2 wordpsecteom = (.objrec[obj$b_rectyp] EQL obj$c_eomw);
;  697          0781  2 !
;  698          0782  2 ! Check record length and determine if a transfer address is present
;  699          0783  2 !
;  700          0784  3 IF (IF .wordpsecteom
;  701          0785  5         THEN ((transfer_address = .recdesc[dsc$w_length] NEQ eomw$c_eommin)
;  702          0786  6                 AND ((.recdesc[dsc$w_length] LSS eomw$c_eommx1)
;  703          0787  4                         OR (.recdesc[dsc$w_length] GTR eomw$c_eommax)))
;  704          0788  5         ELSE ((transfer_address = .recdesc[dsc$w_length] NEQ eom$c_eommin)
;  705          0789  6                 AND ((.recdesc[dsc$w_length] LSS eom$c_eommx1)
;  706          0790  3                         OR (.recdesc[dsc$w_length] GTR eom$c_eommax))))
;  707          0791  3 THEN BEGIN
;  708          0792  3     SIGNAL(lib$_illreclen,2,.recdesc[dsc$w_length],context[obc_b_modnamlng]);
;  709          0793  3     RETURN lib$_illreclen
;  710          0794  2     END;
;  711          0795  2 !
;  712          0796  2 ! Check the module compilation completion code
;  713          0797  2 !
;  714          0798  2 IF (comcode = .objrec[eom$b_comcod]) NEQ 0
;  715          0799  3 THEN BEGIN
;  716          0800  3     IF .comcode GTRU 3
;  717          0801  4     THEN BEGIN
;  718          0802  4         SIGNAL(lib$_badccc,2,.comcode,context[obc_b_modnamlng]);
;  719          0803  4         RETURN lib$_badccc
;  720          0804  4         END
;  721          0805  3     ELSE SIGNAL(.compilecodes[.comcode-1],1,context[obc_b_modnamlng]);
;  722          0806  2     END;
;  723          0807  2 !
```

```
;  724          0808  2 ! Get transfer address info if present
;  725          0809  2 !
;  726          0810  2 IF NOT .transfer_address
;  727          0811  2 THEN transfer_psect = 0
;  728          0812  2 ELSE IF .wordpsecteom
;  729          0813  3 THEN BEGIN
;  730          0814  3     transfer_psect = .objrec[eomw$w_psindx];
;  731          0815  3     transfer_address = .objrec[eomw$l_tfradr];
;  732          0816  3     eomflags = .objrec[eomw$b_tfrflg];
;  733          0817  3     END
;  734          0818  3 ELSE BEGIN
;  735          0819  3     transfer_psect = .objrec[eom$b_psindx];
;  736          0820  3     transfer_address = .objrec[eom$l_tfradr];
;  737          0821  3     eomflags = .objrec[eom$b_tfrflg];
;  738          0822  2     END;
;  739          0823  2 !
;  740          0824  2 ! Call user routine if supplied
;  741          0825  2 !
;  742          0826  2 IF .context[obc_l_eomrtn] NEQ 0
;  743          0827  2     THEN status = (.context[obc_l_eomrtn])(eomflags,transfer_psect,
;  744          0828  2                                            transfer_address,comcode,recdesc)
;  745          0829  2     ELSE status = true;
;  746          0830  2
;  747          0831  2 RETURN .status
;  748          0832  1 END;


                                01FC 00000 PROEOM: .WORD   Save R2,R3,R4,R5,R6,R7,R8                ; 0752
        58 00000000G   8F   D0 00002       MOVL    #LIB$_BADCCC, R8                                ;
        57 00000000G   8F   D0 00009       MOVL    #LIB$_ILLRECLEN, R7                             ;
        56 00000000G   00   9E 00010       MOVAB   LIB$SIGNAL, R6                                  ;
        5E             10   C2 00017       SUBL2   #16, SP                                         ;
        53       14    AB   9E 0001A       MOVAB   20(CONTEXT), R3                                 ; 0761
        52       04    A3   D0 0001E       MOVL    4(R3), R2                                       ; 0762
  20    AB     0800    8F   B0 00022       MOVW    #2048, 32(CONTEXT)                              ; 0773
FC5C    CF             00   FB 00028       CALLS   #0, SEQUENCE_CHECK                              ; 0777
        54             50   D0 0002D       MOVL    R0, STATUS                                      ;
        03             54   E8 00030       BLBS    STATUS, 1$                                      ;
                     00D3   31 00033       BRW     15$                                             ;
                       50   D4 00036 1$:    CLRL    R0                                              ; 0780
        07             62   91 00038       CMPB    (R2), #7                                        ;
                       02   12 0003B       BNEQ    2$                                              ;
                       50   D6 0003D       INCL    R0                                              ;
        55             50   D0 0003F 2$:    MOVL    R0, WORDPSECTEOM                                ;
        1D             55   E9 00042       BLBC    WORDPSECTEOM, 4$                                ; 0784
        50             63   3C 00045       MOVZWL  (R3), R0                                        ; 0785
                       51   D4 00048       CLRL    R1                                              ;
        02             50   B1 0004A       CMPW    R0, #2                                          ;
                       02   13 0004D       BEQL    3$                                              ;
                       51   D6 0004F       INCL    R1                                              ;
  04    AE             51   D0 00051 3$:    MOVL    R1, TRANSFER_ADDRESS                            ;
        37             51   E9 00055       BLBC    R1, 8$                                          ;
        08             50   B1 00058       CMPW    R0, #8                                          ; 0786
                       22   1F 0005B       BLSSU   7$                                              ;
```

```
                                09              50  B1 0005D          CMPW    R0, #9                                  ; 0787
                                                1B  11 00060          BRB     6$
                                50              63  3C 00062 4$:      MOVZWL  (R3), R0                                ; 0788
                                                51  D4 00065          CLRL    R1
                                02              50  B1 00067          CMPW    R0, #2
                                                02  13 0006A          BEQL    5$
                                                51  D6 0006C          INCL    R1
                            04  AE                  51  D0 0006E 5$:      MOVL    R1, TRANSFER_ADDRESS
                                1A                  51  E9 00072          BLBC    R1, 8$
                                07                  50  B1 00075          CMPW    R0, #7                              ; 0789
                                                05  1F 00078          BLSSU   7$
                                08                  50  B1 0007A          CMPW    R0, #8                              ; 0790
                                                10  1B 0007D 6$:      BLEQU   8$
                                        25      AB  9F 0007F 7$:      PUSHAB  37(CONTEXT)                             ; 0792
                                                50  DD 00082          PUSHL   R0
                                                02  DD 00084          PUSHL   #2
                                                57  DD 00086          PUSHL   R7
                                66                  04  FB 00088          CALLS   #4, LIB$SIGNAL
                                50                  57  D0 0008B          MOVL    R7, R0                              ; 0793
                                                    04  0008E          RET
                                6E          01      A2  9A 0008F 8$:      MOVZBL  1(R2), COMCODE                      ; 0798
                                                29  13 00093          BEQL    10$
                                50          25      AB  9E 00095          MOVAB   37(R11), R0                         ; 0802
                                03                  6E  D1 00099          CMPL    COMCODE, #3                         ; 0800
                                                10  1B 0009C          BLEQU   9$
                                                50  DD 0009E          PUSHL   R0                                  ; 0802
                                        04      AE  DD 000A0          PUSHL   COMCODE
                                                02  DD 000A3          PUSHL   #2
                                                58  DD 000A5          PUSHL   R8
                                66                  04  FB 000A7          CALLS   #4, LIB$SIGNAL
                                50                  58  D0 000AA          MOVL    R8, R0                              ; 0803
                                                    04  000AD          RET
                                                50  DD 000AE 9$:      PUSHL   R0                                  ; 0805
                                                01  DD 000B0          PUSHL   #1
                                50          08      AE  D0 000B2          MOVL    COMCODE, R0
                                        FAE8    CF40    DD 000B6          PUSHL   COMPILECODES-4[R0]
                                66                  03  FB 000BB          CALLS   #3, LIB$SIGNAL
                                05              04  AE  E8 000BE 10$:      BLBS    TRANSFER_ADDRESS, 11$               ; 0810
                                                08  AE  D4 000C2          CLRL    TRANSFER_PSECT                      ; 0811
                                                    23  11 000C5          BRB     13$
                                11                  55  E9 000C7 11$:      BLBC    WORDPSECTEOM, 12$                   ; 0812
                            08  AE              02  A2  3C 000CA          MOVZWL  2(R2), TRANSFER_PSECT               ; 0814
                            04  AE              04  A2  D0 000CF          MOVL    4(R2), TRANSFER_ADDRESS             ; 0815
                            0C  AE              08  A2  9A 000D4          MOVZBL  8(R2), EOMFLAGS                     ; 0816
                                                    0F  11 000D9          BRB     13$                                 ; 0812
                            08  AE              02  A2  9A 000DB 12$:      MOVZBL  2(R2), TRANSFER_PSECT               ; 0819
                            04  AE              03  A2  D0 000E0          MOVL    3(R2), TRANSFER_ADDRESS             ; 0820
                            0C  AE              07  A2  9A 000E5          MOVZBL  7(R2), EOMFLAGS                     ; 0821
                                                08  AB  D5 000EA 13$:      TSTL    8(CONTEXT)                          ; 0826
                                                    17  13 000ED          BEQL    14$
                                                53  DD 000EF          PUSHL   R3                                  ; 0827
                                            04  AE  9F 000F1          PUSHAB  COMCODE
                                            0C  AE  9F 000F4          PUSHAB  TRANSFER_ADDRESS
                                            14  AE  9F 000F7          PUSHAB  TRANSFER_PSECT
                                            1C  AE  9F 000FA          PUSHAB  EOMFLAGS
                            08  BB                  05  FB 000FD          CALLS   #5, @8(CONTEXT)
                                54                  50  D0 00101          MOVL    R0, STATUS
```

```
                                03  11 00104                   BRB     15$
                                                                                        ;
                             54 01  D0 00106 14$:               MOVL    #1, STATUS      ; 0829
                             50 54  D0 00109 15$:               MOVL    STATUS, R0      ; 0831
                                    04 0010C                    RET                     ; 0832

; Routine Size:  269 bytes,    Routine Base:  _LIB$CODE + 0459
```

```
;  750     0833  1 %SBTTL  'LIB$$READ_OBJECT - read an object file';
;  751     0834  1 GLOBAL ROUTINE lib$$read_object (read_routine,flags,user_context,
;  752     0835  1                                 global_routine, psect_routine,eomrec_routine,
;  753     0836  1                                 othgsd_routine,othrec_routine) =
;  754     0837  2 BEGIN
;  755     0838  2 !
;  756     0839  2 ! This routine is called to read an object file and return the contents
;  757     0840  2 !
;  758     0841  2 ! INPUTS:
;  759     0842  2 !
;  760     0843  2 !       read_routine    Routine to read the next record of an object file
;  761     0844  2 !                       It is called with onw argument as follows:
;  762     0845  2 !
;  763     0846  2 !                        (.read_routine)(user_context,record_descriptor);
;  764     0847  2 !
;  765     0848  2 !       flags           OPTIONAL - Address of longword of user-requested flags
;  766     0849  2 !                               LIB$M_LNK_1MOD - only process one module
;  767     0850  2 !
;  768     0851  2 !       user_context    OPTIONAL - Longword of context which is passed
;  769     0852  2 !                       to all called routines.
;  770     0853  2 !
;  771     0854  2 !       global_routine  OPTIONAL - Routine that is called with the name
;  772     0855  2 !                       and value of a global symbol.  It is called as:
;  773     0856  2 !
;  774     0857  2 !                       (.global_routine)(symbol_desc,symbol_value,
;  775     0858  2 !                                         symbol_flags,entry_mask,
;  776     0859  2 !                                         user_context,gsdrec);
;  777     0860  2 !
;  778     0861  2 !          WHERE:
;  779     0862  2 !               symbol_desc             is the address of a string descriptor
;  780     0863  2 !                                        for symbol name
;  781     0864  2 !               symbol_value            is the address of the symbol value
;  782     0865  2 !               symbol_flags            is the address of the symbol flags
;  783     0866  2 !               entry_mask              is the address of the entry mask
;  784     0867  2 !               user_context            is the context passed in
;  785     0868  2 !               gsdrec                  is the address of a string descriptor
;  786     0869  2 !                                        for the symbol record
;  787     0870  2 !
;  788     0871  2 !       psect_routine   OPTIONAL - routine that is called for a psect
;  789     0872  2 !                       definition.
;  790     0873  2 !
;  791     0874  2 !                       (.psect_routine)(psectname,psectalign,psectflags,
;  792     0875  2 !                                        psectalloc,user_context,gsdrec)
;  793     0876  2 !
;  794     0877  2 !       eomrec_routine  OPTIONAL - routine that is called for end of module
;  795     0878  2 !                       records
;  796     0879  2 !
;  797     0880  2 !                       (.eomrec_routine)(eomflags, transfer_psect,
;  798     0881  2 !                                         transfer_address,comcode,
;  799     0882  2 !                                         user_context,eomdesc)
;  800     0883  2 !
;  801     0884  2 !       othgsd_routine  OPTIONAL - routine that is called for all other
;  802     0885  2 !                       GSD types
;  803     0886  2 !
;  804     0887  2 !                       (.othgsd_routine)()
;  805     0888  2 !
;  806     0889  2 !       othrec_routine  OPTIONAL - routine that is called for all other
```

```
;  807     0890  2 !                          record types
;  808     0891  2 !
;  809     0892  2 !                          (.othrec_routine)()
;  810     0893  2 !
;  811     0894  2 ! OUTPUTS:
;  812     0895  2 !
;  813     0896  2 !       global_routine is called for each symbol definition
;  814     0897  2 !
;  815     0898  2 BUILTIN
;  816     0899  2     NULLPARAMETER;
;  817     0900  2
;  818     0901  2 GLOBAL REGISTER
;  819     0902  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;  820     0903  2
;  821     0904  2 LOCAL
;  822     0905  2     status,
;  823     0906  2     recdesc : REF $BBLOCK;
;  824     0907  2
;  825     0908  2 !
;  826     0909  2 ! If a context block already exists, then use it. Else allocate one
;  827     0910  2 !
;  828     0911  2 IF .lib$$gl_objctx EQL 0
;  829     0912  3 THEN IF NOT (status = lib$get_vm(%REF(obc_c_size),
;  830     0913  3                                  lib$$gl_objctx))
;  831     0914  3     THEN BEGIN
;  832     0915  3         SIGNAL(.status);
;  833     0916  3         RETURN .status
;  834     0917  2         END;
;  835     0918  2 !
;  836     0919  2 ! Initialize the context block
;  837     0920  2 !
;  838     0921  2 context = .lib$$gl_objctx;
;  839     0922  2 CH$FILL(0,obc_c_size,.context);                  !Zero the context block
;  840     0923  2 context[obc_w_maxrec[ng] = obj$c_maxrecsiz;
;  841     0924  2 context[obc_b_currectyp] = obj$c_eom;            !Initialize current record type as end of module
;  842     0925  2 IF NOT NULLPARAMETER(2)
;  843     0926  2     THEN context[obc_v_1mod] = ..flags AND lib$m_lnk_1mod;
;  844     0927  2 !
;  845     0928  2 ! Fill in routine addresses
;  846     0929  2 !
;  847     0930  2 IF NOT NULLPARAMETER(3)
;  848     0931  2     THEN context[obc_l_usrdata] = .user_context;
;  849     0932  2 IF NOT NULLPARAMETER(4)
;  850     0933  2     THEN context[obc_l_gblrtn] = .global_routine;
;  851     0934  2 IF NOT NULLPARAMETER(5)
;  852     0935  2     THEN context[obc_l_pscrtn] = .psect_routine;
;  853     0936  2 IF NOT NULLPARAMETER(6)
;  854     0937  2     THEN context[obc_l_eomrtn] = .eomrec_routine;
;  855     0938  2 IF NOT NULLPARAMETER(7)
;  856     0939  2     THEN context[obc_l_ogsrtn] = .othgsd_routine;
;  857     0940  2 IF NOT NULLPARAMETER(8)
;  858     0941  2     THEN context[obc_l_orcrtn] = .othrec_routine;
;  859     0942  2
;  860     0943  2 recdesc = context[obc_q_desc];                  !Point to descriptor
;  861     0944  2 recdesc[dsc$b_class] = dsc$k_class_d;
;  862     0945  2 !
;  863     0946  2 ! Call user routine to read file until eof returned
```

```
;  864          0947  2 !
;  865          0948  2 WHILE (.read_routine)(.context[obc_l_usrdata],.recdesc) NEQ rms$_eof
;  866          0949  3 DO BEGIN
;  867          0950  3     BIND
;  868          0951  3         objrec = .recdesc[dsc$a_pointer] : $BBLOCK;
;  869          0952  3
;  870          0953  3     IF .recdesc[dsc$w_length] GTRU .context[obc_w_maxreclng]
;  871          0954  3         OR .recdesc[dsc$w_length] EQL 0
;  872          0955  4     THEN BEGIN
;  873          0956  4         IF .context[obc_b_modnamlng] EQL 0
;  874          0957  4             THEN SIGNAL(lib$_illrecln2,1,.recdesc[dsc$w_length])
;  875          0958  4             ELSE SIGNAL(lib$_illreclen,2,.recdesc[dsc$w_length],
;  876          0959  4                                          context[obc_b_modnamlng]);
;  877          0960  4         dealloc_context();
;  878          0961  4         RETURN lib$_illreclen
;  879          0962  3         END;
;  880          0963  3
;  881          0964  3     context[obc_b_lstrectyp] = .context[obc_b_currectyp];          !Current record becomes last record
;  882          0965  3     context[obc_b_currectyp] = .objrec[obj$b_rectyp];              !Set current record type
;  883          0966  3
;  884          0967  4     IF NOT (status =
;  885          0968  5                 (CASE .objrec[obj$b_rectyp]
;  886          0969  5                         FROM obj$c_hdr TO obj$c_maxrectyp OF
;  887          0970  5     SET
;  888          0971  5
;  889          0972  5     [obj$c_hdr] : prohdr();                                    !Process hdr record
;  890          0973  5     [obj$c_gsd] : progsd();                                    !Process GSD record
;  891          0974  6     [obj$c_eom] : BEGIN                                        !Process eom record
;  892          0975  6                   status=proeom();
;  893          0976  6                   IF .context[obc_v_1mod]                    !Exit if 1 module
;  894          0977  6                       THEN EXITLOOP;
;  895          0978  6                   .status
;  896          0979  5                   END;
;  897          0980  5     [INRANGE] : true;
;  898          0981  6     [OUTRANGE] : BEGIN
;  899          0982  6                   IF .context[obc_b_modnamlng] NEQ 0
;  900          0983  6                       THEN SIGNAL(lib$_illrectyp,2,.objrec[obj$b_rectyp],
;  901          0984  6                                   context[obc_b_modnamlng])
;  902          0985  6                       ELSE SIGNAL(lib$_illrecty2,1,.objrec[obj$b_rectyp]);
;  903          0986  6                   lib$_illrectyp
;  904          0987  5                   END;
;  905          0988  5
;  906          0989  4     TES))
;  907          0990  4         THEN BEGIN
;  908          0991  4             dealloc_context();
;  909          0992  4             RETURN .status
;  910          0993  3             END;
;  911          0994  2     END;
;  912          0995  2 !
;  913          0996  2 ! Check that last record was eom record
;  914          0997  2 !
;  915          0998  2 IF .context[obc_b_currectyp] NEQ obj$c_eom
;  916          0999  3 THEN BEGIN
;  917          1000  3     SIGNAL(lib$_noeom,1,context[obc_b_modnamlng]);
;  918          1001  3     dealloc_context();
;  919          1002  3     RETURN lib$_noeom
;  920          1003  2     END;
```

```
;  921          1004  2
;  922          1005  2 dealloc_context();
;  923          1006  2
;  924          1007  2 RETURN true
;  925          1008  2
;  926          1009  1 END;                                   !Of lib$$read_object


                                                 0FFC 00000        .ENTRY  LIB$$READ_OBJECT, Save R2,R3,R4,R5,R6,R7,-      ; 0834
                                                                            R8,R9,R10,R11
                                          5A 00000000G   8F   D0 00002        MOVL    #LIB$_ILLRECLEN, R10                    ;
                                          59 00000000'   EF   9E 00009        MOVAB   LIB$$GL_OBJCTX, R9                      ;
                                          58     FB05    CF   9E 00010        MOVAB   DEALLOC_CONTEXT, R8                     ;
                                          57 00000000G   00   9E 00015        MOVAB   LIB$SIGNAL, R7                          ;
                                          5E                 04   C2 0001C        SUBL2   #4, SP                          ;
                                                             69   D5 0001F        TSTL    LIB$$GL_OBJCTX                  ; 0911
                                                             1F   12 00021        BNEQ    1$                              ;
                                                             59   DD 00023        PUSHL   R9                              ; 0912
                                                 04    AE    45   8F   9A 00025        MOVZBL  #69, 4(SP)                      ;
                                                       04    AE   9F   0002A        PUSHAB  4(SP)                           ;
                                          00000000G    00         02   FB 0002D        CALLS   #2, LIB$GET_VM                  ;
                                                       56         50   D0 00034        MOVL    R0, STATUS                      ;
                                                       08         56   E8 00037        BLBS    STATUS, 1$                      ;
                                                                  56   DD 0003A        PUSHL   STATUS                          ; 0915
                                                       67         01   FB 0003C        CALLS   #1, LIB$SIGNAL                  ;
                                                             014B   31   0003F        BRW     25$                             ; 0916
                                                       5B         69   D0 00042 1$:    MOVL    LIB$$GL_OBJCTX, CONTEXT         ; 0921
    0045    8F         00          6E         00   2C 00045        MOVC5   #0, (SP), #0, #69, (CONTEXT)                    ; 0922
                                                             6B         0004C                                               ;
                                                 20    AB    0800   8F   B0 0004D        MOVW    #2048, 32(CONTEXT)              ; 0923
                                                 23    AB         03   90 00053        MOVB    #3, 35(CONTEXT)                 ; 0924
                                                       02         6C   91 00057        CMPB    (AP), #2                        ; 0925
                                                                  0C   1F 0005A        BLSSU   2$                              ;
                                                             08    AC   D5 0005C        TSTL    8(AP)                           ;
                                                                  07   13 0005F        BEQL    2$                              ;
    22      AB         01          02         08   BC   F0 00061        INSV    @FLAGS, #2, #1, 34(CONTEXT)             ; 0926
                                                       03         6C   91 00068 2$:    CMPB    (AP), #3                        ; 0930
                                                                  0A   1F 0006B        BLSSU   3$                              ;
                                                             0C    AC   D5 0006D        TSTL    12(AP)                          ;
                                                                  05   13 00070        BEQL    3$                              ;
                                                 1C    AB    0C   AC   D0 00072        MOVL    USER_CONTEXT, 28(CONTEXT)       ; 0931
                                                       04         6C   91 00077 3$:    CMPB    (AP), #4                        ; 0932
                                                                  09   1F 0007A        BLSSU   4$                              ;
                                                             10    AC   D5 0007C        TSTL    16(AP)                          ;
                                                                  04   13 0007F        BEQL    4$                              ;
                                                       6B    10   AC   D0 00081        MOVL    GLOBAL_ROUTINE, (CONTEXT)       ; 0933
                                                       05         6C   91 00085 4$:    CMPB    (AP), #5                        ; 0934
                                                                  0A   1F 00088        BLSSU   5$                              ;
                                                             14    AC   D5 0008A        TSTL    20(AP)                          ;
                                                                  05   13 0008D        BEQL    5$                              ;
                                                 04    AB    14   AC   D0 0008F        MOVL    PSECT_ROUTINE, 4(CONTEXT)       ; 0935
                                                       06         6C   91 00094 5$:    CMPB    (AP), #6                        ; 0936
                                                                  0A   1F 00097        BLSSU   6$                              ;
                                                             18    AC   D5 00099        TSTL    24(AP)                          ;
```

```
                                    05  13 0009C          BEQL    6$                                          ;
                          08  AB  18  AC  D0 0009E          MOVL    EOMREC_ROUTINE, 8(CONTEXT)                  ; 0937
                              07      6C  91 000A3 6$:      CMPB    (AP), #7                                    ; 0938
                                      0A  1F 000A6          BLSSU   7$                                          ;
                                  1C  AC  D5 000A8          TSTL    28(AP)                                      ;
                                      05  13 000AB          BEQL    7$                                          ;
                          0C  AB  1C  AC  D0 000AD          MOVL    OTHGSD_ROUTINE, 12(CONTEXT)                 ; 0939
                              08      6C  91 000B2 7$:      CMPB    (AP), #8                                    ; 0940
                                      0A  1F 000B5          BLSSU   8$                                          ;
                                  20  AC  D5 000B7          TSTL    32(AP)                                      ;
                                      05  13 000BA          BEQL    8$                                          ;
                          10  AB  20  AC  D0 000BC          MOVL    OTHREC_ROUTINE, 16(CONTEXT)                 ; 0941
                              52  14  AB  9E 000C1 8$:      MOVAB   20(R11), RECDESC                            ; 0943
                          03  A2      02  90 000C5          MOVB    #2, 3(RECDESC)                              ; 0944
                                      52  DD 000C9 9$:      PUSHL   RECDESC                                     ; 0948
                                  1C  AB  DD 000CB          PUSHL   28(CONTEXT)                                 ;
                              04  BC      02  FB 000CE          CALLS   #2, @READ_ROUTINE                           ;
                      0001827A    8F      50  D1 000D2          CMPL    R0, #98938                                  ;
                                      03  12 000D9          BNEQ    10$                                         ;
                                  00B3    31 000DB          BRW     26$                                         ;
                          20  AB      62  B1 000DE 10$:      CMPW    (RECDESC), 32(CONTEXT)                      ; 0953
                                      04  1A 000E2          BGTRU   11$                                         ;
                                      62  B5 000E4          TSTW    (RECDESC)                                   ; 0954
                                      29  12 000E6          BNEQ    14$                                         ;
                                  25  AB  95 000E8 11$:      TSTB    37(CONTEXT)                                 ; 0956
                                      10  12 000EB          BNEQ    12$                                         ;
                                  7E      62  3C 000ED          MOVZWL  (RECDESC), -(SP)                            ; 0957
                                      01  DD 000F0          PUSHL   #1                                          ;
                              00000000G   8F  DD 000F2          PUSHL   #LIB$_ILLRECLN2                             ;
                                  67      03  FB 000F8          CALLS   #3, LIB$SIGNAL                              ;
                                      0D  11 000FB          BRB     13$                                         ;
                                  25  AB  9F 000FD 12$:      PUSHAB  37(CONTEXT)                                 ; 0959
                                  7E      62  3C 00100          MOVZWL  (RECDESC), -(SP)                            ;
                                      02  DD 00103          PUSHL   #2                                          ;
                                      5A  DD 00105          PUSHL   R10                                         ;
                                  67      04  FB 00107          CALLS   #4, LIB$SIGNAL                              ;
                                  68      00  FB 0010A 13$:      CALLS   #0, DEALLOC_CONTEXT                         ; 0960
                                  50      5A  D0 0010D          MOVL    R10, R0                                     ; 0961
                                          04  00110          RET                                                 ;
                      24  AB      23  AB  90 00111 14$:      MOVB    35(CONTEXT), 36(CONTEXT)                    ; 0964
                      23  AB      04  B2  90 00116          MOVB    @4(RECDESC), 35(CONTEXT)                    ; 0965
                07        00      04  B2  8F 0011B          CASEB   @4(RECDESC), #0, #7                         ; 0968
            0052    0061    0048      0041      00120 15$:      .WORD   18$-15$,-                                   ;
            0061    0061    0061      0061      00128                 19$-15$,-                                   ;
                                                                      22$-15$,-                                   ;
                                                                      21$-15$,-                                   ;
                                                                      22$-15$,-                                   ;
                                                                      22$-15$,-                                   ;
                                                                      22$-15$,-                                   ;
                                                                      22$-15$                                     ;
                                  25  AB  95 00130          TSTB    37(CONTEXT)                                 ; 0982
                                      14  13 00133          BEQL    16$                                         ;
                                  25  AB  9F 00135          PUSHAB  37(CONTEXT)                                 ; 0984
                                  7E  04  B2  9A 00138          MOVZBL  @4(RECDESC), -(SP)                          ;
                                      02  DD 0013C          PUSHL   #2                                          ;
                              00000000G   8F  DD 0013E          PUSHL   #LIB$_ILLRECTYP                             ;
                                  67      04  FB 00144          CALLS   #4, LIB$SIGNAL                              ;
```

```
                                                 0F  11 00147          BRB     17$                                          ;
                                  7E       04    B2  9A 00149 16$:     MOVZBL  @4(RECDESC), -(SP)                           ; 0985
                                                 01  DD 0014D          PUSHL   #1                                           ;
                                     00000000G   8F  DD 0014F          PUSHL   #LIB$_ILLRECTY2                              ;
                                  67             03  FB 00155          CALLS   #3, LIB$SIGNAL                               ;
                                  56 00000000G   8F  D0 00158 17$:     MOVL    #LIB$_ILLRECTYP, STATUS                      ; 0981
                                                 23  11 0015F          BRB     23$                                          ;
                           00B3   C8             00  FB 00161 18$:     CALLS   #0, PROHDR                                   ; 0972
                                                 05  11 00166          BRB     20$                                          ;
                           0168   C8             00  FB 00168 19$:     CALLS   #0, PROGSD                                   ; 0973
                                  56             50  D0 0016D 20$:     MOVL    R0, STATUS                                   ;
                                                 12  11 00170          BRB     23$                                          ;
                           03DA   C8             00  FB 00172 21$:     CALLS   #0, PROEOM                                   ; 0975
                                  56             50  D0 00177          MOVL    R0, STATUS                                   ;
                     05       22 AB             02  E1 0017A          BBC     #2, 34(CONTEXT), 23$                         ; 0976
                                                 10  11 0017F          BRB     26$                                          ; 0977
                                  56             01  D0 00181 22$:     MOVL    #1, STATUS                                   ; 0968
                                  03             56  E9 00184 23$:     BLBC    STATUS, 24$                                  ;
                                              FF3F   31 00187          BRW     9$                                           ;
                                  68             00  FB 0018A 24$:     CALLS   #0, DEALLOC_CONTEXT                          ; 0991
                                  50             56  D0 0018D 25$:     MOVL    STATUS, R0                                   ; 0992
                                                 04  00190             RET                                                  ;
                                  03       23    AB  91 00191 26$:     CMPB    35(CONTEXT), #3                              ; 0998
                                                 19  13 00195          BEQL    27$                                          ;
                                           25    AB  9F 00197          PUSHAB  37(CONTEXT)                                  ; 1000
                                                 01  DD 0019A          PUSHL   #1                                           ;
                                     00000000G   8F  DD 0019C          PUSHL   #LIB$_NOEOM                                  ;
                                  67             03  FB 001A2          CALLS   #3, LIB$SIGNAL                               ;
                                  68             00  FB 001A5          CALLS   #0, DEALLOC_CONTEXT                          ; 1001
                                  50 00000000G   8F  D0 001A8          MOVL    #LIB$_NOEOM, R0                              ; 1002
                                                 04  001AF             RET                                                  ;
                                  68             00  FB 001B0 27$:     CALLS   #0, DEALLOC_CONTEXT                          ; 1005
                                  50             01  D0 001B3          MOVL    #1, R0                                       ; 1007
                                                 04  001B6             RET                                                  ; 1009

; Routine Size:  439 bytes,    Routine Base:  _LIB$CODE + 0566

;  927          1010  1
;  928          1011  0 END ELUDOM


                                                                       .EXTRN  LIB$SIGNAL

;                                    PSECT SUMMARY
;
;      Name                      Bytes                          Attributes
;
;  _LIB$DATA                       12  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _LIB$CODE                     1821  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  .  ABS  .                        0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)
```

```
;                              Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776       95         0        581         00:00.7
```

```
;                                   COMMAND QUALIFIERS

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

; Size:         1809 code + 24 data bytes
; Run Time:       00:21.9
; Elapsed Time:   01:33.6
; Lines/CPU Min:   2768
; Lexemes/CPU-Min: 27009
; Memory Used:  221 pages
; Compilation Complete
```

0209 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LIBPOLYG
LIS

LIBREMQHI
LIS

LIBRENAME
LIS

LIBSCANC
LIS

LIBSIGSTO
LIS

LIBROOBJ
LIS

LIBRUNPRO
LIS

LIBSIGNAL
LIS

LIBPUTOUT
LIS

LIBREMQTI
LIS

LIBSIGRET
LIS

LIBSIMTRA
LIS

LIBSCOPY
LIS

LIBPOLYH
LIS

LIBREVERT
LIS