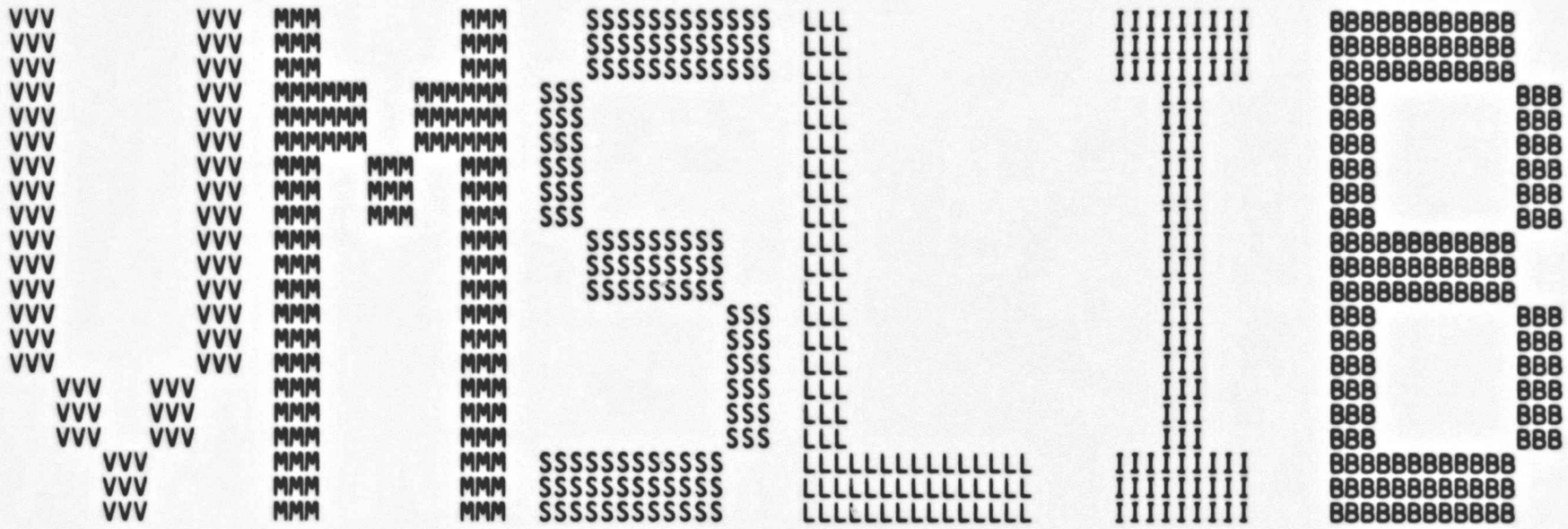
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB

FILEID**READOBJ

```
RRRRRRRR    EEEEEEEEEE    AAAAAA    DDDDDDDD      OOOOOO    BBBBBBBB          JJ
RRRRRRRR    EEEEEEEEEE    AAAAAA    DDDDDDDD      OOOOOO    BBBBBBBB          JJ
RR      RR  EE            AA  AA    DD      DD  OO      OO  BB      BB        JJ
RR      RR  EE            AA  AA    DD      DD  OO      OO  BB      BB        JJ
RR      RR  EE            AA  AA    DD      DD  OO      OO  BB      BB        JJ
RR      RR  EE            AA  AA    DD      DD  OO      OO  BB      BB        JJ
RRRRRRRR    EEEEEEEE      AA  AA    DD      DD  OO      OO  BBBBBBBB          JJ
RRRRRRRR    EEEEEEEE      AA  AA    DD      DD  OO      OO  BBBBBBBB          JJ
RR  RR      EE            AAAAAAAAAA  DD      DD  OO      OO  BB      BB  JJ    JJ
RR  RR      EE            AAAAAAAAAA  DD      DD  OO      OO  BB      BB  JJ    JJ
RR    RR    EE            AA  AA    DD      DD  OO      OO  BB      BB  JJ    JJ      ....
RR    RR    EE            AA  AA    DD      DD  OO      OO  BB      BB  JJ    JJ      ....
RR      RR  EEEEEEEEEE    AA  AA    DDDDDDDD      OOOOOO    BBBBBBBB      JJJJJJ      ....
RR      RR  EEEEEEEEEE    AA  AA    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
```

UTI
V04

```
;     1          0001  0 MODULE util$read_object (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 ADDRESSING_MODE(EXTERNAL=GENERAL,NONEXTERNAL=GENERAL),
;     4          0004  0                 IDENT = 'V04-000'
;     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 global 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 !       V03-002 BLS0225         Benn Schreiber          16-Jun-1983
;    54          0054  1 !               Add flags argument and 1MOD flag
;    55          0055  1 !
;    56          0056  1 !       V03-001 BLS0209         Benn Schreiber          27-Feb-1983
;    57          0057  1 !               Correct PSECT name for read/only OWN data
```

```
;   58          0058  1 !--
```

```
;    60          0059  1 %SBTTL  'Declarations';
;    61          0060  1 !
;    62          0061  1 ! BLISS Libraries
;    63          0062  1 !
;    64          0063  1 LIBRARY
;    65          0064  1     'SYS$LIBRARY:STARLET';                  !Definitions for OBJ$ etc.
;    66          0065  1 !
;    67          0066  1 ! Define UTIL$ psects
;    68          0067  1 !
;    69          0068  1 PSECT
;    70          0069  1     CODE = _UTIL$CODE,
;    71          0070  1     GLOBAL = _UTIL$DATA,
;    72          0071  1     OWN = _UTIL$DATA,
;    73          0072  1     PLIT = _UTIL$CODE;
;    74          0073  1
;    75          0074  1 !
;    76          0075  1 ! Data structure to describe object module
;    77          0076  1 !
;    78          0077  1 FIELD
;    79          0078  1     obc_fields =
;    80          0079  1         SET
;    81          0080  1         obc_l_gblrtn =  [0,0,32,0],         !Address of globals routine
;    82          0081  1         obc_l_pscrtn =  [4,0,32,0],         !Address of psect routine
;    83          0082  1         obc_l_eomrtn =  [8,0,32,0],         !Address of eom rec routine
;    84          0083  1         obc_l_ogsrtn =  [12,0,32,0],        !Address of other GSD routine
;    85          0084  1         obc_l_orcrtn =  [16,0,32,0],        !Address of other record routine
;    86          0085  1         obc_q_desc =    [20,0,0,0],         !Dynamic string descriptor
;    87          0086  1         obc_l_usrdata = [28,0,32,0],        !User data to pass to routines
;    88          0087  1         obc_w_maxreclng = [32,0,16,0],      !Max rec length allowed by caller
;    89          0088  1         obc_b_flags =   [34,0,8,0],         !Flags
;    90          0089  1         obc_v_mhdseen = [34,0,1,0],         ! module header seen
;    91          0090  1         obc_v_lnmseen = [34,1,1,0],         ! lang. name record seen
;    92          0091  1         obc_v_1mod =    [34,2,1,0],         ! only process one module
;    93          0092  1         obc_b_currectyp = [35,0,8,0],       !Current record type
;    94          0093  1         obc_b_lstrectyp = [36,0,8,0],       !Last record type
;    95          0094  1         obc_b_modnamlng = [37,0,8,0],       !Length of module name
;    96          0095  1         obc_t_modname = [38,0,0,0]          !Length 31
;    97          0096  1         TES;
;    98          0097  1
;    99          0098  1 LITERAL
;   100          0099  1         obc_c_size = 38+31;                 !Size of OBC structure
;   101          0100  1
;   102          0101  1 GLOBAL LITERAL
;   103          0102  1         util$m_lnk_1mod = 1;                !Bit mask for flags
;   104          0103  1
;   105          0104  1 LINKAGE
;   106          0105  1     context_11 = CALL : GLOBAL (context = 11);
;   107          0106  1
;   108          0107  1 FORWARD ROUTINE
;   109          0108  1     dealloc_context : context_11,           !Deallocate context block
;   110          0109  1     prohdr : context_11,                    !Process module header records
;   111          0110  1     progsd : context_11,                    !Process GSD records
;   112          0111  1     proeom : context_11,                    !Process end of module records
;   113          0112  1     sequence_check : context_11;            !Check sequence of object records
;   114          0113  1
;   115          0114  1 EXTERNAL ROUTINE
;   116          0115  1     lib$free_vm,                            !Deallocate virtual memory
```

```
;  117          0116  1      lib$get_vm,                         !Allocate virtual memory
;  118          0117  1      str$free1_dx;                       !Deallocate dynamic string
;  119          0118  1
;  120          0119  1  EXTERNAL LITERAL
;  121          0120  1      lnk$_badccc,                        !Illegal compilation completion code
;  122          0121  1      lnk$_eomerror,                      !Errors in eom compilation code
;  123          0122  1      lnk$_eomfatal,                      !Fatal errors in eom compilation code
;  124          0123  1      lnk$_eomwarn,                       !Warnings in eom compilation code
;  125          0124  1      lnk$_gsdtyp,                        !Illegal gsd type
;  126          0125  1      lnk$_illfmlcnt,                     !Illegal formals count
;  127          0126  1      lnk$_illmodnam,                     !Illegal module name length
;  128          0127  1      lnk$_illpsclen,                     !Illegal psect length
;  129          0128  1      lnk$_illreclen,                     !Illegal record length
;  130          0129  1      lnk$_illrecln2,                     !Illegal record length
;  131          0130  1      lnk$_illrectyp,                     !Illegal record type
;  132          0131  1      lnk$_illrecty2,                     !Illegal record type
;  133          0132  1      lnk$_illsymlen,                     !Illegal symbol length
;  134          0133  1      lnk$_noeom,                         !No end of module record in file
;  135          0134  1      lnk$_rectoosml,                     !Record too small to hold data
;  136          0135  1      lnk$_sequence,                      !Illegal record sequence
;  137          0136  1      lnk$_sequence2,                     !Illegal record sequence
;  138          0137  1      lnk$_strlvl;                        !Illegal structure level
;  139          0138  1
;  140          0139  1  LITERAL
;  141          0140  1      true = 1,
;  142          0141  1      false = 0;
;  143          0142  1
;  144          0143  1  GLOBAL
;  145          0144  1      util$gl_objctx : REF $BBLOCK FIELD(obc_fields);!pointer to context block
;  146          0145  1
;  147          0146  1  PSECT OWN = _UTIL$CODE;                 !Read-only data
;  148          0147  1
;  149          0148  1  OWN
;  150          0149  1      compilecodes : VECTOR[3,LONG]       !Translate eom compile codes into messages
;  151          0150  1                     INITIAL (lnk$_eomwarn,
;  152          0151  1                              lnk$_eomerror,
;  153          0152  1                              lnk$_eomfatal);
```

```
;   155          0153  1 %SBTTL  'dealloc_context -- deallocate context block';
;   156          0154  1 ROUTINE dealloc_context : context_11 =
;   157          0155  2 BEGIN
;   158          0156  2 !
;   159          0157  2 ! This routine deallocates the context block
;   160          0158  2 !
;   161          0159  2 EXTERNAL REGISTER
;   162          0160  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;   163          0161  2
;   164          0162  2 LOCAL
;   165          0163  2     status;
;   166          0164  2
;   167          0165  2 IF .context NEQ 0
;   168          0166  3 THEN BEGIN
;   169          0167  3     str$free1_dx(util$gl_objctx[obc_q_desc]);
;   170          0168  3     status = lib$free_vm(%REF(obc_c_size),util$gl_objctx);
;   171          0169  3     util$gl_objctx = context = 0;
;   172          0170  3     RETURN .status
;   173          0171  3     END
;   174          0172  2 ELSE RETURN true
;   175          0173  2
;   176          0174  1 END;


                                                        .TITLE  UTIL$READ_OBJECT Read and dissect object file
                                                        .IDENT  \V04-000\

                                                        .PSECT  _UTIL$DATA,NOEXE,2

                                                00000 UTIL$GL_OBJCTX::
                                                        .BLKB   4

                                                        .PSECT  _UTIL$CODE,NOWRT,2

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

                                                      UTIL$M_LNK_1MOD==   1
                                                        .EXTRN  LIB$FREE_VM, LIB$GET_VM
                                                        .EXTRN  STR$FREE1_DX, LNK$_BADCCC
                                                        .EXTRN  LNK$_EOMERROR, LNK$_EOMFATAL
                                                        .EXTRN  LNK$_EOMWARN, LNK$_GSDTYP
                                                        .EXTRN  LNK$_ILLFMLCNT, LNK$_ILLMODNAM
                                                        .EXTRN  LNK$_ILLPSCLEN, LNK$_ILLRECLEN
                                                        .EXTRN  LNK$_ILLRECLN2, LNK$_ILLRECTYP
                                                        .EXTRN  LNK$_ILLRECTY2, LNK$_ILLSYMLEN
                                                        .EXTRN  LNK$_NOEOM, LNK$_RECTOOSML
                                                        .EXTRN  LNK$_SEQUENCE, LNK$_SEQUENCE2
                                                        .EXTRN  LNK$_STRLVL

                                                0004 00000 DEALLOC_CONTEXT:
                                                        .WORD   Save R2                                         ; 0154
                                52 00000000'  00   9E 00002             MOVAB   UTIL$GL_OBJCTX, R2              ;
                                5E            04   C2 00009             SUBL2   #4, SP                          ;
                                              5B   D5 0000C             TSTL    CONTEXT                         ; 0165
                                              21   13 0000E             BEQL    1$                              ;
                             7E               62   14 C1 00010         ADDL3   #20, UTIL$GL_OBJCTX, -(SP)      ; 0167
```

```
                00000000G  00              01  FB 00014          CALLS   #1, STR$FREE1_DX                        ;
                                            52  DD 0001B          PUSHL   R2                                      ; 0168
                      04   AE        45     8F  9A 0001D          MOVZBL  #69, 4(SP)                              ;
                                      04    AE  9F 00022          PUSHAB  4(SP)                                   ;
                00000000G  00              02  FB 00025          CALLS   #2, LIB$FREE_VM                         ;
                                            5B  D4 0002C          CLRL    CONTEXT                                 ; 0169
                                            62  D4 0002E          CLRL    UTIL$GL_OBJCTX                          ;
                                                04 00030          RET                                             ; 0172
                           50               01  D0 00031  1$:     MOVL    #1, R0                                  ;
                                                04 00034          RET                                             ; 0174

; Routine Size:  53 bytes,    Routine Base:  _UTIL$CODE + 000C
```

```
;  178          0175  1 %SBTTL  'sequence_check -- check record type sequence';
;  179          0176  1 ROUTINE sequence_check : context_11 =
;  180          0177  2 BEGIN
;  181          0178  2 !
;  182          0179  2 ! Check that the record sequence is correct
;  183          0180  2 !
;  184          0181  2 ROUTINE sequence_error : context_11 =
;  185          0182  3 BEGIN
;  186          0183  3 !
;  187          0184  3 ! Signal a record sequence error
;  188          0185  3 !
;  189          0186  3 EXTERNAL REGISTER
;  190          0187  3     context = 11 : REF $BBLOCK FIELD(obc_fields);
;  191          0188  3
;  192          0189  3 IF .context[obc_b_modnamlng] NEQ 0
;  193          0190  3     THEN SIGNAL(lnk$_sequence,1,context[obc_b_modnamlng])
;  194          0191  3     ELSE SIGNAL(lnk$_sequence2);
;  195          0192  3
;  196          0193  3 RETURN lnk$_sequence
;  197          0194  2 END;


                                000C 00000 SEQUENCE_ERROR:
                                           .WORD   Save R2,R3                          ; 0181
                53 00000000G   8F  D0 00002        MOVL    #LNK$_SEQUENCE, R3       ;
                52 00000000G   00  9E 00009        MOVAB   LIB$SIGNAL, R2           ;
                         25    AB  95 00010        TSTB    37(CONTEXT)              ; 0189
                               0C  13 00013        BEQL    1$                       ;
                            25 AB  9F 00015        PUSHAB  37(CONTEXT)              ; 0190
                               01  DD 00018        PUSHL   #1                       ;
                               53  DD 0001A        PUSHL   R3                       ;
                62             03  FB 0001C        CALLS   #3, LIB$SIGNAL           ;
                               02  11 0001F        BRB     2$                       ;
                   00000000G   8F  DD 00021 1$:    PUSHL   #LNK$_SEQUENCE2          ; 0191
                62             01  FB 00027        CALLS   #1, LIB$SIGNAL           ;
                50             53  D0 0002A 2$:    MOVL    R3, R0                   ; 0193
                               04  0002D          RET                              ; 0194

; Routine Size:  46 bytes,    Routine Base:  _UTIL$CODE + 0041

;  198          0195  2 !
;  199          0196  2 ! Main body of sequence_check
;  200          0197  2 !
;  201          0198  2 EXTERNAL REGISTER
;  202          0199  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;  203          0200  2
;  204          0201  2 BIND
;  205          0202  2     recdesc = context[obc_q_desc] : $BBLOCK,
;  206          0203  2     objrec = .recdesc[dsc$a_pointer] : $BBLOCK;
;  207          0204  2
;  208          0205  2 IF .context[obc_b_currectyp] EQL obj$c_hdr
;  209          0206  3     THEN BEGIN
;  210          0207  3         IF .objrec[obj$b_subtyp] EQL obj$c_hdr_mhd
```

```
;  211   0208 4          THEN BEGIN
;  212   0209 4              IF .context[obc_b_lstrectyp] EQL obj$c_eom
;  213   0210 5              THEN BEGIN
;  214   0211 5                  context[obc_v_mhdseen] = true;              !Main mhd record has just followed eom recor
;  215   0212 5                  context[obc_v_lnmseen] = false;             !Flag no lnm mhd seen
;  216   0213 5                  RETURN true
;  217   0214 5                  END
;  218   0215 4              ELSE RETURN sequence_error()                    !Last record was not eom, signal the error
;  219   0216 4              END
;  220   0217 3          ELSE IF .context[obc_v_mhdseen]
;  221   0218 4              THEN BEGIN
;  222   0219 4                  IF .objrec[obj$b_subtyp] EQL obj$c_hdr_lnm
;  223   0220 4                      THEN context[obc_v_lnmseen] = true;
;  224   0221 4                  RETURN true
;  225   0222 4                  END
;  226   0223 3              ELSE RETURN sequence_error()
;  227   0224 3          END
;  228   0225 2      ELSE IF .context[obc_v_mhdseen]
;  229   0226 2                  AND .context[obc_v_lnmseen]
;  230   0227 3              THEN BEGIN
;  231   0228 3                  IF .context[obc_b_currectyp] EQL obj$c_eom  !If current record is end of module
;  232   0229 3                      THEN context[obc_v_mhdseen] = false;    ! then we have no mhd record
;  233   0230 3                  RETURN true
;  234   0231 3                  END
;  235   0232 2              ELSE RETURN sequence_error();
;  236   0233 2
;  237   0234 1 END;


                              0000 00000 SEQUENCE_CHECK:
                                                 .WORD   Save nothing                        ; 0176
            50        14   AB   9E 00002         MOVAB   20(CONTEXT), R0                     ; 0202
            50        04   A0   D0 00006         MOVL    4(R0), R0                           ; 0203
                      23   AB   95 0000A         TSTB    35(CONTEXT)                         ; 0205
                           25   12 0000D         BNEQ    2$                                  ;
                      01   A0   95 0000F         TSTB    1(R0)                               ; 0207
                           10   12 00012         BNEQ    1$                                  ;
            03        24   AB   91 00014         CMPB    36(CONTEXT), #3                     ; 0209
                           31   12 00018         BNEQ    4$                                  ;
       22   AB             01   88 0001A         BISB2   #1, 34(CONTEXT)                     ; 0211
       22   AB             02   8A 0001E         BICB2   #2, 34(CONTEXT)                     ; 0212
                           23   11 00022         BRB     3$                                  ; 0213
            23        22   AB   E9 00024 1$:     BLBC    34(CONTEXT), 4$                     ; 0217
            01        01   A0   91 00028         CMPB    1(R0), #1                           ; 0219
                           19   12 0002C         BNEQ    3$                                  ;
       22   AB             02   88 0002E         BISB2   #2, 34(CONTEXT)                     ; 0220
                           13   11 00032         BRB     3$                                  ; 0221
            13        22   AB   E9 00034 2$:     BLBC    34(CONTEXT), 4$                     ; 0225
  0E   22   AB             01   E1 00038         BBC     #1, 34(CONTEXT), 4$                 ; 0226
            03        23   AB   91 0003D         CMPB    35(CONTEXT), #3                     ; 0228
                           04   12 00041         BNEQ    3$                                  ;
       22   AB             01   8A 00043         BICB2   #1, 34(CONTEXT)                     ; 0229
            50             01   D0 00047 3$:     MOVL    #1, R0                              ; 0230
                                04 0004A         RET                                         ;
```

```
                          83   AF            00  FB 0004B 4$:     CALLS   #0, SEQUENCE_ERROR                ; 0232
                                                04 0004F         RET                                       ; 0234

; Routine Size:  80 bytes,    Routine Base:  _UTIL$CODE + 006F
```

```
;  239     0235  1 %SBTTL  'prohdr -- process MHD records';
;  240     0236  1 ROUTINE prohdr : context_11 =
;  241     0237  2 BEGIN
;  242     0238  2 !
;  243     0239  2 ! This routine processes MHD records
;  244     0240  2 !
;  245     0241  2 ! Inputs:
;  246     0242  2 !
;  247     0243  2 !       recdesc         Address of string descriptor for mhd record
;  248     0244  2 !
;  249     0245  2
;  250     0246  2 EXTERNAL REGISTER
;  251     0247  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;  252     0248  2
;  253     0249  2 BIND
;  254     0250  2     recdesc = context[obc_q_desc] : $BBLOCK,
;  255     0251  2     objrec = .recdesc[dsc$a_pointer] : $BBLOCK;
;  256     0252  2
;  257     0253  2 LOCAL
;  258     0254  2     status;
;  259     0255  2
;  260     0256  2 !
;  261     0257  2 ! Check record sequence
;  262     0258  2 !
;  263     0259  3 IF NOT (status = sequence_check())
;  264     0260  2     THEN RETURN .status;
;  265     0261  2 !
;  266     0262  2 ! Skip all but main module header records
;  267     0263  2 !
;  268     0264  2 IF .objrec[obj$b_subtyp] NEQ obj$c_hdr_mhd
;  269     0265  2     THEN RETURN true;
;  270     0266  2 !
;  271     0267  2 ! Check for legal structure level
;  272     0268  2 !
;  273     0269  2 IF .objrec[mhd$b_strlvl] GTRU obj$c_strlvl
;  274     0270  3 THEN BEGIN
;  275     0271  3     SIGNAL(lnk$_strlvl,1,objrec[mhd$b_namlng]);
;  276     0272  3     RETURN lnk$_strlvl
;  277     0273  2     END;
;  278     0274  2 !
;  279     0275  2 ! Check max record length supplied
;  280     0276  2 !
;  281     0277  2 IF (context[obc_w_maxreclng] = .objrec[mhd$w_recsiz]) GTRU obj$c_maxrecsiz
;  282     0278  3 THEN BEGIN
;  283     0279  3     SIGNAL(lnk$_illreclen,2,.objrec[mhd$w_recsiz],objrec[mhd$b_namlng]);
;  284     0280  3     RETURN lnk$_illreclen
;  285     0281  2     END;
;  286     0282  2 !
;  287     0283  2 ! Check module name length
;  288     0284  2 !
;  289     0285  2 IF .objrec[mhd$b_namlng] GTRU obj$c_symsiz
;  290     0286  2     OR .objrec[mhd$b_namlng] EQL 0
;  291     0287  3 THEN BEGIN
;  292     0288  3     SIGNAL(lnk$_illmodnam,.objrec[mhd$b_namlng],objrec[mhd$b_namlng]);
;  293     0289  3     RETURN lnk$_illmodnam
;  294     0290  2     END;
;  295     0291  2 !
```

```
;  296          0292  2 ! Copy module name into context block for error messages
;  297          0293  2 !
;  298          0294  2 context[obc_b_modnamlng] = .objrec[mhd$b_namlng];
;  299          0295  2 CH$MOVE(.objrec[mhd$b_namlng],objrec[mhd$t_name],context[obc_t_modname]);
;  300          0296  2 !
;  301          0297  2 ! Call user action routine for "other records" if specified
;  302          0298  2 !
;  303          0299  2 IF .context[obc_l_orcrtn] NEQ 0
;  304          0300  2     THEN status = (.context[obc_l_orcrtn])(recdesc,.context[obc_l_usrdata])
;  305          0301  2     ELSE status = true;
;  306          0302  2
;  307          0303  2 RETURN .status
;  308          0304  1 END;


                                        07FC 00000 PROHDR: .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10          ; 0236
                                5A 00000000G   8F   D0 00002       MOVL    #LNK$_ILLRECLEN, R10              ;
                                59 00000000G   8F   D0 00009       MOVL    #LNK$_STRLVL, R9                  ;
                                58 00000000G   00   9E 00010       MOVAB   LIB$SIGNAL, R8                    ;
                                56        14   AB   9E 00017       MOVAB   20(CONTEXT), R6                   ; 0250
                                52        04   A6   D0 0001B       MOVL    4(R6), R2                         ; 0251
                             8D AF              00   FB 0001F       CALLS   #0, SEQUENCE_CHECK                ; 0259
                                57              50   D0 00023       MOVL    R0, STATUS                        ;
                                03              57   E8 00026       BLBS    STATUS, 1$                        ;
                                           0085 31   00029         BRW     8$                                ;
                                      01   A2   95   0002C 1$:      TSTB    1(R2)                             ; 0264
                                           04   13   0002F         BEQL    2$                                ;
                                50              01   D0 00031       MOVL    #1, R0                            ; 0265
                                                04   00034         RET                                       ;
                                      02   A2   95   00035 2$:      TSTB    2(R2)                             ; 0269
                                           0E   13   00038         BEQL    3$                                ;
                                      05   A2   9F   0003A         PUSHAB  5(R2)                             ; 0271
                                           01   DD   0003D         PUSHL   #1                                ;
                                           59   DD   0003F         PUSHL   R9                                ;
                                68              03   FB 00041       CALLS   #3, LIB$SIGNAL                    ;
                                50              59   D0 00044       MOVL    R9, R0                            ; 0272
                                                04   00047         RET                                       ;
                                50        03   A2   3C   00048 3$:      MOVZWL  3(R2), R0                         ; 0277
                             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)                             ; 0279
                                7E    03   A2   3C   0005A         MOVZWL  3(R2), -(SP)                      ;
                                           02   DD   0005E         PUSHL   #2                                ;
                                           5A   DD   00060         PUSHL   R10                               ;
                                68              04   FB 00062       CALLS   #4, LIB$SIGNAL                    ;
                                50              5A   D0 00065       MOVL    R10, R0                           ; 0280
                                                04   00068         RET                                       ;
                                1F    05   A2   91   00069 4$:      CMPB    5(R2), #31                        ; 0285
                                           05   1A   0006D         BGTRU   5$                                ;
                                      05   A2   95   0006F         TSTB    5(R2)                             ; 0286
                                           18   12   00072         BNEQ    6$                                ;
                                      05   A2   9F   00074 5$:      PUSHAB  5(R2)                             ; 0288
                                7E    05   A2   9A   00077         MOVZBL  5(R2), -(SP)                      ;
```

```
                                    00000000G   8F   DD 0007B          PUSHL   #LNK$_ILLMODNAM                          ;
                                 68             03   FB 0C081          CALLS   #3, LIB$SIGNAL                          ;
                                 50 00000000G   8F   D0 00084          MOVL    #LNK$_ILLMODNAM, R0                     ; 0289
                                                     04 0008B          RET                                             ;
                              25 AB       05    A2   90 0008C 6$:      MOVB    5(R2), 37(CONTEXT)                      ; 0294
                                 50       05    A2   9A 00091          MOVZBL  5(R2), R0                               ; 0295
                           26 AB       06    A2        50 28 00095     MOVC3   R0, 6(R2), 38(CONTEXT)                  ;
                                                10   AB   D5 0009B     TSTL    16(CONTEXT)                             ; 0299
                                                0E   13   0009E       BEQL    7$                                      ;
                                                1C   AB   DD 000A0     PUSHL   28(CONTEXT)                             ; 0300
                                                     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$:  MOVL    #1, STATUS                              ; 0301
                                 50                  57   D0 000B1 8$:  MOVL    STATUS, R0                              ; 0303
                                                     04   000B4        RET                                             ; 0304

; Routine Size:  181 bytes,    Routine Base:  _UTIL$CODE + 00BF
```

```
;   310          0305  1 %SBTTL  'progsd -- process GSD records';
;   311          0306  1 ROUTINE progsd : context_11 =
;   312          0307  2 BEGIN
;   313          0308  2 !
;   314          0309  2 ! This routine processes GSD records
;   315          0310  2 !
;   316          0311  2 ! Inputs:
;   317          0312  2 !
;   318          0313  2 !       recdesc         Address of string descriptor for gsd record
;   319          0314  2 !
;   320          0315  2 !
;   321          0316  2 BUILTIN
;   322          0317  2     NULLPARAMETER;
;   323          0318  2
;   324          0319  2 EXTERNAL REGISTER
;   325          0320  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;   326          0321  2
;   327          0322  2 LOCAL
;   328          0323  2     symboldesc : $BBLOCK[dsc$c_s_bln],              !String descriptor for symbol name
;   329          0324  2     symbolvalue,                                    !Value of symbol
;   330          0325  2     symbolflags,                                    !Symbol flags
;   331          0326  2     gsd_desc : $BBLOCK[dsc$c_s_bln],                !String descriptor for gsd subrecord
;   332          0327  2     status,                                         !Status from processing entry point
;   333          0328  2     length,                                         !Length of def/ref
;   334          0329  2     gsdoffset,                                      !Offset into record
;   335          0330  2     objrec : REF $BBLOCK;                           !pointer to object record
;   336          0331  2
;   337          0332  2 BIND
;   338          0333  2     recdesc = context[obc_q_desc] : $BBLOCK,
;   339          0334  2     objvec = .recdesc[dsc$a_pointer] : VECTOR[,BYTE];   !Name record as byte vector
;   340          0335  2
;   341          0336  2 IF .context[obc_l_gblrtn] EQL 0                      !If no routine to process them
;   342          0337  2     THEN RETURN true;                               ! then don't bother with the record
;   343          0338  2
;   344          0339  2 gsd_desc[dsc$b_dtype] = gsd_desc[dsc$b_class] = 0;
;   345          0340  2 gsdoffset = obj$c_subtyp;                           !Init pointer into record
;   346          0341  2
;   347          0342  2 !
;   348          0343  2 ! Process the GSD record
;   349          0344  2 !
;   350          0345  2 WHILE .gsdoffset LSSU .recdesc[dsc$w_length]        !Loop through the record
;   351          0346  3 DO BEGIN
;   352          0347  3     LOCAL
;   353          0348  3         recordtype,
;   354          0349  3         wordpsectgsd;                               !Contains word of psect rather than byte
;   355          0350  3
;   356          0351  3     objrec = .recdesc[dsc$a_pointer] + .gsdoffset;  !Update record pointer
;   357          0352  5     wordpsectgsd = ((.objrec[gsd$b_gsdtyp] GEQU gsd$c_symw) !Test for word of psect number
;   358          0353  3                                 AND (.objrec[gsd$b_gsdtyp] LEQU gsd$c_prow));
;   359          0354  3
;   360          0355  4     CASE (recordtype = .objvec[.gsdoffset])         !Dispatch to process GSD
;   361          0356  3                         FROM gsd$c_psc TO gsd$c_maxrectyp OF
;   362          0357  3     SET
```

```
;  364          0358  3     [gsd$c_psc] :                                   !Psect definition
;  365          0359  3     !
;  366          0360  3     ! PSECT definitions
;  367          0361  3     !
;  368          0362  4         BEGIN
;  369          0363  4             BIND
;  370          0364  4                 psectdef = objvec[.gsdoffset] : $BBLOCK;        !Name the definition
;  371          0365  4
;  372          0366  4             LOCAL
;  373          0367  4                 psectdesc : $BBLOCK[dsc$c_s_bln],
;  374          0368  4                 psectalign,
;  375          0369  4                 psectflags,
;  376          0370  4                 psectalloc;
;  377          0371  4
;  378          0372  4             IF (.gsdoffset + gps$c_name + 1) GEQU .recdesc[dsc$w_length]
;  379          0373  5             THEN BEGIN
;  380          0374  5                 SIGNAL(lnk$_rectoosml,1,context[obc_b_modnamlng]);
;  381          0375  5                 RETURN lnk$_rectoosml
;  382          0376  4                 END;
;  383          0377  4             psectdesc[dsc$w_length] = .psectdef[gps$b_namlng];
;  384          0378  4             psectdesc[dsc$b_dtype] = psectdesc[dsc$b_class] = 0;
;  385          0379  4             psectdesc[dsc$a_pointer] = psectdef[gps$t_name];
;  386          0380  4             IF .psectdef[gps$b_namlng] EQL 0                 !Check length of psect name
;  387          0381  4                 OR .psectdef[gps$b_namlng] GTRU obj$c_symsiz
;  388          0382  5             THEN BEGIN
;  389          0383  5                 SIGNAL(lnk$_illpsclen,3,psectdef[gps$b_namlng],
;  390          0384  5                         .psectdef[gps$b_namlng],context[obc_b_modnamlng]);
;  391          0385  5                 RETURN lnk$_illpsclen
;  392          0386  4                 END;
;  393          0387  4             length = gps$c_name + .psectdef[gps$b_namlng];  !Compute length of psect def.
;  394          0388  4             IF .context[obc_l_pscrtn] NEQ 0                 !If user psect routine supplied
;  395          0389  5             THEN BEGIN                                      ! then set up and call it now
;  396          0390  5                 psectalign = .psectdef[gps$b_align];
;  397          0391  5                 psectflags = .psectdef[gps$w_flags];
;  398          0392  5                 psectalloc = .psectdef[gps$l_alloc];
;  399          0393  5                 gsd_desc[dsc$w_length] = .length;           !Set up descriptor for psect def.
;  400          0394  5                 gsd_desc[dsc$a_pointer] = .objrec;
;  401          0395  5                 (.context[obc_l_pscrtn])(psectdesc,         !Call the user routine now
;  402          0396  5                                 psectalign,psectflags,psectalloc,
;  403          0397  5                                 .context[obc_l_usrdata],gsd_desc);
;  404          0398  4                 END;
;  405          0399  4             gsdoffset = .gsdoffset + .length;               !Update pointer into record
;  406          0400  3         END;
```

```
;  408          0401  3     !
;  409          0402  3     ! All types of symbols
;  410          0403  3     !
;  411          0404  3     [gsd$c_sym TO gsd$c_prow] :                                     !All symbols
;  412          0405  4         BEGIN
;  413          0406  4             BIND
;  414          0407  4                 symbolrec = objvec[.gsdoffset] : $BBLOCK;           !Name the symbol gsd
;  415          0408  4
;  416          0409  4             LOCAL
;  417          0410  4                 entrymask,
;  418          0411  4                 symbolstring : REF VECTOR[,BYTE];                   !Pointer to symbol ascic name
;  419          0412  4
;  420          0413  4             IF .recordtype EQL gsd$c_epm                            !Process entry points and procedures
;  421          0414  4                 OR .recordtype EQL gsd$c_epmw
;  422          0415  4                 OR .recordtype EQL gsd$c_pro
;  423          0416  4                 OR .recordtype EQL gsd$c_prow
;  424          0417  5             THEN BEGIN
;  425          0418  5                 !
;  426          0419  5                 ! Process entry points and procedure definitions
;  427          0420  5                 !
;  428          0421  5                 IF .wordpsectgsd
;  429          0422  6                 THEN BEGIN
;  430          0423  6                     !
;  431          0424  6                     ! Entry point with word of psect
;  432          0425  6                     !
;  433          0426  6                     entrymask = .symbolrec[epmw$w_mask];
;  434          0427  6                     length = epmw$c_name + .symbolrec[epmw$b_namlng];
;  435          0428  6                     symbolvalue = .symbolrec[epmw$l_addrs];
;  436          0429  6                     symbolstring = symbolrec[epmw$b_namlng];
;  437          0430  6                     END
;  438          0431  6                 ELSE BEGIN
;  439          0432  6                     !
;  440          0433  6                     ! Entry point with byte of psect
;  441          0434  6                     !
;  442          0435  6                     entrymask = .symbolrec[epm$w_mask];
;  443          0436  6                     length = epm$c_name + .symbolrec[epm$b_namlng];
;  444          0437  6                     symbolvalue = .symbolrec[epm$l_addrs];
;  445          0438  6                     symbolstring = symbolrec[epm$b_namlng];
;  446          0439  5                     END;
;  447          0440  5                 !
;  448          0441  5                 ! If this is procedure definition, then skip the argument
;  449          0442  5                 !  descriptors
;  450          0443  5                 !
;  451          0444  5                 IF .recordtype EQL gsd$c_pro
;  452          0445  5                     OR .recordtype EQL gsd$c_prow
;  453          0446  6                 THEN BEGIN
;  454          0447  6                     BIND
;  455          0448  6                         formals = objvec[.gsdoffset+.length] : $BBLOCK; !Name formal argument descriptors
;  456          0449  6
;  457          0450  6                     LOCAL
;  458          0451  6                         argcount;
;  459          0452  6
;  460          0453  6                     IF .formals[fml$b_minargs] GTRU .formals[fml$b_maxargs]
;  461          0454  7                     THEN BEGIN
;  462          0455  7                         SIGNAL(lnk$_illfmlcnt,2,.symbolstring,context[obc_b_modnamlng]);
;  463          0456  7                         RETURN lnk$_illfmlcnt
;  464          0457  6                         END;
```

```
;  465     0458  6                    IF (.gsdoffset + .length + fml$c_size) GEQU .recdesc[dsc$w_length]
;  466     0459  7                    THEN BEGIN
;  467     0460  7                        SIGNAL(lnk$_rectoosml,1,context[obc_b_modnamlng]);
;  468     0461  7                        RETURN lnk$_rectoosml
;  469     0462  6                        END;
;  470     0463  6                    length = .length + fml$c_size;                  !Skip fixed part of formals
;  471     0464  6                    IF (argcount = .formals[fml$b_maxargs]) NEQ 0   !If there are argument descriptors
;  472     0465  6                    THEN INCR i FROM 1 TO .argcount                 ! then process them
;  473     0466  7                    DO BEGIN
;  474     0467  7                        BIND
;  475     0468  7                            argdesc = objvec[.gsdoffset+.length] :  !Name the argument descriptor
;  476     0469  7                                                  $BBLOCK;
;  477     0470  7
;  478     0471  7                        length = .length + .argdesc[arg$b_bytecnt] + arg$c_size;
;  479     0472  6                        END;
;  480     0473  5                END;
;  481     0474  4            END;
;  482     0475  4
;  483     0476  4            ! Process ordinary symbol definitions and references
;  484     0477  4            !
;  485     0478  4            IF .recordtype EQL gsd$c_sym
;  486     0479  4                OR .recordtype EQL gsd$c_symw
;  487     0480  5            THEN BEGIN
;  488     0481  5                !
;  489     0482  5                ! Ordinary symbol definitions and references
;  490     0483  5                !
;  491     0484  5                entrymask = 0;                                  !No entry mask
;  492     0485  5                IF NOT .symbolrec[gsy$v_def]                    !If a reference
;  493     0486  6                THEN BEGIN
;  494     0487  6                    !
;  495     0488  6                    ! Symbol reference
;  496     0489  6                    !
;  497     0490  6                    length = srf$c_name + .symbolrec[srf$b_namlng];     !Simply compute length of ref
;  498     0491  6                    symbolvalue = 0;                                    !Value is 0 if a reference
;  499     0492  6                    symbolstring = symbolrec[srf$b_namlng];
;  500     0493  6                    END
;  501     0494  6                ELSE BEGIN
;  502     0495  6                    !
;  503     0496  6                    ! Symbol definition
;  504     0497  6                    !
;  505     0498  6                    IF .wordpsectgsd                                !If a word of psect number
;  506     0499  7                    THEN BEGIN
;  507     0500  7                        !
;  508     0501  7                        ! ...with word of psect number
;  509     0502  7                        !
;  510     0503  7                        length = sdfw$c_name + .symbolrec[sdfw$b_namlng];
;  511     0504  7                        symbolvalue = .symbolrec[sdfw$l_value];         !Point to value
;  512     0505  7                        symbolstring = symbolrec[sdfw$b_namlng];        !Point to the symbol name
;  513     0506  7                        END
;  514     0507  7                    ELSE BEGIN
;  515     0508  7                        !
;  516     0509  7                        ! ...with byte of psect number
;  517     0510  7                        !
;  518     0511  7                        length = sdf$c_name + .symbolrec[sdf$b_namlng];
;  519     0512  7                        symbolvalue = .symbolrec[sdf$l_value];          !Point to symbol value
;  520     0513  7                        symbolstring = symbolrec[sdf$b_namlng];         !Point to the symbol name
;  521     0514  6                        END;
```

```
;   522          0515  5                    END;
;   523          0516  4                END;                                          !Symbol definition
;   524          0517  4
;   525          0518  4                !
;   526          0519  4                ! Check length of symbol name
;   527          0520  4                !
;   528          0521  4                IF .symbolstring[0] EQL 0                    !Check validity of symbol name
;   529          0522  5                    OR .symbolstring[0] GTRU obj$c_symsiz
;   530          0523  5                THEN BEGIN
;   531          0524  5                    SIGNAL(lnk$_illsymlen,3,.symbolstring,   !Signal illegal symbol name
;   532          0525  5                           .symbolstring[0],context[obc_b_modnamlng]);
;   533          0526  4                    RETURN lnk$_illsymlen
;   534          0527  4                    END;
;   535          0528  4                !
;   536          0529  4                ! Create string descriptor for symbol name
;   537          0530  4                !
;   538          0531  4                symbolflags = .symbolrec[sdf$w_flags];       !Get the symbol flags
;   539          0532  4                symboldesc[dsc$w_length] = .symbolstring[0];
;   540          0533  4                symboldesc[dsc$b_dtype] = 0;
;   541          0534  4                symboldesc[dsc$b_class] = 0;
;   542          0535  4                symboldesc[dsc$a_pointer] = symbolstring[1];
;   543          0536  4                gsd_desc[dsc$w_length] = .length;
;   544          0537  4                gsd_desc[dsc$a_pointer] = .objrec;
;   545          0538  5                (.context[obc_l_gblrtn])                     !Call the user global symbol routine
;   546          0539  4                    (symboldesc,symbolvalue,symbolflags,entrymask,
;   547          0540  4                                .context[obc_l_usrdata],gsd_desc);
;   548          0541  4                gsdoffset = .gsdoffset + .length;            !Update the pointer into the record
;   549          0542  3                END;
;   550          0543  3
;   551          0544  3            [gsd$c_idc] :                                    !Entity ident check
;   552          0545  4                BEGIN
;   553          0546  4 !                  BIND
;   554          0547  4 !                      entity_name = ,
;   555          0548  4 !                      entity_ident =,
;   556          0549  4 !                      object_name =;
;   557          0550  4
;   558          0551  4                    true
;   559          0552  4
;   560          0553  3                    END;
;   561          0554  3            [INRANGE] :
;   562          0555  4                BEGIN
;   563          0556  4                    true
;   564          0557  3                    END;
;   565          0558  3            TES;
;   566          0559  2            END;                                             !GSD record
;   567          0560  2
;   568          0561  2 RETURN true
;   569          0562  2
;   570          0563  1 END;                                                        !of progsd


                                     07FC 00000 PROGSD: .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10          ; 0306
                                5E         34   C2 00002        SUBL2  #52, SP                           ;
                                55      14 AB   9E 00005        MOVAB  20(CONTEXT), R5                   ; 0333
```

```
                                       6B  D5 00009        TSTL    (CONTEXT)                              ; 0336
                                       0D  13 0000B        BEQL    2$                                     ;
                                 26    AE  B4 0000D        CLRW    GSD_DESC+2                             ; 0339
                         52            01  D0 00010        MOVL    #1, GSDOFFSET                          ; 0340
     52          65      10            00  ED 00013 1$:    CMPZV   #0, #16, (R5), GSDOFFSET               ; 0345
                                       03  1A 00018        BGTRU   3$                                     ;
                                     022C  31 0001A 2$:    BRW     31$                                    ;
                 5A      52      04    A5  C1 0001D 3$:    ADDL3   4(R5), GSDOFFSET, OBJREC               ; 0351
                                       51  D4 00022        CLRL    R1                                     ; 0352
                         04      01    AA  91 00024        CMPB    1(OBJREC), #4                          ;
                                       02  1F 00028        BLSSU   4$                                     ;
                                       51  D6 0002A        INCL    R1                                     ;
                                       50  D4 0002C 4$:    CLRL    R0                                     ; 0353
                         06      01    AA  91 0002E        CMPB    1(OBJREC), #6                          ;
                                       02  1A 00032        BGTRU   5$                                     ;
                                       50  D6 00034        INCL    R0                                     ;
                         53            51  D2 00036 5$:    MCOML   R1, R3                                 ;
                 6E      50            53  CB 00039        BICL3   R3, R0, WORDPSECTGSD                   ;
                 53      52      04    A5  C1 0003D        ADDL3   4(R5), GSDOFFSET, R3                   ; 0355
                         58            63  9A 00042        MOVZBL  (R3), RECORDTYPE                       ;
                 0C      00            58  CF 00045        CASEL   RECORDTYPE, #0, #12                    ;
   009D        009D    009D          001A     00049 6$:    .WORD   7$-6$,-                                ;
   FFCA        009D    009D          009D     00051                12$-6$,-                               ;
   FFCA        FFCA    FFCA          FFCA     00059                12$-6$,-                               ;
                                     FFCA     00061                12$-6$,-                               ;
                                                                   12$-6$,-                               ;
                                                                   12$-6$,-                               ;
                                                                   12$-6$,-                               ;
                                                                   1$-6$,-                                ;
                                                                   1$-6$,-                                ;
                                                                   1$-6$,-                                ;
                                                                   1$-6$,-                                ;
                                                                   1$-6$,-                                ;
                                                                   1$-6$                                  ;
                         50      0A    A2  9E 00063 7$:    MOVAB   10(R2), R0                             ; 0372
     50          65      10            00  ED 00067        CMPZV   #0, #16, (R5), R0                      ;
                                       03  1A 0006C        BGTRU   8$                                     ;
                                     00F8  31 0006E        BRW     19$                                    ;
                       1C    AE  08    A3  9B 00071 8$:    MOVZBW  8(R3), PSECTDESC                       ; 0377
                                       1E  AE  B4 00076    CLRW    PSECTDESC+2                            ; 0378
                       20    AE  09    A3  9E 00079        MOVAB   9(R3), PSECTDESC+4                     ; 0379
                                 08    A3  95 0007E        TSTB    8(R3)                                  ; 0380
                                       06  13 00081        BEQL    9$                                     ;
                             1F  08    A3  91 00083        CMPB    8(R3), #31                             ; 0381
                                       21  1B 00087        BLEQU   10$                                    ;
                                 25    AB  9F 00089 9$:    PUSHAB  37(CONTEXT)                            ; 0384
                             7E  08    A3  9A 0008C        MOVZBL  8(R3), -(SP)                           ;
                                 08    A3  9F 00090        PUSHAB  8(R3)                                  ; 0383
                                       03  DD 00093        PUSHL   #3                                     ; 0384
                          00000000G    8F  DD 00095        PUSHL   #LNK$_ILLPSCLEN                        ;
             00000000G   00            05  FB 0009B        CALLS   #5, LIB$SIGNAL                         ;
                         50 00000000G  8F  D0 000A2        MOVL    #LNK$_ILLPSCLEN, R0                    ; 0385
                                       04     000A9        RET                                            ;
                         57      08    A3  9A 000AA 10$:   MOVZBL  8(R3), LENGTH                          ; 0387
                         57            09  C0 000AE        ADDL2   #9, LENGTH                             ;
                                 04    AB  D5 000B1        TSTL    4(CONTEXT)                             ; 0388
                                       2D  13 000B4        BEQL    11$                                    ;
```

```
                                0C   AE        01   A3   9A 000B6          MOVZBL  1(R3), PSECTALIGN                ; 0390
                                08   AE        02   A3   3C 000BB          MOVZWL  2(R3), PSECTFLAGS               ; 0391
                                04   AE        04   A3   D0 000C0          MOVL    4(R3), PSECTALLOC               ; 0392
                                24   AE             57   B0 000C5          MOVW    LENGTH, GSD_DESC                ; 0393
                                28   AE             5A   D0 000C9          MOVL    OBJREC, GSD_DESC+4              ; 0394
                                                24   AE   9F 000CD          PUSHAB  GSD_DESC                        ; 0395
                                                1C   AB   DD 000D0          PUSHL   28(CONTEXT)                     ; 0397
                                                0C   AE   9F 000D3          PUSHAB  PSECTALLOC                      ; 0395
                                                14   AE   9F 000D6          PUSHAB  PSECTFLAGS                      ;
                                                1C   AE   9F 000D9          PUSHAB  PSECTALIGN                      ;
                                                30   AE   9F 000DC          PUSHAB  PSECTDESC                       ;
                                04   BB                  06   FB 000DF          CALLS   #6, @4(CONTEXT)             ;
                                              015D   31 000E3 11$:          BRW     30$                             ; 0399
                                     02               58   D1 000E6 12$:          CMPL    RECORDTYPE, #2          ; 0413
                                                      0F   13 000E9          BEQL    13$                             ;
                                     05               58   D1 000EB          CMPL    RECORDTYPE, #5          ; 0414
                                                      0A   13 000EE          BEQL    13$                             ;
                                     03               58   D1 000F0          CMPL    RECORDTYPE, #3          ; 0415
                                                      05   13 000F3          BEQL    13$                             ;
                                     06               58   D1 000F5          CMPL    RECORDTYPE, #6          ; 0416
                                                      37   12 000F8          BNEQ    16$                             ;
                                     17               6E   E9 000FA 13$:          BLBC    WORDPSECTGSD, 14$       ; 0421
                                10   AE        0A   A3   3C 000FD          MOVZWL  10(R3), ENTRYMASK               ; 0426
                                     57        0C   A3   9A 00102          MOVZBL  12(R3), LENGTH                  ; 0427
                                     57             0D   C0 00106          ADDL2   #13, LENGTH                     ;
                                18   AE        06   A3   D0 00109          MOVL    6(R3), SYMBOLVALUE              ; 0428
                                     56        0C   A3   9E 0010E          MOVAB   12(R3), SYMBOLSTRING            ; 0429
                                                      15   11 00112          BRB     15$                             ; 0421
                                10   AE        09   A3   3C 00114 14$:          MOVZWL  9(R3), ENTRYMASK        ; 0435
                                     57        0B   A3   9A 00119          MOVZBL  11(R3), LENGTH                  ; 0436
                                     57             0C   C0 0011D          ADDL2   #12, LENGTH                     ;
                                18   AE        05   A3   D0 00120          MOVL    5(R3), SYMBOLVALUE              ; 0437
                                     56        0B   A3   9E 00125          MOVAB   11(R3), SYMBOLSTRING            ; 0438
                                     03               58   D1 00129 15$:          CMPL    RECORDTYPE, #3          ; 0444
                                                      05   13 0012C          BEQL    17$                             ;
                                     06               58   D1 0012E          CMPL    RECORDTYPE, #6          ; 0445
                                                      72   12 00131 16$:          BNEQ    23$                     ;
                           59                 52        57   C1 00133 17$:          ADDL3   LENGTH, GSDOFFSET, R9   ; 0448
                           54                 59   04   A5   C1 00137          ADDL3   4(R5), R9, R4           ;
                                01   A4                  64   91 0013C          CMPB    (R4), 1(R4)             ; 0453
                                                      1C   1B 00140          BLEQU   18$                             ;
                                                24   AB   9F 00142          PUSHAB  37(CONTEXT)                     ; 0455
                                                56   DD 00145          PUSHL   SYMBOLSTRING                    ;
                                                02   DD 00147          PUSHL   #2                              ;
                                     00000000G  8F   DD 00149          PUSHL   #LNK$_ILLFMLCNT                 ;
                           00000000G 00              04   FB 0014F          CALLS   #4, LIB$SIGNAL                  ;
                                     50 00000000G    8F   D0 00156          MOVL    #LNK$_ILLFMLCNT, R0     ; 0456
                                                      04   0015D          RET                                     ;
                                     50        02   A9   9E 0015E 18$:          MOVAB   2(R9), R0               ; 0458
                      50         65             10        00   ED 00162          CMPZV   #0, #16, (R5), R0       ;
                                                      1A   1A 00167          BGTRU   20$                             ;
                                                25   AB   9F 00169 19$:          PUSHAB  37(CONTEXT)             ; 0460
                                                01   DD 0016C          PUSHL   #1                              ;
                                     00000000G  8F   DD 0016E          PUSHL   #LNK$_RECTOOSML                 ;
                           00000000G 00              03   FB 00174          CALLS   #3, LIB$SIGNAL                  ;
                                     50 00000000G    8F   D0 0017B          MOVL    #LNK$_RECTOOSML, R0     ; 0461
                                                      04   00182          RET                                     ;
```

```
                            57           02  C0 00183 20$:   ADDL2   #2, LENGTH                          : 0463
                            59      01   A4  9A 00186        MOVZBL  1(R4), ARGCOUNT                     : 0464
                                         19  13 0018A        BEQL    23$                                 :
                                         51  D4 0018C        CLRL    I                                   : 0465
                                         11  11 0018E        BRB     22$                                 :
          50                52           57  C1 00190 21$:   ADDL3   LENGTH, GSDOFFSET, R0               : 0468
                            50      04   A5  C0 00194        ADDL2   4(R5), R0                           :
                            50      01   A0  9A 00198        MOVZBL  1(R0), R0                           : 0471
                            57      02 A047  9E 0019C        MOVAB   2(R0)[LENGTH], LENGTH               :
          EB                51           59  F3 001A1 22$:   AOBLEQ  ARGCOUNT, I, 21$                    : 0465
                            01           58  D1 001A5 23$:   CMPL    RECORDTYPE, #1                      : 0478
                                         05  13 001A8        BEQL    24$                                 :
                            04           58  D1 001AA        CMPL    RECORDTYPE, #4                      : 0479
                                         3D  12 001AD        BNEQ    27$                                 :
                                    10   AE  D4 001AF 24$:   CLRL    ENTRYMASK                           : 0484
          10          02    A3           01  E0 001B2        BBS     #1, 2(R3), 25$                      : 0485
                            57      04   A3  9A 001B7        MOVZBL  4(R3), LENGTH                       : 0490
                            57           05  C0 001BB        ADDL2   #5, LENGTH                          :
                                    18   AE  D4 001BE        CLRL    SYMBOLVALUE                         : 0491
                            56      04   A3  9E 001C1        MOVAB   4(R3), SYMBOLSTRING                 : 0492
                                         25  11 001C5        BRB     27$                                 : 0485
                            12           6E  E9 001C7 25$:   BLBC    WORDPSECTGSD, 26$                   : 0498
                            57      0A   A3  9A 001CA        MOVZBL  10(R3), LENGTH                      : 0503
                            57           0B  C0 001CE        ADDL2   #11, LENGTH                         :
                      18    AE      06   A3  D0 001D1        MOVL    6(R3), SYMBOLVALUE                  : 0504
                            56      0A   A3  9E 001D6        MOVAB   10(R3), SYMBOLSTRING                : 0505
                                         10  11 001DA        BRB     27$                                 : 0498
                            57      09   A3  9A 001DC 26$:   MOVZBL  9(R3), LENGTH                       : 0511
                            57           0A  C0 001E0        ADDL2   #10, LENGTH                         :
                      18    AE      05   A3  D0 001E3        MOVL    5(R3), SYMBOLVALUE                  : 0512
                            56      09   A3  9E 001E8        MOVAB   9(R3), SYMBOLSTRING                 : 0513
                                         66  95 001EC 27$:   TSTB    (SYMBOLSTRING)                      : 0520
                                         05  13 001EE        BEQL    28$                                 :
                            1F           66  91 001F0        CMPB    (SYMBOLSTRING), #31                 : 0521
                                         1F  1B 001F3        BLEQU   29$                                 :
                                    25   AB  9F 001F5 28$:   PUSHAB  37(CONTEXT)                         : 0524
                            7E           66  9A 001F8        MOVZBL  (SYMBOLSTRING), -(SP)               :
                                         56  DD 001FB        PUSHL   SYMBOLSTRING                        :
                                         03  DD 001FD        PUSHL   #3                                  :
                             00000000G   8F  DD 001FF        PUSHL   #LNK$_ILLSYMLEN                     :
           00000000G  00                 05  FB 00205        CALLS   #5, LIB$SIGNAL                      :
                      50     00000000G   8F  D0 0020C        MOVL    #LNK$_ILLSYMLEN, R0                 : 0525
                                             04 00213        RET                                         :
                      14    AE      02   A3  3C 00214 29$:   MOVZWL  2(R3), SYMBOLFLAGS                  : 0530
                      2C    AE           66  9B 00219        MOVZBW  (SYMBOLSTRING), SYMBOLDESC          : 0531
                                    2E   AE  B4 0021D        CLRW    SYMBOLDESC+2                        : 0532
                      30    AE      01   A6  9E 00220        MOVAB   1(R6), SYMBOLDESC+4                 : 0534
                      24    AE           57  B0 00225        MOVW    LENGTH, GSD_DESC                    : 0535
                      28    AE           5A  D0 00229        MOVL    OBJREC, GSD_DESC+4                  : 0536
                                    24   AE  9F 0022D        PUSHAB  GSD_DESC                            : 0539
                                    1C   AB  DD 00230        PUSHL   28(CONTEXT)                         : 0540
                                    18   AE  9F 00233        PUSHAB  ENTRYMASK                           : 0539
                                    20   AE  9F 00236        PUSHAB  SYMBOLFLAGS                         :
                                    28   AE  9F 00239        PUSHAB  SYMBOLVALUE                         :
                                    40   AE  9F 0023C        PUSHAB  SYMBOLDESC                          :
                      00    BB           06  FB 0023F        CALLS   #6, @0(CONTEXT)                     :
                            52           57  C0 00243 30$:   ADDL2   LENGTH, GSDOFFSET                   : 0541
```

```
                                 FDCA  31 00246          BRW     1$                              ; 0355
                           50     01  D0 00249 31$:     MOVL    #1, R0                          ; 0561
                                      04 0024C          RET                                     ; 0563

; Routine Size:  589 bytes,    Routine Base:  _UTIL$CODE + 0174
```

```
;   572          0564  1 %SBTTL  'proeom -- process EOM records';
;   573          0565  1 ROUTINE proeom : context_11 =
;   574          0566  2 BEGIN
;   575          0567  2 !
;   576          0568  2 ! Process end of module records
;   577          0569  2 !
;   578          0570  2 EXTERNAL REGISTER
;   579          0571  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;   580          0572  2
;   581          0573  2 BIND
;   582          0574  2     recdesc = context[obc_q_desc] : $BBLOCK,
;   583          0575  2     objrec = .recdesc[dsc$a_pointer] : $BBLOCK;
;   584          0576  2
;   585          0577  2
;   586          0578  2 LOCAL
;   587          0579  2     eomflags,
;   588          0580  2     transfer_psect,
;   589          0581  2     transfer_address,
;   590          0582  2     comcode,
;   591          0583  2     wordpsecteom,
;   592          0584  2     status;
;   593          0585  2
;   594          0586  2 context[obc_w_maxreclng] = obj$c_maxrecsiz;                    !Reset to maximum allowed by language
;   595          0587  2 !
;   596          0588  2 ! Check record sequence
;   597          0589  2 !
;   598          0590  3 IF NOT (status = sequence_check())
;   599          0591  2     THEN RETURN .status;
;   600          0592  2
;   601          0593  2 wordpsecteom = (.objrec[obj$b_rectyp] EQL obj$c_eomw);
;   602          0594  2 !
;   603          0595  2 ! Check record length and determine if a transfer address is present
;   604          0596  2 !
;   605          0597  3 IF (IF .wordpsecteom
;   606          0598  5         THEN ((transfer_address = .recdesc[dsc$w_length] NEQ eomw$c_eommin)
;   607          0599  6                 AND ((.recdesc[dsc$w_length] LSS eomw$c_eommx1)
;   608          0600  4                         OR (.recdesc[dsc$w_length] GTR eomw$c_eommax)))
;   609          0601  5         ELSE ((transfer_address = .recdesc[dsc$w_length] NEQ eom$c_eommin)
;   610          0602  6                 AND ((.recdesc[dsc$w_length] LSS eom$c_eommx1)
;   611          0603  3                         OR (.recdesc[dsc$w_length] GTR eom$c_eommax))))
;   612          0604  3 THEN BEGIN
;   613          0605  3     SIGNAL(lnk$_illreclen,2,.recdesc[dsc$w_length],context[obc_b_modnamlng]);
;   614          0606  3     RETURN lnk$_illreclen
;   615          0607  2     END;
;   616          0608  2 !
;   617          0609  2 ! Check the module compilation completion code
;   618          0610  2 !
;   619          0611  2 IF (comcode = .objrec[eom$b_comcod]) NEQ 0
;   620          0612  3 THEN BEGIN
;   621          0613  3     IF .comcode GTRU 3
;   622          0614  4     THEN BEGIN
;   623          0615  4         SIGNAL(lnk$_badccc,2,.comcode,context[obc_b_modnamlng]);
;   624          0616  4         RETURN lnk$_badccc
;   625          0617  4         END
;   626          0618  3     ELSE SIGNAL(.compilecodes[.comcode-1],1,context[obc_b_modnamlng]);
;   627          0619  2     END;
;   628          0620  2 !
```

```
;  629          0621  2 ! Get transfer address info if present
;  630          0622  2 !
;  631          0623  2 IF NOT .transfer_address
;  632          0624  2 THEN transfer_psect = 0
;  633          0625  2 ELSE IF .wordpsecteom
;  634          0626  3 THEN BEGIN
;  635          0627  3     transfer_psect = .objrec[eomw$w_psindx];
;  636          0628  3     transfer_address = .objrec[eomw$l_tfradr];
;  637          0629  3     eomflags = .objrec[eomw$b_tfrflg];
;  638          0630  3     END
;  639          0631  3 ELSE BEGIN
;  640          0632  3     transfer_psect = .objrec[eom$b_psindx];
;  641          0633  3     transfer_address = .objrec[eom$l_tfradr];
;  642          0634  3     eomflags = .objrec[eom$b_tfrflg];
;  643          0635  2     END;
;  644          0636  2 !
;  645          0637  2 ! Call user routine if supplied
;  646          0638  2 !
;  647          0639  2 IF .context[obc_l_eomrtn] NEQ 0
;  648          0640  2     THEN status = (.context[obc_l_eomrtn])(eomflags,transfer_psect,
;  649          0641  2                                            transfer_address,comcode,recdesc)
;  650          0642  2     ELSE status = true;
;  651          0643  2
;  652          0644  2 RETURN .status
;  653          0645  1 END;

                     03FC 00000 PROEOM:  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9          ; 0565
          59  FCA8   CF   9E 00002        MOVAB   SEQUENCE_CHECK, R9                  ;
          58  00000000G 8F D0 00007       MOVL    #LNK$_BADCCC, R8                    ;
          57  00000000G 8F D0 0000E       MOVL    #LNK$_ILLRECLEN, R7                 ;
          56  00000000G 00 9E 00015       MOVAB   LIB$SIGNAL, R6                      ;
          5E       10 C2 0001C            SUBL2   #16, SP                             ;
          53       14 AB 9E 0001F         MOVAB   20(CONTEXT), R3                     ; 0574
          52       04 A3 D0 00023         MOVL    4(R3), R2                           ; 0575
      20  AB     0800 8F B0 00027         MOVW    #2048, 32(CONTEXT)                  ; 0586
          69       00 FB 0002D            CALLS   #0, SEQUENCE_CHECK                  ; 0590
          54       50 D0 00030            MOVL    R0, STATUS                          ;
          03       54 E8 00033            BLBS    STATUS, 1$                          ;
                 00D3 31 00036            BRW     15$                                 ;
          50       50 D4 00039 1$:        CLRL    R0                                  ; 0593
          07       62 91 0003B            CMPB    (R2), #7                            ;
                   02 12 0003E            BNEQ    2$                                  ;
                   50 D6 00040            INCL    R0                                  ;
          55       50 D0 00042 2$:        MOVL    R0, WORDPSECTEOM                    ;
          1D       55 E9 00045            BLBC    WORDPSECTEOM, 4$                    ; 0597
          50       63 3C 00048            MOVZWL  (R3), R0                            ; 0598
                   51 D4 0004B            CLRL    R1                                  ;
          02       50 B1 0004D            CMPW    R0, #2                              ;
                   02 13 00050            BEQL    3$                                  ;
                   51 D6 00052            INCL    R1                                  ;
      04  AE       51 D0 00054 3$:        MOVL    R1, TRANSFER_ADDRESS                ;
          37       51 E9 00058            BLBC    R1, 8$                              ;
          08       50 B1 0005B            CMPW    R0, #8                              ; 0599
```

```
                                 22 1F 0005E          BLSSU   7$                                      ;
                              09 50 B1 00060          CMPW    R0, #9                                  ; 0600
                                 1B 11 00063          BRB     6$                                      ;
                              50 63 3C 00065 4$:      MOVZWL  (R3), R0                                ; 0601
                                 51 D4 00068          CLRL    R1                                      ;
                              02 50 B1 0006A          CMPW    R0, #2                                  ;
                                 02 13 0006D          BEQL    5$                                      ;
                                 51 D6 0006F          INCL    R1                                      ;
                           04 AE 51 D0 00071 5$:      MOVL    R1, TRANSFER_ADDRESS                    ;
                              1A 51 E9 00075          BLBC    R1, 8$                                  ;
                              07 50 B1 00078          CMPW    R0, #7                                  ; 0602
                                 05 1F 0007B          BLSSU   7$                                      ;
                              08 50 B1 0007D          CMPW    R0, #8                                  ; 0603
                                 10 1B 00080 6$:      BLEQU   8$                                      ;
                        25 AB 9F 00082 7$:            PUSHAB  37(CONTEXT)                             ; 0605
                              50 DD 00085             PUSHL   R0                                      ;
                              02 DD 00087             PUSHL   #2                                      ;
                              57 DD 00089             PUSHL   R7                                      ;
                              66 04 FB 0008B          CALLS   #4, LIB$SIGNAL                          ;
                              50 57 D0 0008E          MOVL    R7, R0                                  ; 0606
                                 04 00091             RET                                             ;
                        6E 01 A2 9A 00092 8$:         MOVZBL  1(R2), COMCODE                          ; 0611
                                 29 13 00096          BEQL    10$                                     ;
                              50 25 AB 9E 00098       MOVAB   37(R11), R0                             ; 0615
                              03 6E D1 0009C          CMPL    COMCODE, #3                             ; 0613
                                 10 1B 0009F          BLEQU   9$                                      ;
                              50 DD 000A1             PUSHL   R0                                      ; 0615
                        04 AE 04 AE DD 000A3          PUSHL   COMCODE                                 ;
                              02 DD 000A6             PUSHL   #2                                      ;
                              58 DD 000A8             PUSHL   R8                                      ;
                           66 04 FB 000AA             CALLS   #4, LIB$SIGNAL                          ;
                           50 58 D0 000AD             MOVL    R8, R0                                  ; 0616
                              04 000B0                RET                                             ;
                              50 DD 000B1 9$:         PUSHL   R0                                      ; 0618
                              01 DD 000B3             PUSHL   #1                                      ;
                        50 08 AE D0 000B5             MOVL    COMCODE, R0                             ;
                     FB7D CF40 DD 000B9               PUSHL   COMPILECODES-4[R0]                      ;
                           66 03 FB 000BE             CALLS   #3, LIB$SIGNAL                          ;
                        05 04 AE E8 000C1 10$:        BLBS    TRANSFER_ADDRESS, 11$                   ; 0623
                           08 AE D4 000C5             CLRL    TRANSFER_PSECT                          ; 0624
                              23 11 000C8             BRB     13$                                     ;
                        11 55 E9 000CA 11$:           BLBC    WORDPSECTEOM, 12$                       ; 0625
                  08 AE 02 A2 3C 000CD                MOVZWL  2(R2), TRANSFER_PSECT                   ; 0627
                  04 AE 04 A2 D0 000D2                MOVL    4(R2), TRANSFER_ADDRESS                 ; 0628
                  0C AE 08 A2 9A 000D7                MOVZBL  8(R2), EOMFLAGS                         ; 0629
                              0F 11 000DC             BRB     13$                                     ; 0625
                  08 AE 02 A2 9A 000DE 12$:           MOVZBL  2(R2), TRANSFER_PSECT                   ; 0632
                  04 AE 03 A2 D0 000E3                MOVL    3(R2), TRANSFER_ADDRESS                 ; 0633
                  0C AE 07 A2 9A 000E8                MOVZBL  7(R2), EOMFLAGS                         ; 0634
                        08 AB D5 000ED 13$:           TSTL    8(CONTEXT)                              ; 0639
                              17 13 000F0             BEQL    14$                                     ;
                              53 DD 000F2             PUSHL   R3                                      ; 0640
                        04 AE 9F 000F4                PUSHAB  COMCODE                                 ;
                        0C AE 9F 000F7                PUSHAB  TRANSFER_ADDRESS                        ;
                        14 AE 9F 000FA                PUSHAB  TRANSFER_PSECT                          ;
                        1C AE 9F 000FD                PUSHAB  EOMFLAGS                                ;
                  08 BB 05 FB 00100                   CALLS   #5, @8(CONTEXT)                         ;
```

```
                        54        50  D0 00104          MOVL    R0, STATUS                      ;
                                  03  11 00107          BRB     15$                             ;
                        54        01  D0 00109 14$:     MOVL    #1, STATUS                      ; 0642
                        50        54  D0 0010C 15$:     MOVL    STATUS, R0                      ; 0644
                                      04 0010F          RET                                     ; 0645

; Routine Size:  272 bytes,    Routine Base:  _UTIL$CODE + 03C1
```

```
;  655     0646  1 %SBTTL  'UTIL$READ_OBJECT - read an object file';
;  656     0647  1 GLOBAL ROUTINE util$read_object (read_routine,flags,user_context,
;  657     0648  1                                  global_routine, psect_routine,eomrec_routine,
;  658     0649  1                                  othgsd_routine,othrec_routine) =
;  659     0650  2 BEGIN
;  660     0651  2 !
;  661     0652  2 ! This routine is called to read an object file and return the contents
;  662     0653  2 !
;  663     0654  2 ! INPUTS:
;  664     0655  2 !
;  665     0656  2 !       read_routine    Routine to read the next record of an object file
;  666     0657  2 !                       It is called with onw argument as follows:
;  667     0658  2 !
;  668     0659  2 !                        (.read_routine)(user_context,record_descriptor);
;  669     0660  2 !
;  670     0661  2 !       flags           OPTIONAL - Address of longword of user-requested flags
;  671     0662  2 !                               UTIL$M_LNK_1MOD - only process one module
;  672     0663  2 !
;  673     0664  2 !       user_context    OPTIONAL - Longword of context which is passed
;  674     0665  2 !                       to all called routines.
;  675     0666  2 !
;  676     0667  2 !       global_routine  OPTIONAL - Routine that is called with the name
;  677     0668  2 !                       and value of a global symbol.  It is called as:
;  678     0669  2 !
;  679     0670  2 !                       (.global_routine)(symbol_desc,symbol_value,
;  680     0671  2 !                                         symbol_flags,entry_mask,
;  681     0672  2 !                                         user_context,gsdrec);
;  682     0673  2 !
;  683     0674  2 !          WHERE:
;  684     0675  2 !               symbol_desc             is the address of a string descriptor
;  685     0676  2 !                                        for symbol name
;  686     0677  2 !               symbol_value            is the address of the symbol value
;  687     0678  2 !               symbol_flags            is the address of the symbol flags
;  688     0679  2 !               entry_mask              is the address of the entry mask
;  689     0680  2 !               user_context            is the context passed in
;  690     0681  2 !               gsdrec                  is the address of a string descriptor
;  691     0682  2 !                                        for the symbol record
;  692     0683  2 !
;  693     0684  2 !       psect_routine   OPTIONAL - routine that is called for a psect
;  694     0685  2 !                       definition.
;  695     0686  2 !
;  696     0687  2 !                       (.psect_routine)(psectname,psectalign,psectflags,
;  697     0688  2 !                                        psectalloc,user_context,gsdrec)
;  698     0689  2 !
;  699     0690  2 !       eomrec_routine  OPTIONAL - routine that is called for end of module
;  700     0691  2 !                       records
;  701     0692  2 !
;  702     0693  2 !                       (.eomrec_routine)(eomflags, transfer_psect,
;  703     0694  2 !                                         transfer_address,comcode,
;  704     0695  2 !                                         user_context,eomdesc)
;  705     0696  2 !
;  706     0697  2 !       othgsd_routine  OPTIONAL - routine that is called for all other
;  707     0698  2 !                       GSD types
;  708     0699  2 !
;  709     0700  2 !                       (.othgsd_routine)()
;  710     0701  2 !
;  711     0702  2 !       othrec_routine  OPTIONAL - routine that is called for all other
```

```
;   712          0703  2 !                          record types
;   713          0704  2 !
;   714          0705  2 !                          (.othrec_routine)()
;   715          0706  2 !
;   716          0707  2 ! OUTPUTS:
;   717          0708  2 !
;   718          0709  2 !       global_routine is called for each symbol definition
;   719          0710  2 !
;   720          0711  2 BUILTIN
;   721          0712  2     NULLPARAMETER;
;   722          0713  2
;   723          0714  2 GLOBAL REGISTER
;   724          0715  2     context = 11 : REF $BBLOCK FIELD(obc_fields);
;   725          0716  2
;   726          0717  2 LOCAL
;   727          0718  2     status,
;   728          0719  2     recdesc : REF $BBLOCK;
;   729          0720  2
;   730          0721  2 !
;   731          0722  2 ! If a context block already exists, then use it. Else allocate one
;   732          0723  2 !
;   733          0724  2 IF .util$gl_objctx EQL 0
;   734          0725  3 THEN IF NOT (status = lib$get_vm(%REF(obc_c_size),
;   735          0726  3                                 util$gl_objctx))
;   736          0727  3     THEN BEGIN
;   737          0728  3         SIGNAL(.status);
;   738          0729  3         RETURN .status
;   739          0730  2         END;
;   740          0731  2 !
;   741          0732  2 ! Initialize the context block
;   742          0733  2 !
;   743          0734  2 context = .util$gl_objctx;
;   744          0735  2 CH$FILL(0,obc_c_size,.context);                  !Zero the context block
;   745          0736  2 context[obc_w_maxrec[ng] = obj$c_maxrecsiz;
;   746          0737  2 context[obc_b_currectyp] = obj$c_eom;            !Initialize current record type as end of module
;   747          0738  2 IF NOT NULLPARAMETER(2)
;   748          0739  2     THEN context[obc_v_1mod] = ..flags AND util$m_lnk_1mod;
;   749          0740  2 !
;   750          0741  2 ! Fill in routine addresses
;   751          0742  2 !
;   752          0743  2 IF NOT NULLPARAMETER(3)
;   753          0744  2     THEN context[obc_l_usrdata] = .user_context;
;   754          0745  2 IF NOT NULLPARAMETER(4)
;   755          0746  2     THEN context[obc_l_gblrtn] = .global_routine;
;   756          0747  2 IF NOT NULLPARAMETER(5)
;   757          0748  2     THEN context[obc_l_pscrtn] = .psect_routine;
;   758          0749  2 IF NOT NULLPARAMETER(6)
;   759          0750  2     THEN context[obc_l_eomrtn] = .eomrec_routine;
;   760          0751  2 IF NOT NULLPARAMETER(7)
;   761          0752  2     THEN context[obc_l_ogsrtn] = .othgsd_routine;
;   762          0753  2 IF NOT NULLPARAMETER(8)
;   763          0754  2     THEN context[obc_l_orcrtn] = .othrec_routine;
;   764          0755  2
;   765          0756  2 recdesc = context[obc_q_desc];                          !Point to descriptor
;   766          0757  2 recdesc[dsc$b_class] = dsc$k_class_d;
;   767          0758  2 !
;   768          0759  2 ! Call user routine to read file until eof returned
```

```
;   769          0760  2 !
;   770          0761  2 WHILE (.read_routine)(.context[obc_l_usrdata],.recdesc) NEQ rms$_eof
;   771          0762  3 DO BEGIN
;   772          0763  3     BIND
;   773          0764  3         objrec = .recdesc[dsc$a_pointer] : $BBLOCK;
;   774          0765  3
;   775          0766  3     IF .recdesc[dsc$w_length] GTRU .context[obc_w_maxreclng]
;   776          0767  3         OR .recdesc[dsc$w_length] EQL 0
;   777          0768  4     THEN BEGIN
;   778          0769  4         IF .context[obc_b_modnamlng] EQL 0
;   779          0770  4             THEN SIGNAL(lnk$_illrecln2,1,.recdesc[dsc$w_length])
;   780          0771  4             ELSE SIGNAL(lnk$_illreclen,2,.recdesc[dsc$w_length],
;   781          0772  4                                          context[obc_b_modnamlng]);
;   782          0773  4         dealloc_context();
;   783          0774  4         RETURN lnk$_illreclen;
;   784          0775  3         END;
;   785          0776  3
;   786          0777  3     context[obc_b_lstrectyp] = .context[obc_b_currectyp];          !Current record becomes last record
;   787          0778  3     context[obc_b_currectyp] = .objrec[obj$b_rectyp];              !Set current record type
;   788          0779  3
;   789          0780  4     IF NOT (status =
;   790          0781  5             (CASE .objrec[obj$b_rectyp]
;   791          0782  5                 FROM obj$c_hdr TO obj$c_maxrectyp OF
;   792          0783  5     SET
;   793          0784  5
;   794          0785  5     [obj$c_hdr] : prohdr();                                    !Process hdr record
;   795          0786  5     [obj$c_gsd] : progsd();                                    !Process GSD record
;   796          0787  6     [obj$c_eom] : BEGIN                                        !Process eom record
;   797          0788  6                   proeom();
;   798          0789  6                   IF .context[obc_v_1mod]                    !Exit if 1 module
;   799          0790  6                       THEN EXITLOOP;
;   800          0791  5                   END;
;   801          0792  5     [INRANGE] : true;
;   802          0793  6     [OUTRANGE] : BEGIN
;   803          0794  6                   IF .context[obc_b_modnamlng] NEQ 0
;   804          0795  6                       THEN SIGNAL(lnk$_illrectyp,2,.objrec[obj$b_rectyp],
;   805          0796  6                                   context[obc_b_modnamlng])
;   806          0797  6                       ELSE SIGNAL(lnk$_illrecty2,1,.objrec[obj$b_rectyp]);
;   807          0798  6                   lnk$_illrectyp
;   808          0799  5                   END;
;   809          0800  5
;   810          0801  4     TES))
;   811          0802  4         THEN BEGIN
;   812          0803  4             dealloc_context();
;   813          0804  4             RETURN .status;
;   814          0805  3             END;
;   815          0806  2     END;
;   816          0807  2 !
;   817          0808  2 ! Check that last record was eom record
;   818          0809  2 !
;   819          0810  2 IF .context[obc_b_currectyp] NEQ obj$c_eom
;   820          0811  3 THEN BEGIN
;   821          0812  3     SIGNAL(lnk$_noeom,1,context[obc_b_modnamlng]);
;   822          0813  3     dealloc_context();
;   823          0814  3     RETURN lnk$_noeom
;   824          0815  2     END;
;   825          0816  2
```

```
;  826          0817  2 dealloc_context();
;  827          0818  2
;  828          0819  2 RETURN true
;  829          0820  2
;  830          0821  1 END;                                              !Of util$read_object
; INFO#212                L1:0647
; Null expression appears in value-required context
```

```
                                                 0FFC 00000        .ENTRY  UTIL$READ_OBJECT, Save R2,R3,R4,R5,R6,R7,-  ; 0647
                                                                           R8,R9,R10,R11
                                          5A 00000000G   8F  D0 00002    MOVL    #LNK$_ILLRECLEN, R10
                                          59 00000000'   00  9E 00009    MOVAB   UTIL$GL_OBJCTX, R9
                                          58 00000000G   00  9E 00010    MOVAB   LIB$SIGNAL, R8
                                          57      FB20   CF  9E 00017    MOVAB   DEALLOC_CONTEXT, R7
                                          5E             04  C2 0001C    SUBL2   #4, SP
                                                         69  D5 0001F    TSTL    UTIL$GL_OBJCTX                          ; 0724
                                                         1F  12 00021    BNEQ    1$
                                                         59  DD 00023    PUSHL   R9                                      ; 0725
                                    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                                  ; 0728
                                          68             01  FB 0003C    CALLS   #1, LIB$SIGNAL
                                                       014A  31 0003F    BRW     25$                                     ; 0729
                                          5B             69  D0 00042 1$: MOVL    UTIL$GL_OBJCTX, CONTEXT                 ; 0734
   0045   8F              00             6E             00  2C 00045    MOVC5   #0, (SP), #0, #69, (CONTEXT)            ; 0735
                                                         6B             0004C
                                    20    AB     0800    8F  B0 0004D    MOVW    #2048, 32(CONTEXT)                      ; 0736
                                    23    AB             03  90 00053    MOVB    #3, 35(CONTEXT)                         ; 0737
                                          02             6C  91 00057    CMPB    (AP), #2                                ; 0738
                                                         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)             ; 0739
                                          03             6C  91 00068 2$: CMPB    (AP), #3                                ; 0743
                                                         0A  1F 0006B    BLSSU   3$
                                                   0C    AC  D5 0006D    TSTL    12(AP)
                                                         05  13 00070    BEQL    3$
                                    1C    AB       0C    AC  D0 00072    MOVL    USER_CONTEXT, 28(CONTEXT)               ; 0744
                                          04             6C  91 00077 3$: CMPB    (AP), #4                                ; 0745
                                                         09  1F 0007A    BLSSU   4$
                                                   10    AC  D5 0007C    TSTL    16(AP)
                                                         04  13 0007F    BEQL    4$
                                          6B       10    AC  D0 00081    MOVL    GLOBAL_ROUTINE, (CONTEXT)               ; 0746
                                          05             6C  91 00085 4$: CMPB    (AP), #5                                ; 0747
                                                         0A  1F 00088    BLSSU   5$
                                                   14    AC  D5 0008A    TSTL    20(AP)
                                                         05  13 0008D    BEQL    5$
                                    04    AB       14    AC  D0 0008F    MOVL    PSECT_ROUTINE, 4(CONTEXT)               ; 0748
                                          06             6C  91 00094 5$: CMPB    (AP), #6                                ; 0749
                                                         0A  1F 00097    BLSSU   6$
```

```
                          18   AC  D5 00099        TSTL    24(AP)                              ;
                               05  13 0009C        BEQL    6$                                  ;
              08   AB     18   AC  D0 0009E        MOVL    EOMREC_ROUTINE, 8(CONTEXT)          ; 0750
                   07          6C  91 000A3 6$:    CMPB    (AP), #7                            ; 0751
                               0A  1F 000A6        BLSSU   7$                                  ;
                          1C   AC  D5 000A8        TSTL    28(AP)                              ;
                               05  13 000AB        BEQL    7$                                  ;
              0C   AB     1C   AC  D0 000AD        MOVL    OTHGSD_ROUTINE, 12(CONTEXT)         ; 0752
                   08          6C  91 000B2 7$:    CMPB    (AP), #8                            ; 0753
                               0A  1F 000B5        BLSSU   8$                                  ;
                          20   AC  D5 000B7        TSTL    32(AP)                              ;
                               05  13 000BA        BEQL    8$                                  ;
              10   AB     20   AC  D0 000BC        MOVL    OTHREC_ROUTINE, 16(CONTEXT)         ; 0754
                   52     14   AB  9E 000C1 8$:    MOVAB   20(R11), RECDESC                    ; 0756
              03   A2          02  90 000C5        MOVB    #2, 3(RECDESC)                      ; 0757
                               52  DD 000C9 9$:    PUSHL   RECDESC                             ; 0761
                          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$                                 ;
                        00B2       31 000DB        BRW     26$                                 ;
              20   AB          62  B1 000DE 10$:   CMPW    (RECDESC), 32(CONTEXT)              ; 0766
                               04  1A 000E2        BGTRU   11$                                 ;
                               62  B5 000E4        TSTW    (RECDESC)                           ; 0767
                               29  12 000E6        BNEQ    14$                                 ;
                          25   AB  95 000E8 11$:   TSTB    37(CONTEXT)                         ; 0769
                               10  12 000EB        BNEQ    12$                                 ;
                   7E          62  3C 000ED        MOVZWL  (RECDESC), -(SP)                    ; 0770
                               01  DD 000F0        PUSHL   #1                                  ;
               00000000G  8F       DD 000F2        PUSHL   #LNK$_ILLRECLN2                     ;
                   68          03  FB 000F8        CALLS   #3, LIB$SIGNAL                      ;
                               0D  11 000FB        BRB     13$                                 ;
                          25   AB  9F 000FD 12$:   PUSHAB  37(CONTEXT)                         ; 0772
                   7E          62  3C 00100        MOVZWL  (RECDESC), -(SP)                    ;
                               02  DD 00103        PUSHL   #2                                  ;
                               5A  DD 00105        PUSHL   R10                                 ;
                   68          04  FB 00107        CALLS   #4, LIB$SIGNAL                      ;
                   67          00  FB 0010A 13$:   CALLS   #0, DEALLOC_CONTEXT                 ; 0773
                   50          5A  D0 0010D        MOVL    R10, R0                             ; 0774
                                   04 00110        RET                                         ;
              24   AB     23   AB  90 00111 14$:   MOVB    35(CONTEXT), 36(CONTEXT)            ; 0777
              23   AB     04   B2  90 00116        MOVB    @4(RECDESC), 35(CONTEXT)            ; 0778
                   07     00       
                          04   B2  8F 0011B        CASEB   @4(RECDESC), #0, #7                 ; 0781
  0052         0060       0048  0041    00120 15$: .WORD   18$-15$,-                           ;
  0060         0060       0060  0060    00128              19$-15$,-                           ;
                                                           22$-15$,-                           ;
                                                           21$-15$,-                           ;
                                                           22$-15$,-                           ;
                                                           22$-15$,-                           ;
                                                           22$-15$,-                           ;
                                                           22$-15$                             ;
                          25   AB  95 00130        TSTB    37(CONTEXT)                         ; 0794
                               14  13 00133        BEQL    16$                                 ;
                          25   AB  9F 00135        PUSHAB  37(CONTEXT)                         ; 0796
                   7E     04   B2  9A 00138        MOVZBL  @4(RECDESC), -(SP)                  ;
                               02  DD 0013C        PUSHL   #2                                  ;
               00000000G  8F       DD 0013E        PUSHL   #LNK$_ILLRECTYP                     ;
```

```
                                68            04 FB 00144          CALLS   #4, LIB$SIGNAL                                ;
                                              0F 11 00147          BRB     17$                                            ;
                                7E      04    B2 9A 00149 16$:     MOVZBL  @4(RECDESC), -(SP)                             ; 0797
                                              01 DD 0014D          PUSHL   #1                                             ;
                                   00000000G  8F DD 0014F          PUSHL   #LNK$_ILLRECTY2                                ;
                                68            03 FB 00155          CALLS   #3, LIB$SIGNAL                                 ;
                                56 00000000G  8F D0 00158 17$:     MOVL    #LNK$_ILLRECTYP, STATUS                        ; 0793
                                              22 11 0015F          BRB     23$                                            ;
                          00B3  C7            00 FB 00161 18$:     CALLS   #0, PROHDR                                     ; 0785
                                              05 11 00166          BRB     20$                                            ;
                          0168  C7            00 FB 00168 19$:     CALLS   #0, PROGSD                                     ; 0786
                                56            50 D0 0016D 20$:     MOVL    R0, STATUS                                     ;
                                              11 11 00170          BRB     23$                                            ;
                          03B5  C7            00 FB 00172 21$:     CALLS   #0, PROEOM                                     ; 0788
                    14      22  AB            02 E0 00177          BBS     #2, 34(CONTEXT), 26$                           ; 0789
                                              56 D4 0017C          CLRL    STATUS                                         ; 0781
                                              03 11 0017E          BRB     23$                                            ;
                                56            01 D0 00180 22$:     MOVL    #1, STATUS                                     ;
                                03            56 E9 00183 23$:     BLBC    STATUS, 24$                                    ;
                                          FF40  31 00186          BRW     9$                                             ;
                                67            00 FB 00189 24$:     CALLS   #0, DEALLOC_CONTEXT                            ; 0803
                                50            56 D0 0018C 25$:     MOVL    STATUS, R0                                     ; 0804
                                              04 0018F             RET                                                    ;
                                03      23    AB 91 00190 26$:     CMPB    35(CONTEXT), #3                                ; 0810
                                              19 13 00194          BEQL    27$                                            ;
                                      25    AB 9F 00196          PUSHAB  37(CONTEXT)                                    ; 0812
                                              01 DD 00199          PUSHL   #1                                             ;
                                   00000000G  8F DD 0019B          PUSHL   #LNK$_NOEOM                                    ;
                                68            03 FB 001A1          CALLS   #3, LIB$SIGNAL                                 ;
                                67            00 FB 001A4          CALLS   #0, DEALLOC_CONTEXT                            ; 0813
                                50 00000000G  8F D0 001A7          MOVL    #LNK$_NOEOM, R0                                ; 0814
                                              04 001AE             RET                                                    ;
                                67            00 FB 001AF 27$:     CALLS   #0, DEALLOC_CONTEXT                            ; 0817
                                50            01 D0 001B2          MOVL    #1, R0                                         ; 0819
                                              04 001B5             RET                                                    ; 0821

; Routine Size:  438 bytes,    Routine Base:  _UTIL$CODE + 04D1

;  831          0822  1
;  832          0823  0 END ELUDOM


                                                                   .EXTRN  LIB$SIGNAL

;                                   PSECT SUMMARY

;       Name                        Bytes                         Attributes

;  _UTIL$DATA                          4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  _UTIL$CODE                       1671  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,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        77          0        581         00:01.0



; Information:  1
; Warnings:     0
; Errors:       0


;                                    COMMAND QUALIFIERS

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

; Size:          1659 code + 16 data bytes
; Run Time:        00:27.7
; Elapsed Time:    00:29.5
; Lines/CPU Min:    1783
; Lexemes/CPU-Min: 17926
; Memory Used:  207 pages
; Compilation Complete
```

READOBJ
LIS

SCREEN
LIS

LIBEXECLI
LIS

MOUNTMSG
LIS

PASMSG
LIS

LIBFILPRO
LIS

LIBNETCON
LIS

LIBEXTCON
LIS

LIBMERGE
LIS

LNKMSG
LIS

REPORTIOE
LIS

LIBUNLFIL
LIS

LIBFIDNAM
LIS

MATCHNAME
LIS

OPCMSG
LIS

PLIMSG
LIS