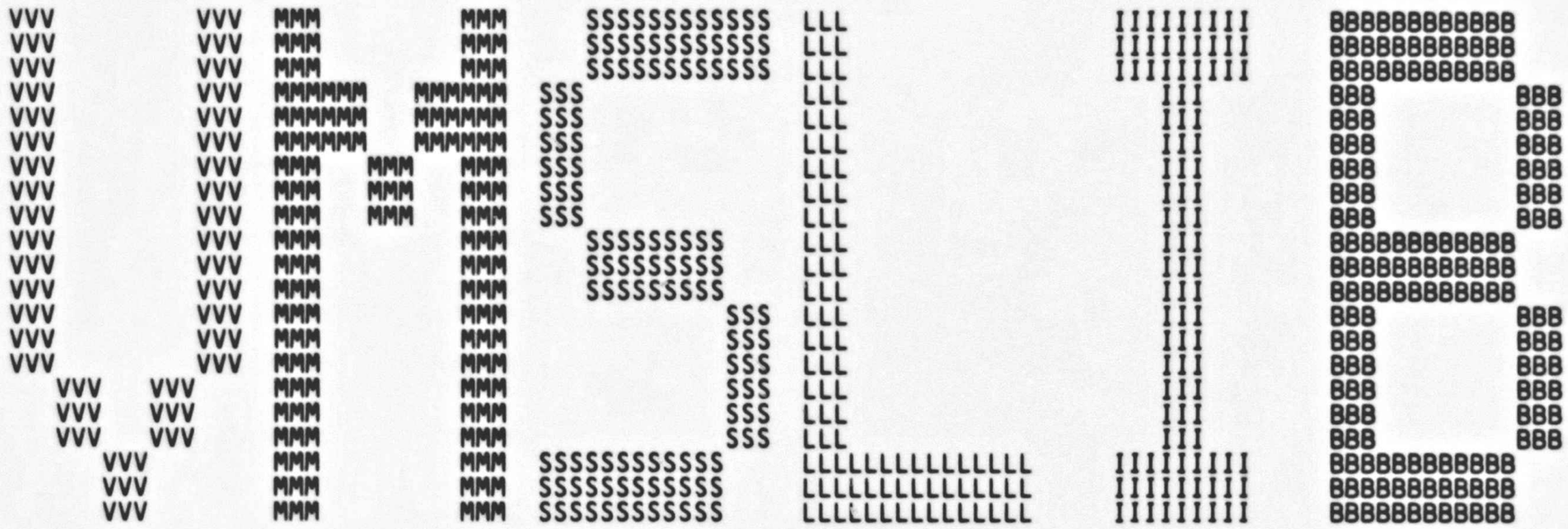
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMM MMM 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**CALBYNAME

```
  CCCCCCCC    AAAAAA    LL          BBBBBBBB    YY      YY  NN      NN    AAAAAA    MM      MM  EEEEEEEEEE
  CCCCCCCC    AAAAAA    LL          BBBBBBBB    YY      YY  NN      NN    AAAAAA    MM      MM  EEEEEEEEEE
CC           AA      AA  LL          BB      BB  YY      YY  NN      NN   AA      AA  MMMM  MMMM  EE
CC           AA      AA  LL          BB      BB  YY      YY  NN      NN   AA      AA  MMMM  MMMM  EE
CC           AA      AA  LL          BB      BB    YY  YY    NNNN    NN   AA      AA  MM  MM  MM  EE
CC           AA      AA  LL          BB      BB    YY  YY    NNNN    NN   AA      AA  MM  MM  MM  EE
CC           AA      AA  LL          BBBBBBBB        YY      NN  NN  NN   AA      AA  MM      MM  EEEEEEEE
CC           AA      AA  LL          BBBBBBBB        YY      NN  NN  NN   AA      AA  MM      MM  EEEEEEEE
CC           AAAAAAAAAA  LL          BB      BB      YY      NN    NNNN   AAAAAAAAAA  MM      MM  EE
CC           AAAAAAAAAA  LL          BB      BB      YY      NN    NNNN   AAAAAAAAAA  MM      MM  EE
CC           AA      AA  LL          BB      BB      YY      NN      NN   AA      AA  MM      MM  EE                ....
  CCCCCCCC   AA      AA  LLLLLLLLLL  BBBBBBBB        YY      NN      NN   AA      AA  MM      MM  EEEEEEEEEE        ....
  CCCCCCCC   AA      AA  LLLLLLLLLL  BBBBBBBB        YY      NN      NN   AA      AA  MM      MM  EEEEEEEEEE        ....

LL             IIIIII      SSSSSSSS
LL             IIIIII    SSSSSSSS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II        SSSSSS
LL               II        SSSSSS
LL               II              SS
LL               II              SS
LL               II              SS
LL               II              SS
LLLLLLLLLL     IIIIII    SSSSSSSS
LLLLLLLLLL     IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE util$call_by_name(                    ! File: CALBYNAME.B32 Edit: BLS0225
;     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  'Dynamically call routine in shareable image by name';
;     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 set of procedures implements dynamic linking to shareable images
;    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: 5-Feb-1983
;    50          0050  1 !
;    51          0051  1 ! MODIFIED BY:
;    52          0052  1 !
;    53          0053  1 !       V03-004 BLS0225         Benn Schreiber          16-Jun-1983
;    54          0054  1 !               Add flags argument to call to util$read_object.  Upcase
;    55          0055  1 !               routine name before looking up.
;    56          0056  1 !
;    57          0057  1 !       V03-003 BLS0222         Benn Schreiber          20-May-1983
```

```
;    58          0058  1 !                 Copy file spec to hdrbuf if error before signalling.
;    59          0059  1 !                 spec goes away if message file image activated
;    60          0060  1 !
;    61          0061  1 !       V03-002 PDG0002         Peter D Gilbert         22-Mar-1983
;    62          0062  1 !                 Remove .ADDRESS reference in UTIL$FIND_SYMBOL
;    63          0063  1 !
;    64          0064  1 !       V03-001 BLS0208         Benn Schreiber          18-Feb-1983
;    65          0065  1 !                 Report full file spec on failed activation if available
;    66          0066  1 !--
```

```
;    68          0067  1 %SBTTL  'Declarations';
;    69          0068  1 !
;    70          0069  1 ! BLISS Libraries
;    71          0070  1 !
;    72          0071  1 LIBRARY
;    73          0072  1     'SYS$LIBRARY:LIB';                  !Need to get IFD definitions from LIB
;    74          0073  1 !
;    75          0074  1 ! Define UTIL$ psects
;    76          0075  1 !
;    77          0076  1 PSECT
;    78          0077  1     CODE = _UTIL$CODE,
;    79          0078  1     GLOBAL = _UTIL$DATA,
;    80          0079  1     OWN = _UTIL$DATA,
;    81          0080  1     PLIT = _UTIL$CODE;
;    82          0081  1
;    83          0082  1 FIELD
;    84          0083  1 !
;    85          0084  1 ! Define the Module information descriptor.  There is one for each image
;    86          0085  1 ! mapped.
;    87          0086  1 !
;    88          0087  1     mid_fields =
;    89          0088  1         SET
;    90          0089  1         mid_l_next =    [0,0,32,0],     !Next in list or 0 if end
;    91          0090  1         mid_l_symlst =  [4,0,32,0],     !Listhead of symbols this image
;    92          0091  1         mid_l_base =    [8,0,32,0],     !Base of image in memory
;    93          0092  1         mid_l_vbn =     [12,0,32,0],    !VBN of GST
;    94          0093  1         mid_b_namlen =  [16,0,8,0],     !Length of name
;    95          0094  1         mid_t_name =    [17,0,0,0]      !Start of image name
;    96          0095  1         TES;
;    97          0096  1 !
;    98          0097  1 ! Define a symbol descriptor
;    99          0098  1 !
;   100          0099  1 FIELD
;   101          0100  1     msd_fields =
;   102          0101  1         SET
;   103          0102  1         msd_l_left =    [0,0,32,0],     !Left subtree pointer
;   104          0103  1         msd_l_right =   [4,0,32,0],     !Right subtree pointer
;   105          0104  1         msd_w_bal =     [8,0,16,0],     !Balance this node
;   106          0105  1         msd_l_value =   [10,0,32,0],    !Value of this symbol
;   107          0106  1         msd_b_namlen =  [14,0,8,0],     !Length of name
;   108          0107  1         msd_t_name =    [15,0,0,0]      !Start of symbol name
;   109          0108  1         TES;
;   110          0109  1
;   111          0110  1 LITERAL
;   112          0111  1     msd_c_size = 15;                    !Length of msd without name
;   113          0112  1
;   114          0113  1 GLOBAL
;   115          0114  1     util$gl_imagelist : REF $BBLOCK FIELD(mid_fields); !Mapped image listhead
;   116          0115  1
;   117          0116  1 EXTERNAL LITERAL
;   118          0117  1     util$m_lnk_1mod : UNSIGNED(8);      !Only read one module
;   119          0118  1
;   120          0119  1 EXTERNAL ROUTINE
;   121          0120  1     lib$get_vm,                         !Allocate virtual memory
;   122          0121  1     lib$free_vm,                        !Deallocate it
;   123          0122  1     lib$insert_tree,                    !Insert element into binary tree
;   124          0123  1     lib$lookup_tree,                    !Lookup element in binary tree
```

```
;   125           0124  1     lib$scopy_r_dx,                   !Dynamic string copy
;   126           0125  1     str$free1_dx,                     !Free dynamic string
;   127           0126  1     str$upcase,                       !Convert string to upper case
;   128           0127  1     util$getfilename,                 !Get filename descr. from fab/nam
;   129           0128  1     util$report_io_error,             !Report I/O error
;   130           0129  1     util$read_object;                 !Read object file
;   131           0130  1 !
;   132           0131  1 ! Define facility-specific names for shared messages
;   133           0132  1 !
;   134         P 0133  1 $SHR_MSGDEF(UTIL,286,LOCAL,
;   135         P 0134  1           (actimage,error),           !Error activating image
;   136         P 0135  1           (insvirmem,error),          !Insufficient virtual memory
;   137         P 0136  1           (openin,error),             !Error opening input file
;   138         P 0137  1           (closein,warning),          !Error closing input file
;   139           0138  1           (readerr,error));           !Read error
```

```
;   141          0139  1 %SBTTL  'get_rec - Read GST record';
;   142          0140  1 ROUTINE get_rec(datavector,recdesc) =
;   143          0141  2 BEGIN
;   144          0142  2 !++
;   145          0143  2 ! FUNCTIONAL DESCRIPTION:
;   146          0144  2 !
;   147          0145  2 !       Read the next record of the GST.  This routine called by
;   148          0146  2 !       UTIL$READ_OBJECT
;   149          0147  2 !
;   150          0148  2 ! INPUTS:
;   151          0149  2 !
;   152          0150  2 !       datavector = address of data vector passed by read_gst
;   153          0151  2 !                    1st longword is current MID block
;   154          0152  2 !                    2nd longword is file RAB
;   155          0153  2 !       recdesc = address of dynamic string descriptor to return record
;   156          0154  2 !
;   157          0155  2 !--
;   158          0156  2 MAP
;   159          0157  2     datavector : REF VECTOR[,LONG];
;   160          0158  2
;   161          0159  2 BIND
;   162          0160  2     midptr = .datavector[0] : $BBLOCK FIELD(mid_fields),
;   163          0161  2     irab = .datavector[1] : $BBLOCK;
;   164          0162  2
;   165          0163  2 LOCAL
;   166          0164  2     status;
;   167          0165  2
;   168          0166  2 !
;   169          0167  2 ! Read the record.  If successful read, copy record to dynamic string
;   170          0168  2 !
;   171          0169  2 status = $GET(RAB=irab,ERR=util$report_io_error);
;   172          0170  2 IF NOT .status
;   173          0171  2     THEN RETURN .status;
;   174          0172  2
;   175          0173  2 lib$scopy_r_dx(irab[rab$w_rsz],.irab[rab$l_rbf],.recdesc);
;   176          0174  2
;   177          0175  2 RETURN ss$_normal
;   178          0176  1 END;


                                                                .TITLE  UTIL$CALL_BY_NAME Dynamically call routine in s
                                                                                          hareable image by
;                                                               .IDENT  \V04-000\

                                                                .PSECT  _UTIL$DATA,NOEXE,2

                                                        00000 UTIL$GL_IMAGELIST::
                                                                .BLKB   4

                                                                .EXTRN  UTIL$M_LNK_1MOD
                                                                .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                                                .EXTRN  LIB$INSERT_TREE
                                                                .EXTRN  LIB$LOOKUP_TREE
                                                                .EXTRN  LIB$SCOPY_R_DX, STR$FREE1_DX
                                                                .EXTRN  STR$UPCASE, UTIL$GETFILENAME
                                                                .EXTRN  UTIL$REPORT_IO_ERROR
                                                                .EXTRN  UTIL$READ_OBJECT
```

```
                                                                .EXTRN  SYS$GET

                                                                .PSECT  _UTIL$CODE,NOWRT,2

                                                0004 00000 GET_REC:.WORD    Save R2                                         ; 0140
                                    50      04  AC   D0 00002          MOVL   DATAVECTOR, R0                              ; 0160
                                    52      04  A0   D0 00006          MOVL   4(R0), R2                                   ; 0161
                                       00000000G 00   9F 0000A          PUSHAB UTIL$REPORT_IO_ERROR                        ; 0169
                                                 52   DD 00010          PUSHL  R2                                          ;
                                00000000G       00   FB 00012          CALLS  #2, SYS$GET                                 ;
                                                13   50 E9 00019          BLBC   STATUS, 1$                                  ; 0170
                                                     08 AC DD 0001C          PUSHL  RECDESC                                 ; 0173
                                                     28 A2 DD 0001F          PUSHL  40(R2)                                  ;
                                                     22 A2 9F 00022          PUSHAB 34(R2)                                  ;
                                00000000G       00   03 FB 00025          CALLS  #3, LIB$SCOPY_R_DX                          ;
                                                50   01 D0 0002C          MOVL   #1, R0                                      ; 0175
                                                     04 0002F 1$:        RET                                                ; 0176

; Routine Size:  48 bytes,    Routine Base:  _UTIL$CODE + 0000
```

```
;  180          0177  1 %SBTTL  'compare_symbols - Compare new symbol to node';
;  181          0178  1 ROUTINE compare_symbols(desc,node) =
;  182          0179  2 BEGIN
;  183          0180  2 !++
;  184          0181  2 ! FUNCTIONAL DESCRIPTION:
;  185          0182  2 !
;  186          0183  2 !       Compare symbol against symbol in current node
;  187          0184  2 !
;  188          0185  2 ! INPUTS:
;  189          0186  2 !
;  190          0187  2 !       desc = address of descriptor of new symbol
;  191          0188  2 !       node = pointer to an MSD block
;  192          0189  2 !
;  193          0190  2 !--
;  194          0191  2 MAP
;  195          0192  2     desc : REF $BBLOCK,
;  196          0193  2     node : REF $BBLOCK FIELD(msd_fields);
;  197          0194  2
;  198          0195  2 LOCAL
;  199          0196  2     retval;
;  200          0197  2
;  201          0198  3 IF (retval = CH$COMPARE(.desc[dsc$w_length],.desc[dsc$a_pointer],
;  202          0199  2                         .node[msd_b_namlen],node[msd_t_name],0)) NEQ 0
;  203          0200  2     THEN RETURN .retval
;  204          0201  2     ELSE RETURN SIGN(.desc[dsc$w_length] - .node[msd_b_namlen])
;  205          0202  1 END;


                                        007C 00000 COMPARE_SYMBOLS:
                                                   .WORD   Save R2,R3,R4,R5,R6                          ; 0178
                          56        04    AC  D0 00002       MOVL    DESC, R6                            ; 0198
                          54        08    AC  D0 00006       MOVL    NODE, R4                            ; 0199
                          50        0E    A4  9A 0000A       MOVZBL  14(R4), R0                          ;
                          55              01  D0 0000E       MOVL    #1, R5                              ;
        50          00        04    B6      66  2D 00011     CMPC5   (R6), @4(R6), #0, R0, 15(R4)        ;
                                    0F    A4      00017      
                                          03  1A 00019       BGTRU   1$                                  ;
                          55              01  D9 0001B       SBWC    #1, R5                              ;
                          50              55  D0 0001E 1$:    MOVL    R5, RETVAL                          ;
                                          15  12 00021       BNEQ    2$                                  ;
                          54        0E    A4  9A 00023       MOVZBL  14(R4), R4                          ; 0201
                          50              66  3C 00027       MOVZWL  (R6), R0                            ;
                          54              50  C2 0002A       SUBL2   R0, R4                              ;
                                          54  D5 0002D       TSTL    R4                                  ;
                                          50  DC 0002F       MOVPSL  R0                                  ;
        50          50            02      02  EF 00031       EXTZV   #2, #2, R0, R0                      ;
                                          50  D7 00036       DECL    R0                                  ;
                                              04 00038 2$:    RET                                         ; 0202

; Routine Size:  57 bytes,    Routine Base:  _UTIL$CODE + 0030
```

```
;   207          0203  1 %SBTTL  'alloc_symbol - Allocate MSD block for new symbol';
;   208          0204  1 ROUTINE alloc_symbol(desc,retadr) =
;   209          0205  2 BEGIN
;   210          0206  2 !++
;   211          0207  2 ! FUNCTIONAL DESCRIPTION:
;   212          0208  2 !
;   213          0209  2 !       Allocate an MSD block for new symbol
;   214          0210  2 !
;   215          0211  2 ! INPUTS:
;   216          0212  2 !
;   217          0213  2 !       desc = address of string descriptor of new symbol
;   218          0214  2 !       retadr = address of longword to return address in
;   219          0215  2 !
;   220          0216  2 !--
;   221          0217  2 MAP
;   222          0218  2     desc : REF $BBLOCK,
;   223          0219  2     retadr : REF VECTOR[,LONG];
;   224          0220  2
;   225          0221  2 LOCAL
;   226          0222  2     status,
;   227          0223  2     msdptr : REF $BBLOCK FIELD (msd_fields);
;   228          0224  2
;   229          0225  2 status = lib$get_vm(%REF(msd_c_size+.desc[dsc$w_length]),msdptr);
;   230          0226  2 IF NOT .status
;   231          0227  3 THEN BEGIN
;   232          0228  3     SIGNAL(util$_insvirmem,0,.status);
;   233          0229  3     RETURN .status
;   234          0230  2     END;
;   235          0231  2
;   236          0232  2 msdptr[msd_l_value] = 0;
;   237          0233  2 msdptr[msd_b_namlen] = .desc[dsc$w_length];
;   238          0234  2 CH$MOVE(.msdptr[msd_b_namlen],.desc[dsc$a_pointer],msdptr[msd_t_name]);
;   239          0235  2 retadr[0] = .msdptr;
;   240          0236  2
;   241          0237  2 RETURN ss$_normal
;   242          0238  1 END;


                                 007C 00000 ALLOC_SYMBOL:
                                            .WORD   Save R2,R3,R4,R5,R6                     ;: 0204
                          5E             08  C2 00002          SUBL2   #8, SP                  ;..
                                   04    AE  9F 00005          PUSHAB  MSDPTR                  ;: 0225
                          52       04    AC  D0 00008          MOVL    DESC, R2                ;..
                    04    AE             62  3C 0000C          MOVZWL  (R2), 4(SP)             ;..
                    04    AE             0F  C0 00010          ADDL2   #15, 4(SP)              ;..
                                   04    AE  9F 00014          PUSHAB  4(SP)                   ;..
              00000000G   00             02  FB 00017          CALLS   #2, LIB$GET_VM          ;..
                          53             50  D0 0001E          MOVL    R0, STATUS              ;..
                          15             53  E8 00021          BLBS    STATUS, 1$              ;: 0226
                                         53  DD 00024          PUSHL   STATUS                  ;: 0228
                                         7E  D4 00026          CLRL    -(SP)                   ;..
                               011E12F2  8F  DD 00028          PUSHL   #18748146               ;..
              00000000G   00             03  FB 0002E          CALLS   #3, LIB$SIGNAL          ;..
                          50             53  D0 00035          MOVL    STATUS, R0              ;: 0229
```

```
                                                    04 00038                RET                                                     ;
                                          56        04    AE  D0 00039 1$:    MOVL    MSDPTR, R6                              ; 0232
                                                    0A    A6  D4 0003D        CLRL    10(R6)                                  ;
                                    0E    A6              62  90 00040        MOVB    (R2), 14(R6)                            ; 0233
                                          50        0E    A6  9A 00044        MOVZBL  14(R6), R0                              ; 0234
                              0F    A6        04    B2      50  28 00048        MOVC3   R0, @4(R2), 15(R6)                      ;
                                    08    BC              56  D0 0004E        MOVL    R6, @RETADR                             ; 0235
                                          50              01  D0 00052        MOVL    #1, R0                                  ; 0237
                                                          04 00055            RET                                             ; 0238

; Routine Size:  86 bytes,    Routine Base:  _UTIL$CODE + 0069
```

```
;   244          0239  1 %SBTTL  'global_routine - Process global symbol from GST';
;   245          0240  1 ROUTINE global_routine(symbol_desc,symbol_value,symbol_flags,
;   246          0241  1                                     entry_mask,datavector,gsd_desc) =
;   247          0242  2 BEGIN
;   248          0243  2 !++
;   249          0244  2 ! FUNCTIONAL DESCRIPTION:
;   250          0245  2 !
;   251          0246  2 !       This routine called with global symbol from shareable image.  A
;   252          0247  2 !       block describing the symbol is allocated and put in the symbols
;   253          0248  2 !       list for this image.
;   254          0249  2 !
;   255          0250  2 !--
;   256          0251  2 MAP
;   257          0252  2     symbol_desc : REF $BBLOCK,
;   258          0253  2     symbol_value : REF VECTOR[,LONG],
;   259          0254  2     symbol_flags : REF VECTOR[,LONG],
;   260          0255  2     datavector : REF VECTOR[,LONG];
;   261          0256  2
;   262          0257  2 BIND
;   263          0258  2     midptr = .datavector[0] : $BBLOCK FIELD(mid_fields);
;   264          0259  2
;   265          0260  2 LOCAL
;   266          0261  2     msdptr : REF $BBLOCK FIELD(msd_fields),
;   267          0262  2     status;
;   268          0263  2
;   269          0264  2 status = lib$insert_tree(midptr[mid_l_symlst],.symbol_desc,%REF(1),
;   270          0265  2                                     compare_symbols,alloc_symbol,msdptr);
;   271          0266  2 IF NOT .status
;   272          0267  2     THEN RETURN .status;
;   273          0268  2 !
;   274          0269  2 ! Set symbol value.  Relocate if needed
;   275          0270  2 !
;   276          0271  2 msdptr[msd_l_value] = .symbol_value[0];
;   277          0272  2 IF (.symbol_flags[0] AND gsy$m_rel) NEQ 0
;   278          0273  2     THEN msdptr[msd_l_value] = .msdptr[msd_l_value] + .midptr[mid_l_base];
;   279          0274  2
;   280          0275  2 RETURN ss$_normal
;   281          0276  1 END;


                                0004 00000 GLOBAL_ROUTINE:
                                           .WORD   Save R2                                 ; 0240
                             5E            08  C2 00002          SUBL2   #8, SP                          ;
                             52       14   BC  D0 00005          MOVL    @DATAVECTOR, R2                 ; 0258
                                  04   AE  9F  9F 00009          PUSHAB  MSDPTR                          ; 0264
                                  9B   AF  9F  9F 0000C          PUSHAB  ALLOC_SYMBOL                    ;
                                FF5E   CF  9F  9F 0000F          PUSHAB  COMPARE_SYMBOLS                 ;
                          0C    AE       01  D0 00013          MOVL    #1, 12(SP)                      ;
                                  0C   AE  9F  9F 00017          PUSHAB  12(SP)                          ;
                                  04   AC  DD  DD 0001A          PUSHL   SYMBOL_DESC                     ;
                                  04   A2  9F  9F 0001D          PUSHAB  4(R2)                           ;
                    00000000G     00       06  FB 00020          CALLS   #6, LIB$INSERT_TREE             ;
                                  16       50  E9 00027          BLBC    STATUS, 2$                      ; 0266
                                  50   04  AE  D0 0002A          MOVL    MSDPTR, R0                      ; 0271
```

```
                          0A   A0       08   BC  D0 0002E          MOVL    @SYMBOL_VALUE, 10(R0)                   ;
                    05    0C   BC            03  E1 00033          BBC     #3, @SYMBOL_FLAGS, 1$                   ; 0272
                          0A   A0       08   A2  C0 00038          ADDL2   8(R2), 10(R0)                           ; 0273
                               50            01  D0 0003D 1$:       MOVL    #1, R0                                  ; 0275
                                             04  00040 2$:       RET                                             ; 0276

; Routine Size:  65 bytes,    Routine Base:  _UTIL$CODE + 00BF
```

```
;   283          0277  1 %SBTTL  'read_gst - Read GST from image';
;   284          0278  1 ROUTINE read_gst (ifdptr,midptr) =
;   285          0279  2 BEGIN
;   286          0280  2 !++
;   287          0281  2 ! FUNCTIONAL DESCRIPTION:
;   288          0282  2 !
;   289          0283  2 !       Read the GST of the image just loaded
;   290          0284  2 !
;   291          0285  2 ! INPUTS:
;   292          0286  2 !
;   293          0287  2 !       ifdptr = address of the image IFD (image file descriptor)
;   294          0288  2 !       midptr = address of MID block for image
;   295          0289  2 !--
;   296          0290  2 MAP
;   297          0291  2     ifdptr : REF $BBLOCK,
;   298          0292  2     midptr : REF $BBLOCK FIELD(mid_fields);
;   299          0293  2
;   300          0294  2 LOCAL
;   301          0295  2     status,
;   302          0296  2     datavector : VECTOR[2,LONG],
;   303          0297  2     ubuf : $BBLOCK[obj$c_maxrecsiz],
;   304          0298  2     ifab : $FAB_DECL,
;   305          0299  2     irab : $RAB_DECL,
;   306          0300  2     inam : $NAM_DECL,
;   307          0301  2     esbuf : $BBLOCK[nam$c_maxrss];
;   308          0302  2
;   309        P 0303  2 $FAB_INIT(FAB=ifab,
;   310        P 0304  2           FNS=.(ifdptr[ifd$q_curprog])<0,16,0>,
;   311        P 0305  2           FNA=.(ifdptr[ifd$q_curprog])<32,32,0>,
;   312        P 0306  2           FAC=(BRO,GET),
;   313        P 0307  2           NAM=inam,
;   314          0308  2           SHR=(GET,PUT,UPI));
;   315          0309  2
;   316        P 0310  2 $NAM_INIT(NAM=inam,
;   317        P 0311  2           RSS=nam$c_maxrss,
;   318        P 0312  2           RSA=esbuf,
;   319        P 0313  2           ESS=nam$c_maxrss,
;   320          0314  2           ESA=esbuf);
;   321          0315  2
;   322          0316  2 ifab[fab$l_ctx] = util$_openin;
;   323          0317  2 status = $OPEN(FAB=ifab,ERR=util$report_io_error);
;   324          0318  2 IF NOT .status
;   325          0319  2     THEN RETURN .status;
;   326          0320  2
;   327        P 0321  2 $RAB_INIT(RAB=irab,
;   328        P 0322  2           FAB=ifab,
;   329        P 0323  2           UBF=ubuf,
;   330        P 0324  2           USZ=obj$c_maxrecsiz,
;   331          0325  2           ROP=LOC);
;   332          0326  2
;   333          0327  2 irab[rab$l_ctx] = util$_openin;
;   334          0328  2 status = $CONNECT(RAB=irab,ERR=util$report_io_error);
;   335          0329  2 IF NOT .status
;   336          0330  3 THEN BEGIN
;   337          0331  3     $CLOSE(FAB=ifab);
;   338          0332  3     RETURN .status
;   339          0333  2     END;
```

```
;   340          0334  2 !
;   341          0335  2 ! Tell RMS that records are variable length, so we can read the GST
;   342          0336  2 !
;   343          0337  2 ifab[fab$v_esc] = 1;
;   344          0338  2 ifab[fab$l_ctx] = rme$c_setrfm;
;   345          0339  2 ifab[fab$b_rfm] = fab$c_var;
;   346          0340  2 status = $MODIFY(FAB=ifab,ERR=util$report_io_error);
;   347          0341  2 IF NOT .status
;   348          0342  3 THEN BEGIN
;   349          0343  3     $CLOSE(FAB=ifab);
;   350          0344  3     RETURN .status
;   351          0345  2     END;
;   352          0346  2 !
;   353          0347  2 ! Point to and read the GST
;   354          0348  2 !
;   355          0349  2 irab[rab$l_rfa0] = .midptr[mid_l_vbn];
;   356          0350  2 irab[rab$w_rfa4] = 0;
;   357          0351  2 irab[rab$b_rac] = rab$c_rfa;
;   358          0352  2 irab[rab$l_ctx] = util$_readerr;
;   359          0353  2 status = $FIND(RAB=irab,ERR=util$report_io_error);
;   360          0354  2 IF NOT .status
;   361          0355  3 THEN BEGIN
;   362          0356  3     $CLOSE(FAB=ifab);
;   363          0357  3     RETURN .status
;   364          0358  2     END;
;   365          0359  2 irab[rab$b_rac] = rab$c_seq;
;   366          0360  2
;   367          0361  2 datavector[0] = .midptr;
;   368          0362  2 datavector[1] = irab;
;   369          0363  2
;   370          0364  2 status = util$read_object(get_rec,%REF(util$m_lnk_1mod),
;   371          0365  2                                 datavector,global_routine);
;   372          0366  2 ifab[fab$l_ctx] = util$_closein;
;   373          0367  2 $CLOSE(FAB=ifab,ERR=util$report_io_error);
;   374          0368  2
;   375          0369  2 RETURN .status
;   376          0370  1 END;


                                                        .EXTRN  SYS$OPEN, SYS$CONNECT
                                                        .EXTRN  SYS$CLOSE, SYS$MODIFY
                                                        .EXTRN  SYS$FIND

                                                01FC 00000 READ_GST:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8                       ; 0278
                                    58 00000000G   00  9E 00002         MOVAB   SYS$CLOSE, R8                       ;
                                    57 00000000G   00  9E 00009         MOVAB   UTIL$REPORT_IO_ERROR, R7            ;
                                    5E      F600   CE  9E 00010         MOVAB   -2560(SP), SP                       ;
    0050   8F          00           6E             00  2C 00015         MOVC5   #0, (SP), #0, #80, $RMS_PTR         ; 0308
                                            01A8   CE     0001C                                                    ;
                              01A8  CE        5003   8F  B0 0001F         MOVW    #20483, $RMS_PTR                    ;
                              01BE  CE        4342   8F  B0 00026         MOVW    #17218, $RMS_PTR+22                 ;
                              01C7  CE               02  90 0002D         MOVB    #2, $RMS_PTR+31                     ;
                              01D0  CE        0104   CE  9E 00032         MOVAB   INAM, $RMS_PTR+40                   ;
                                    50          04   AC  D0 00039         MOVL    IFDPTR, R0                          ;
                              01D4  CE          18   A0  D0 0003D         MOVL    24(R0), $RMS_PTR+44                 ;
```

```
                                01DC   CE        14    A0  90 00043          MOVB    20(R0), $RMS_PTR+52                                 ;
    0060    8F          00              6E              00  2C 00049          MOVC5   #0, (SP), #0, #96, $RMS_PTR                         ; 0314
                                                  0104  CE     00050          ;                                                           ;
                                0104    CE      6002    8F  B0 00053          MOVW    #24578, $RMS_PTR                                    ;
                                0106    CE              01  8E 0005A          MNEGB   #1, $RMS_PTR+2                                      ;
                                0108    CE        04    AE  9E 0005F          MOVAB   ESBUF, $RMS_PTR+4                                   ;
                                010E    CE              01  8E 00065          MNEGB   #1, $RMS_PTR+10                                     ;
                                0110    CE        04    AE  9E 0006A          MOVAB   ESBUF, $RMS_PTR+12                                  ;
                                01C0    CE  011E109A    8F  D0 00070          MOVL    #18747546, IFAB+24                                  ; 0316
                                                        57  DD 00079          PUSHL   R7                                                  ; 0317
                                                  01AC  CE  9F 0007B          PUSHAB  IFAB                                                ;
                            00000000G   00              02  FB 0007F          CALLS   #2, SYS$OPEN                                        ;
                                        56              50  D0 00086          MOVL    R0, STATUS                                          ;
                                        03              56  E8 00089          BLBS    STATUS, 1$                                          ; 0318
                                                      00E2  31 0008C          BRW     4$                                                  ;
    0044    8F          00              6E              00  2C 0008F 1$:      MOVC5   #0, (SP), #0, #68, $RMS_PTR                         ; 0325
                                                  0164  CE     00096          ;                                                           ;
                                0164    CE      4401    8F  B0 00099          MOVW    #17409, $RMS_PTR                                    ;
                                0168    CE  00010000    8F  D0 000A0          MOVL    #65536, $RMS_PTR+4                                  ;
                                0184    CE      0800    8F  B0 000A9          MOVW    #2048, $RMS_PTR+32                                  ;
                                0188    CE      01F8    CE  9E 000B0          MOVAB   UBUF, $RMS_PTR+36                                   ;
                                01A0    CE      01A8    CE  9E 000B7          MOVAB   IFAB, $RMS_PTR+60                                   ;
                                017C    CE  011E109A    8F  D0 000BE          MOVL    #18747546, IRAB+24                                  ; 0327
                                                        57  DD 000C7          PUSHL   R7                                                  ; 0328
                                                  0168  CE  9F 000C9          PUSHAB  IRAB                                                ;
                            00000000G   00              02  FB 000CD          CALLS   #2, SYS$CONNECT                                     ;
                                        56              50  D0 000D4          MOVL    R0, STATUS                                          ;
                                        51              56  E9 000D7          BLBC    STATUS, 2$                                          ; 0329
                                01AF    CE              08  88 000DA          BISB2   #8, IFAB+7                                          ; 0337
                                01C0    CE              01  D0 000DF          MOVL    #1, IFAB+24                                         ; 0338
                                01C7    CE              02  90 000E4          MOVB    #2, IFAB+31                                         ; 0339
                                                        57  DD 000E9          PUSHL   R7                                                  ; 0340
                                                  01AC  CE  9F 000EB          PUSHAB  IFAB                                                ;
                            00000000G   00              02  FB 000EF          CALLS   #2, SYS$MODIFY                                      ;
                                        56              50  D0 000F6          MOVL    R0, STATUS                                          ;
                                        2F              56  E9 000F9          BLBC    STATUS, 2$                                          ; 0341
                                        52        08    AC  D0 000FC          MOVL    MIDPTR, R2                                          ; 0349
                                0174    CE        0C    A2  D0 00100          MOVL    12(R2), IRAB+16                                     ;
                                                  0178  CE  B4 00106          CLRW    IRAB+20                                             ; 0350
                                0182    CE              02  90 0010A          MOVB    #2, IRAB+30                                         ; 0351
                                017C    CE  011E10B2    8F  D0 0010F          MOVL    #18747570, IRAB+24                                  ; 0352
                                                        57  DD 00118          PUSHL   R7                                                  ; 0353
                                                  0168  CE  9F 0011A          PUSHAB  IRAB                                                ;
                            00000000G   00              02  FB 0011E          CALLS   #2, SYS$FIND                                        ;
                                        56              50  D0 00125          MOVL    R0, STATUS                                          ;
                                        09              56  E8 00128          BLBS    STATUS, 3$                                          ; 0354
                                                  01A8  CE  9F 0012B 2$:      PUSHAB  IFAB                                                ; 0356
                                        68              01  FB 0012F          CALLS   #1, SYS$CLOSE                                       ;
                                                        3D  11 00132          BRB     4$                                                  ; 0357
                                                  0182  CE  94 00134 3$:      CLRB    IRAB+30                                             ; 0359
                                    F8  AD              52  D0 00138          MOVL    R2, DATAVECTOR                                      ; 0361
                                    FC  AD        0164  CE  9E 0013C          MOVAB   IRAB, DATAVECTOR+4                                  ; 0362
                                          FE79    CF  9F 00142          PUSHAB  GLOBAL_ROUTINE                                      ; 0364
                                            F8    AD  9F 00146          PUSHAB  DATAVECTOR                                          ;
                                    08  AE        00G   8F  9A 00149          MOVZBL  #UTIL$M_LNK_1MOD, 8(SP)                             ;
                                                    08  AE  9F 0014E          PUSHAB  8(SP)                                               ;
                                          FDAB    CF  9F 00151          PUSHAB  GET_REC                                             ;
```

```
                        00000000G  00              04  FB 00155          CALLS   #4, UTIL$READ_OBJECT                    ;
                                   56              50  D0 0015C          MOVL    R0, STATUS                             ;
                            01C0   CE 011E1050     8F  D0 0015F          MOVL    #18747472, IFAB+24                     ; 0366
                                                   57  DD 00168          PUSHL   R7                                     ; 0367
                                           01AC    CE  9F 0016A          PUSHAB  IFAB                                   ;
                                   68              02  FB 0016E          CALLS   #2, SYS$CLOSE                          ;
                                   50              56  D0 00171 4$:      MOVL    STATUS, R0                             ; 0369
                                                       04 00174          RET                                            ; 0370

; Routine Size:  373 bytes,    Routine Base:  _UTIL$CODE + 0100
```

```
;  378          0371  1 %SBTTL  'find_image - Find image in mapped images list';
;  379          0372  1 ROUTINE find_image(desc,midptr) =
;  380          0373  2 BEGIN
;  381          0374  2 !++
;  382          0375  2 ! FUNCTIONAL DESCRIPTION:
;  383          0376  2 !
;  384          0377  2 !       Look up image in mapped image list
;  385          0378  2 !
;  386          0379  2 ! INPUTS:
;  387          0380  2 !
;  388          0381  2 !       desc = address of descriptor of image name
;  389          0382  2 !       midptr = address of longword to store MID block address if found
;  390          0383  2 !--
;  391          0384  2 MAP
;  392          0385  2     desc : REF $BBLOCK,
;  393          0386  2     midptr : REF VECTOR[,LONG];
;  394          0387  2
;  395          0388  2 LOCAL
;  396          0389  2     ptr : REF $BBLOCK FIELD(mid_fields);
;  397          0390  2
;  398          0391  2 ptr = util$gl_imagelist;
;  399          0392  2
;  400          0393  2 WHILE (ptr = .ptr[mid_l_next]) NEQ 0
;  401          0394  2 DO IF CH$EQL(.desc[dsc$w_length],.desc[dsc$a_pointer],
;  402          0395  2                 .ptr[mid_b_namlen],ptr[mid_t_name],0)
;  403          0396  3 THEN BEGIN
;  404          0397  3     midptr[0] = .ptr;
;  405          0398  3     RETURN ss$_normal
;  406          0399  2     END;
;  407          0400  2
;  408          0401  2 RETURN 0
;  409          0402  1 END;


                                        003C 00000 FIND_IMAGE:
                                                   .WORD   Save R2,R3,R4,R5                                    ; 0372
                          54 00000000'  00  9E 00002       MOVAB   UTIL$GL_IMAGELIST, PTR                      ;; 0391
                          55        04  AC  D0 00009       MOVL    DESC, R5                                    ;; 0394
                          54            64  D0 0000D 1$:   MOVL    (PTR), PTR                                  ;; 0393
                                        17  13 00010       BEQL    2$                                          ;;
                          50        10  A4  9A 00012       MOVZBL  16(PTR), R0                                 ;; 0395
          50      00      04  B5    04  BC  2D 00016       CMPC5   @DESC, @4(R5), #0, R0, 17(PTR)              ;;
                                    11  A4      0001D                                                          ;;
                                        EC  12 0001F       BNEQ    1$                                          ;;
                              08  BC    54  D0 00021       MOVL    PTR, @MIDPTR                                ;; 0397
                                  50    01  D0 00025       MOVL    #1, R0                                      ;; 0398
                                        04 00028           RET                                                 ;;
                                        50  D4 00029 2$:   CLRL    R0                                          ;; 0401
                                        04 0002B           RET                                                 ;; 0402

; Routine Size:  44 bytes,    Routine Base:  _UTIL$CODE + 0275
```

```
;  411          0403  1 %SBTTL  'util$find_symbol - Find symbol by string name';
;  412          0404  1 GLOBAL ROUTINE util$find_symbol (image_desc,routine_desc,retadr) =
;  413          0405  2 BEGIN
;  414          0406  2 !++
;  415          0407  2 ! FUNCTIONAL DESCRIPTION:
;  416          0408  2 !
;  417          0409  2 !       Return the address of the named routine.
;  418          0410  2 !
;  419          0411  2 ! INPUTS:
;  420          0412  2 !
;  421          0413  2 !       image_desc = Address of string descriptor for image name.  If 0,
;  422          0414  2 !               the system shareable image library will be searched
;  423          0415  2 !       routine_desc = Address of string descriptor of symbol to find
;  424          0416  2 !       retadr = Address to return symbol value in
;  425          0417  2 !
;  426          0418  2 ! OUTPUTS:
;  427          0419  2 !
;  428          0420  2 !       If the image can be found and the symbol is found in the image, the
;  429          0421  2 !       relocated (if needed) symbol value will be returned in the longword
;  430          0422  2 !       pointed to by retadr.
;  431          0423  2 !
;  432          0424  2 ! ROUTINE VALUE:
;  433          0425  2 !
;  434          0426  2 !       ss$_normal = found, value returned ok
;  435          0427  2 !       any errors from services called
;  436          0428  2 !--
;  437          0429  2 MAP
;  438          0430  2     image_desc : REF $BBLOCK,
;  439          0431  2     routine_desc : REF $BBLOCK,
;  440          0432  2     retadr : REF VECTOR[,LONG];
;  441          0433  2
;  442          0434  2 LOCAL
;  443          0435  2     status,
;  444          0436  2     tempvec : REF VECTOR[,LONG],
;  445          0437  2     d_desc : $BBLOCK[dsc$c_s_bln],
;  446          0438  2     desc : VECTOR[2,LONG],
;  447          0439  2     ptr : REF $BBLOCK,
;  448          0440  2     ifdptr : REF $BBLOCK,
;  449          0441  2     midptr : REF $BBLOCK FIELD(mid_fields),
;  450          0442  2     msdptr : REF $BBLOCK FIELD (msd_fields),
;  451          0443  2     dflnam : VECTOR[2,LONG],
;  452          0444  2     adrvec : VECTOR[2,LONG];
;  453          0445  2
;  454          0446  2 IF NOT find_image(.image_desc,midptr)
;  455          0447  3 THEN BEGIN
;  456          0448  3     !
;  457          0449  3     ! Image not yet mapped.  Attempt to map it and then read in the GST
;  458          0450  3     !
;  459          0451  4     IF NOT (status = lib$get_vm(%REF(512),midptr))
;  460          0452  4     THEN BEGIN
;  461          0453  4         SIGNAL(util$_insvirmem,0,.status);
;  462          0454  4         RETURN .status
;  463          0455  3         END;
;  464          0456  3
;  465          0457  3     adrvec[0] = 1;
;  466          0458  3     adrvec[1] = 0;
;  467          0459  3     dflnam[0] = %CHARCOUNT('SYS$SHARE:.EXE');
```

```
;  468          0460  3     dflnam[1] = UPLIT BYTE('SYS$SHARE:.EXE');
;  469        P 0461  3     status = $IMGACT(NAME=.image_desc,
;  470        P 0462  3                      DFLNAM=dflnam,
;  471        P 0463  3                      HDRBUF=.midptr,
;  472        P 0464  3                      IMGCTL=iac$m_merge OR iac$m_expreg,
;  473        P 0465  3                      INADR=adrvec,
;  474          0466  3                      RETADR=adrvec);
;  475          0467  3     IF .status
;  476          0468  3         THEN status = $IMGFIX;
;  477          0469  3     IF NOT .status
;  478          0470  4     THEN BEGIN
;  479          0471  4         tempvec = .midptr;
;  480          0472  4         ptr = .tempvec[2];                          !Get FAB address
;  481          0473  4         IF .ptr NEQ 0
;  482          0474  4             THEN CH$MOVE(dsc$c_s_bln,util$getfilename(.ptr),desc);
;  483          0475  4         IF .desc[0] EQL 0
;  484          0476  4             THEN CH$MOVE(dsc$c_s_bln,.image_desc,desc);
;  485          0477  4         CH$MOVE(.desc[0],.desc[1],.midptr);             !Copy the file spec
;  486          0478  4         desc[1] = .midptr;
;  487          0479  4         SIGNAL(util$_actimage,1,desc,.status);
;  488          0480  4         lib$free_vm(%REF(512),midptr);
;  489          0481  4         RETURN .status
;  490          0482  4         END;
;  491          0483  3     !
;  492          0484  3     ! Image is now mapped.  Insert into the mapped image list ad
;  493          0485  3     ! then read the GST
;  494          0486  3     !
;  495          0487  3     tempvec = .midptr;                      !Get IFD address to get filespec
;  496          0488  3     ptr = .tempvec[0];
;  497          0489  3     ptr = .ptr + .ptr[ihd$w_symdbgoff];
;  498          0490  3     ifdptr = .tempvec[1];
;  499          0491  3     midptr[mid_l_next] = .util$gl_imagelist;
;  500          0492  3     util$gl_imagelist = .midptr;
;  501          0493  3     midptr[mid_l_symlst] = 0;
;  502          0494  3     midptr[mid_l_base] = .adrvec[0];
;  503          0495  3     midptr[mid_l_vbn] = .ptr[ihs$l_gstvbn];
;  504          0496  3     midptr[mid_b_namlen] = .image_desc[dsc$w_length];
;  505          0497  3     CH$MOVE(.image_desc[dsc$w_length],.image_desc[dsc$a_pointer],
;  506          0498  3                 midptr[mid_t_name]);
;  507          0499  3
;  508          0500  3     read_gst(.ifdptr,.midptr);
;  509          0501  2     END;
;  510          0502  2 !
;  511          0503  2 ! Image is mapped, lookup and return symbol value
;  512          0504  2 !
;  513          0505  2 d_desc[dsc$w_length] = 0;
;  514          0506  2 d_desc[dsc$b_class] = dsc$k_class_d;
;  515          0507  2 d_desc[dsc$b_dtype] = dsc$k_dtype_t;
;  516          0508  2 d_desc[dsc$a_pointer] = 0;
;  517          0509  2 str$upcase(d_desc,.routine_desc);
;  518          0510  2 status = lib$lookup_tree(midptr[mid_l_symlst],d_desc,
;  519          0511  2                          compare_symbols,msdptr);
;  520          0512  2 str$free1_dx(d_desc);
;  521          0513  2 IF .status
;  522          0514  3 THEN BEGIN
;  523          0515  3     retadr[0] = .msdptr[msd_l_value];
;  524          0516  3     RETURN ss$_normal
```

```
;  525          0517  3     END
;  526          0518  3 ELSE BEGIN
;  527          0519  3     SIGNAL((.status AND NOT sts$m_severity) OR sts$k_error);
;  528          0520  3     retadr[0] = 0;
;  529          0521  3     RETURN .status
;  530          0522  2     END;
;  531          0523  2
;  532          0524  1 END;


    45  58  45  2E  3A  45  52  41  48  53  24  53  59  53  002A1 P.AAA:  .ASCII  \SYS$SHARE:.EXE\                                ;

                                                                        .EXTRN  SYS$IMGACT, SYS$IMGFIX

                                                        0FFC 00000      .ENTRY  UTIL$FIND_SYMBOL, Save R2,R3,R4,R5,R6,R7,-      ; 0404
                                                                                R8,R9,R10,R11
                                                5B      C1   AF  9E 00002      MOVAB   FIND_IMAGE, R11                         ;
                                                5E           2C  C2 00006      SUBL2   #44, SP                                 ;
                                                04      AE   9F  9E 00009      PUSHAB  MIDPTR                                  ; 0446
                                                59      04   AC  D0 0000C      MOVL    IMAGE_DESC, R9                          ;
                                                             59  DD 00010      PUSHL   R9                                      ;
                                                6B           02  FB 00012      CALLS   #2, FIND_IMAGE                          ;
                                                03           50  E9 00015      BLBC    R0, 1$                                  ;
                                                         00FD    31 00018      BRW     7$                                      ;
                                                04      AE   9F  0001B 1$:     PUSHAB  MIDPTR                                  ; 0451
                                        04      AE      0200 8F  3C 0001E      MOVZWL  #512, 4(SP)                             ;
                                                04      AE   9F  00024      PUSHAB  4(SP)                                   ;
                                00000000G       00           02  FB 00027      CALLS   #2, LIB$GET_VM                          ;
                                                5A           50  D0 0002E      MOVL    R0, STATUS                              ;
                                                14           5A  E8 00031      BLBS    STATUS, 2$                              ;
                                                             5A  DD 00034      PUSHL   STATUS                                  ; 0453
                                                             7E  D4 00036      CLRL    -(SP)                                   ;
                                                011E12F2     8F  DD 00038      PUSHL   #18748146                               ;
                                00000000G       00           03  FB 0003E      CALLS   #3, LIB$SIGNAL                          ;
                                                         012D    31 00045      BRW     9$                                      ; 0454
                                        0C      AE           01  7D 00048 2$:     MOVQ    #1, ADRVEC                          ; 0457
                                        14      AE           0E  D0 0004C      MOVL    #14, DFLNAM                             ; 0459
                                        18      AE      9F   AF  9E 00050      MOVAB   P.AAA, DFLNAM+4                         ; 0460
                                                             7E  7C 00055      CLRQ    -(SP)                                   ; 0466
                                                14      AE   9F  9E 00057      PUSHAB  ADRVEC                                  ;
                                                18      AE   9F  9E 0005A      PUSHAB  ADRVEC                                  ;
                                                             30  DD 0005D      PUSHL   #48                                     ;
                                                18      AE   DD  DD 0005F      PUSHL   MIDPTR                                  ;
                                                2C      AE   9F  9E 00062      PUSHAB  DFLNAM                                  ;
                                                             59  DD 00065      PUSHL   R9                                      ;
                                00000000G       00           08  FB 00067      CALLS   #8, SYS$IMGACT                          ;
                                                5A           50  D0 0006E      MOVL    R0, STATUS                              ;
                                                0D           5A  E9 00071      BLBC    STATUS, 3$                              ; 0467
                                00000000G       00           00  FB 00074      CALLS   #0, SYS$IMGFIX                          ; 0468
                                                5A           50  D0 0007B      MOVL    R0, STATUS                              ;
                                                58           5A  E8 0007E      BLBS    STATUS, 6$                              ; 0469
                                                58      04   AE  D0 00081 3$:     MOVL    MIDPTR, TEMPVEC                         ; 0471
                                                57      08   A8  D0 00085      MOVL    8(TEMPVEC), PTR                         ; 0472
                                                             0E  13 00089      BEQL    4$                                      ; 0473
                                                             57  DD 0008B      PUSHL   PTR                                     ; 0474
                                00000000G       00           01  FB 0008D      CALLS   #1, UTIL$GETFILENAME                    ;
```

```
                 1C   AE             60              08  28 00094          MOVC3   #8, (R0), DESC                                  ;
                                                1C   AE  D5 00099 4$:      TSTL    DESC                                            ; 0475
                                                     05  12 0009C          BNEQ    5$                                              ;
                 1C   AE             69              08  28 0009E          MOVC3   #8, (R9), DESC                                  ; 0476
                 04   BE       20    BE        1C    AE  28 000A3 5$:      MOVC3   DESC, @DESC+4, @MIDPTR                          ; 0477
                           20    AE        04    AE  D0 000AA          MOVL    MIDPTR, DESC+4                                  ; 0478
                                                     5A  DD 000AF          PUSHL   STATUS                                          ; 0479
                                                20   AE  9F 000B1          PUSHAB  DESC                                            ;
                                                     01  DD 000B4          PUSHL   #1                                              ;
                                           011E12BA   8F  DD 000B6          PUSHL   #18748090                                       ;
                      00000000G  00                   04  FB 000BC          CALLS   #4, LIB$SIGNAL                                  ;
                                                04   AE  9F 000C3          PUSHAB  MIDPTR                                          ; 0480
                           04    AE        0200    8F  3C 000C6          MOVZWL  #512, 4(SP)                                     ;
                                                04   AE  9F 000CC          PUSHAB  4(SP)                                           ;
                      00000000G  00                   02  FB 000CF          CALLS   #2, LIB$FREE_VM                                 ;
                                                009C   31  000D6          BRW     9$                                              ; 0481
                                 56        04    AE  D0 000D9 6$:      MOVL    MIDPTR, R6                                      ; 0487
                                 58              56  D0 000DD          MOVL    R6, TEMPVEC                                     ;
                                 57              68  7D 000E0          MOVQ    (TEMPVEC), PTR                                  ; 0488
                                 50        04    A7  3C 000E3          MOVZWL  4(PTR), R0                                      ; 0489
                                 57              50  C0 000E7          ADDL2   R0, PTR                                         ;
                                 66  00000000'   00  D0 000EA          MOVL    UTIL$GL_IMAGELIST, (R6)                         ; 0491
                      00000000'  00              56  D0 000F1          MOVL    R6, UTIL$GL_IMAGELIST                           ; 0492
                                                04   A6  D4 000F8          CLRL    4(R6)                                           ; 0493
                           08    A6        0C    AE  D0 000FB          MOVL    ADRVEC, 8(R6)                                   ; 0494
                           0C    A6        04    A7  D0 00100          MOVL    4(PTR), 12(R6)                                  ; 0495
                           10    A6              69  90 00105          MOVB    (R9), 16(R6)                                    ; 0496
                 11   A6        04    B9            69  28 00109          MOVC3   (R9), @4(R9), 17(R6)                            ; 0498
                                                     56  DD 0010F          PUSHL   R6                                              ; 0500
                                                     58  DD 00111          PUSHL   IFDPTR                                          ;
                           FE8B    CB              02  FB 00113          CALLS   #2, READ_GST                                    ;
                             24    AE  020E0000    8F  D0 00118 7$:      MOVL    #34471936, D_DESC                               ; 0505
                                                28   AE  D4 00120          CLRL    D_DESC+4                                        ; 0508
                                                08   AC  DD 00123          PUSHL   ROUTINE_DESC                                    ; 0509
                                                28   AE  9F 00126          PUSHAB  D_DESC                                          ;
                      00000000G  00                   02  FB 00129          CALLS   #2, STR$UPCASE                                  ;
                                                08   AE  9F 00130          PUSHAB  MSDPTR                                          ; 0510
                                              FDBB   CB  9F 00133          PUSHAB  COMPARE_SYMBOLS                                 ;
                                                2C   AE  9F 00137          PUSHAB  D_DESC                                          ;
                 7E         10    AE              04  C1 0013A          ADDL3   #4, MIDPTR, -(SP)                               ;
                      00000000G  00                   04  FB 0013F          CALLS   #4, LIB$LOOKUP_TREE                             ;
                                 5A              50  D0 00146          MOVL    R0, STATUS                                      ;
                                                24   AE  9F 00149          PUSHAB  D_DESC                                          ; 0512
                      00000000G  00                   01  FB 0014C          CALLS   #1, STR$FREE1_DX                                ;
                                 0D              5A  E9 00153          BLBC    STATUS, 8$                                      ; 0513
                                 50        08    AE  D0 00156          MOVL    MSDPTR, R0                                      ; 0515
                           0C    BC        0A    A0  D0 0015A          MOVL    10(R0), @RETADR                                 ;
                                 50              01  D0 0015F          MOVL    #1, R0                                          ; 0518
                                                     04  00162          RET                                                     ;
                 50              5A                  07  CB 00163 8$:      BICL3   #7, STATUS, R0                                  ; 0519
                 7E              50                  02  C9 00167          BISL3   #2, R0, -(SP)                                   ;
                      00000000G  00                   01  FB 0016B          CALLS   #1, LIB$SIGNAL                                  ;
                                                0C   BC  D4 00172          CLRL    @RETADR                                         ; 0520
                                 50              5A  D0 00175 9$:      MOVL    STATUS, R0                                      ; 0521
                                                     04  00178          RET                                                     ; 0524

; Routine Size:  377 bytes,    Routine Base:  _UTIL$CODE + 02AF
```

```
;   534          0525  1 %SBTTL  'util$call_symbol - Call routine by string name';
;   535          0526  1 GLOBAL ROUTINE util$call_symbol(image_desc,routine_desc,arglst) =
;   536          0527  2 BEGIN
;   537          0528  2 !++
;   538          0529  2 ! FUNCTIONAL DESCRIPTION:
;   539          0530  2 !
;   540          0531  2 !       Call a routine by name.  ARGLST argument is the start of the argument
;   541          0532  2 !       list (i.e. the count of arguments), and .ARGLST arguments follow.
;   542          0533  2 !
;   543          0534  2 ! CALLING SEQUENCE:
;   544          0535  2 !
;   545          0536  2 !       CALL UTIL$CALL_SYMBOL(''IMAGE-NAME'',''ROUTINE-NAME'',
;   546          0537  2 !                                               ARGCOUNT,ARG1,ARG2...)
;   547          0538  2 !
;   548          0539  2 !--
;   549          0540  2 BUILTIN
;   550          0541  2     CALLG;
;   551          0542  2
;   552          0543  2 LOCAL
;   553          0544  2     status,
;   554          0545  2     routine_addr;
;   555          0546  2
;   556          0547  3 IF NOT (status = util$find_symbol(.image_desc,.routine_desc,routine_addr))
;   557          0548  2     THEN RETURN .status;
;   558          0549  2
;   559          0550  2 RETURN CALLG(arglst,.routine_addr)
;   560          0551  1 END;


                                0000 00000         .ENTRY  UTIL$CALL_SYMBOL, Save nothing                  ; 0526
                          5E    04   C2 00002         SUBL2   #4, SP                                          ;
                          5E   DD 00005         PUSHL   SP                                              ; 0547
                       7E 04   AC   7D 00007         MOVQ    IMAGE_DESC, -(SP)                               ;
                    FE77 CF    03   FB 0000B         CALLS   #3, UTIL$FIND_SYMBOL                            ;
                       05      50   E9 00010         BLBC    STATUS, 1$                                      ;
                    00 BE   0C AC   FA 00013         CALLG   ARGLST, @ROUTINE_ADDR                           ; 0550
                                04 00018 1$:         RET                                                     ; 0551

; Routine Size:  25 bytes,    Routine Base:  _UTIL$CODE + 0428
```

```
;  562          0552  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                           1089  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                     Library Statistics
;
;                                              -------- Symbols --------      Pages       Processing
;       File                                   Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1                18619       121         0        1000        00:01.9



;                                       COMMAND QUALIFIERS

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

; Size:          1075 code + 18 data bytes
; Run Time:        00:22.0
; Elapsed Time:    00:45.0
; Lines/CPU Min:    1506
; Lexemes/CPU-Min: 28504
; Memory Used:  180 pages
; Compilation Complete
```

CLIMAC
REQ

TPAMAC
REQ

UTLDEFB
B32

STARMISC
MAR

EQDEFM
MAR

TPAMAR
MAR

CALBYNAME
LIS

SCRPROLOG
REQ

STARLET
SDL

SCRTCB
REQ

OFFSET
MAR

SYITABLE
MAR

EQDEFB
B32

JPITABLE
MAR

DVITABLE
MAR

UTLDEFM
MAR

B32MSG
LIS

SCRTERM
REQ

RSXLBLDF
MAR

C74MSG
LIS

LIBACP
LIS

CALLIMAGE
LIS

DISMOUMSG
LIS

LIBCLICAL
LIS

LIBCOMAT
LIS

FILEDFLT
LIS

LIBASSIGN
LIS

CORMSG
LIS

CVTATB
LIS

LIBCQUAL
LIS

INITMSG
LIS

CREMAPSEC
LIS