```
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG   GGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG   GGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG   GGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
```

```
**FILE**ID**DBGLEVEL3                              J  6
```

```
DDDDDDDD    BBBBBBBB      GGGGGGGG  LL          EEEEEEEEEE  VV      VV  EEEEEEEEEE  LL            33333
DDDDDDDD    BBBBBBBB      GGGGGGGG  LL          EEEEEEEEEE  VV      VV  EEEEEEEEEE  LL            333333
DD       DD BB      BB  GG          LL          EE          VV      VV  EE          LL          33      33
DD       DD BB      BB  GG          LL          EE          VV      VV  EE          LL          33      33
DD       DD BB      BB  GG          LL          EE          VV      VV  EE          LL                  33
DD       DD BB      BB  GG          LL          EE          VV      VV  EE          LL                  33
DD       DD BBBBBBBB    GG          LL          EEEEEEEE    VV      VV  EEEEEEEE    LL                33
DD       DD BBBBBBBB    GG          LL          EEEEEEEE    VV      VV  EEEEEEEE    LL                33
DD       DD BB      BB  GG  GGGGGG  LL          EE          VV      VV  EE          LL                  33
DD       DD BB      BB  GG  GGGGGG  LL          EE          VV      VV  EE          LL                  33
DD       DD BB      BB  GG      GG  LL          EE            VV  VV    EE          LL          33      33      ....
DD       DD BB      BB  GG      GG  LL          EE            VV  VV    EE          LL          33      33      ....
DDDDDDDD    BBBBBBBB      GGGGGG    LLLLLLLLLL  EEEEEEEEEE      VV      EEEEEEEEEE  LLLLLLLLLL    333333        ....
DDDDDDDD    BBBBBBBB      GGGGGG    LLLLLLLLLL  EEEEEEEEEE      VV      EEEEEEEEEE  LLLLLLLLLL    333333        ....

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
```

```
0 MODULE DBGLEVEL3 (IDENT = 'V04-000') =
0
1 BEGIN
1
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1 ! WRITTEN BY
1 !       John Francis    August, 1982
1 !
1 ! MODULE FUNCTION
1 !       This module contains the DEBUG kernel code for performing the
1 !       EVALUATE, EXAMINE and DEPOSIT commands.
1 !
1
1 REQUIRE 'SRC$:DBGPROLOG.REQ';
1
1 LIBRARY 'LIB$:DBGGEN.L32';
1
1 FORWARD ROUTINE
1     DBG$COLLECT: NOVALUE,
1     DEPOSIT_HANDLER,
1     DBG$DEPOSIT: NOVALUE,
1     DBG$EVALUATE: NOVALUE,
1     DBG$EXAMINE: NOVALUE,
1     DBG$NEXTLOC,
1     DBG$PREVLOC,
1     MODIFY_PRIMARY,
1     PRIMARY_ORDER,
1     CHECK_TEXT_DESCRIPTOR,
1     FIX_UP_LENGTH;
```

```
EXTERNAL
    DBG$GL_CURRENT_PRIMARY,                 ! Pointer to the primary being processed
    DBG$REG_VALUES: VECTOR[,LONG],          ! Vector of user register values in the
    DBG$GL_CURLOC_VMSDESC,                  ! Override type for %CURLOC
    DBG$GL_DEPOSIT_TOKEN,                   ! Assignment operator token
    DBG$GL_IDENTITY_TOKEN,                  ! Identity operator token
    DBG$GL_DFLTTYP,                         ! Default type from "SET TYPE"
    DBG$GW_DFLTLENG: WORD,                  ! Length of default data-type
    DBG$GL_SIGN_FLAG;                       ! Print '+' before signed variable

EXTERNAL ROUTINE
    DBG$BUILD_PRIMARY_SUBNODE: NOVALUE,     ! Add new primary sub-node
    DBG$DATA_LENGTH,                        ! Get length of a data-item
    DBG$DO_MAPPING,                         ! Perform "type mapping"
    DBG$EVAL_LANG_OPERATOR,                 ! Evaluate operator expressions
    DBG$FLUSHBUF: NOVALUE,                  ! Initialize a new print line
    DBG$GET_TEMPMEM,                        ! Allocate temporary memory
    DBG$IS_IT_ENTRY,                        ! Check for CALL entry-mask address
    DBG$INS_DECODE,                         ! Get length of instruction
    DBG$MAKE_VAL_DESC,                      ! Construct value descriptor
    DBG$NGET_PAGES,                         ! Construct page list
    DBG$PC_TO_LINE_LOOKUP,                  ! Get line & statement number
    DBG$PC_TO_SYMID,                        ! Look up address in SAT
    DBG$PRIM_TO_VAL,                        ! Obtain value of data-item
    DBG$PRINT: NOVALUE,                     ! Formats an output line.
    DBG$PRINT_AGGREGATE : NOVALUE,          ! Print array or record
    DBG$PRINT_FIELD_REF : NOVALUE,          ! Print <p,s,e> information
    DBG$PRINT_IDENTIFIER,                   ! Print name of data-item
    DBG$PRINT_VALUE: NOVALUE,               ! Print value in a given radix
    DBG$PRINT_VALUE_AS_INTEGER: NOVALUE,    ! Print absolute address
    DBG$PUSH_TEMPMEM,                       ! Save temporary memory state
    DBG$POP_TEMPMEM: NOVALUE,               ! Restore   "        "      "
    DBG$NEWLINE: NOVALUE,                   ! Outputs the output buffer.
    DBG$SAVE_LOC: NOVALUE,                  ! Save dot
    DBG$SAVE_VAL: NOVALUE,                  ! Save backslash
    DBG$SET_PAGE_PROT,                      ! Set page protections
    DBG$SRC_TYPE_PC_SOURCE: NOVALUE,        ! Type source text
    DBG$STA_ADDRESS_TO_REGDESCR,            ! Translate address to reg descr
    DBG$STA_REGISTER_NAME,                  ! Obtain reg name from reg descr
    DBG$STA_SETREGISTERS: NOVALUE,          ! Store context register values
    DBG$STA_SETCONTEXT: NOVALUE,            ! Set up context correctly
    DBG$STA_SYMKIND: NOVALUE,               ! Get KIND of data item
    DBG$STA_SYMNAME: NOVALUE,               ! Get NAME of data item
    DBG$STA_SYMSIZE: NOVALUE,               ! Get SIZE of data item
    DBG$STA_SYMTYPE: NOVALUE,               ! Get TYPE of data item
    DBG$STA_TYP_RECORD: NOVALUE,            ! Get symbol table information
    DBG$STA_VARIANT_SELECT,                 ! Get entry from variant set
    DBG$TYPEID_FOR_ATOMIC,                  ! Make dummy RST entry
    DBG$UPDATE_WATCHPOINTS: NOVALUE,        ! Update watched values after DEPOSIT
    LIB$SIGNAL;                             ! Signal an error

LITERAL

    ! Verb codes for the EVALUATE command.
    !
    EVALUATE                    = 1,        ! EVALUATE verb code
    EVALUATE_ADDR               = 2,        ! EVALUATE/ADDRESS verb code
```

```
1      EVALUATE_COND               = 3,    ! EVALUATE/CONDITION verb code
1
1
1
1      ! Verb codes for the EXAMINE command.
1      !
1      EXAMINE                     = 1,    ! EXAMINE verb code
1      EXAMINE_INSTRUCTION         = 2,    ! EXAMINE/INSTRUCTION verb code
1      EXAMINE_REGISTER            = 3,    ! EXAMINE register verb code
1      EXAMINE_SOURCE              = 4,    ! EXAMINE/SOURCE verb code
1      EXAMINE_CONDITION_VALUE     = 5,    ! EXAMINE/CONDITION verb code
1      EXAMINE_PSL                 = 6,    ! EXAMINE the PSL verb code
1      EXAMINE_PSW                 = 7;    ! EXAMINE the PSW verb code
1
1  OWN
1      PAGE_LIST;                          ! Pointer to list of pages whose protec-
1                                          !       tion we may have changed
```

```
1 GLOBAL ROUTINE DBG$CHANGE_DTYPE(PRM_DESC, NEW_TYPE, NEW_SIZE) =
1 !
1 ! FUNCTION
1 !          -----------------------------
1 !
1 ! INPUTS
1 !          -----------------------------
1 !
1 ! OUTPUTS
1 !          -----------------------------
1 !
1
2     BEGIN
2
2     MAP
2         PRM_DESC: REF DBG$PRIMARY;        ! Pointer to Primary Descriptor
2
2     LOCAL
2         ADDR,                             ! The current instruction address
2         SIZE,                             ! Size of V-Value Descriptor header
2         VAL_DESC: REF DBG$VALDESC;        ! Pointer to Value Descriptor
2
2
2
2     ! Determine what kind of descriptor we have.
2     !
2     SELECTONE .PRM_DESC[DBG$B_DHDR_TYPE] OF
2         SET
2
2
2         ! Handle Primary Descriptors.
2         !
2         [DBG$K_PRIMARY_DESC]:
2             DBG$PRIM_TO_VAL(.PRM_DESC,DBG$K_V_VALUE_DESC,VAL_DESC);
2
2
2         ! Handle Volatile Value Descriptors.
2         !
2         [DBG$K_V_VALUE_DESC]:
3             BEGIN
3             SIZE = .PRM_DESC[DBG$W_DHDR_LENGTH];
3             VAL_DESC = DBG$GET_TEMPMEM((.SIZE + (%UPVAL - 1))/%UPVAL);
3             CH$MOVE(.SIZE,.PRM_DESC,.VAL_DESC);
3             IF .PRM_DESC[DBG$L_DHDR_SYMID0] NEQ 0
3             THEN
3                 DBG$STA_SETCONTEXT(.PRM_DESC[DBG$L_DHDR_SYMID0]);
3
2             END;
2
2
2         ! Any other descriptor type is invalid so we signal an internal
2         ! DEBUG error.
2         !
2         [OTHERWISE]:
2             SIGNAL(DBG$_ILLTYPE);
2
2         TES;
```

```

! If the type is DBG$K_NOTYPE, meaning type instruction, we return now.
!
IF .NEW_TYPE EQL DBG$K_NOTYPE THEN RETURN .VAL_DESC;


! If we get here then we are overriding the type information.  In this
! case, set the FCODE to "descriptor".  Also set the "override" flag.
!
VAL_DESC[DBG$B_DHDR_FCODE] = RST$K_TYPE_DESCR;
VAL_DESC[DBG$V_DHDR_OVERRIDE] = TRUE;
SELECTONE .NEW_TYPE OF
    SET


    ! Handle the /ASCIZ, /ASCIC, and /ASCIW qualifiers.  These refer to the
    ! zero-terminated and counted ASCII string types.
    !
    [DSC$K_DTYPE_AZ,
     DSC$K_DTYPE_AC,
     DSC$K_DTYPE_VT]:
        BEGIN
        IF (.VAL_DESC[DBG$B_VALUE_CLASS] EQL DSC$K_CLASS_UBS)
        THEN
            SIGNAL(DBG$_UNALIGNED);

        VAL_DESC[DBG$B_VALUE_CLASS]  = DSC$K_CLASS_VS;
        VAL_DESC[DBG$B_VALUE_DTYPE]  = .NEW_TYPE;
        VAL_DESC[DBG$W_VALUE_LENGTH] =
                        FIX_UP_LENGTH(VAL_DESC[DBG$A_VALUE_VMSDESC]);
        END;


    ! Handle the /ASCID qualifier (ASCII string via its descriptor).
    !
    [DBG$K_DTYPE_AD]:
        BEGIN
        IF NOT CHECK_TEXT_DESCRIPTOR(.VAL_DESC)
        THEN
            SIGNAL(DBG$_DESCNOTSET);

        END;


    ! Handle the plain ASCII text string data type (the /ASCII qualifier).
    !
    [DSC$K_DTYPE_T]:
        BEGIN
        IF .NEW_SIZE NEQ 0
        THEN
            VAL_DESC[DBG$W_VALUE_LENGTH] = .NEW_SIZE

        ELSE
            VAL_DESC[DBG$W_VALUE_LENGTH] = DBG$DATA_LENGTH(
                        VAL_DESC[DBG$A_VALUE_VMSDESC])/%BPUNIT;

```

```
3            VAL_DESC[DBG$B_VALUE_CLASS]  = DSC$K_CLASS_Z;
3            VAL_DESC[DBG$B_VALUE_DTYPE]  = DSC$K_DTYPE_T;
2            END;
2
2
2        ! Handle the /INSTRUCTION qualifier.
2        !
2        [DSC$K_DTYPE_ZI]:
3            BEGIN
3            VAL_DESC[DBG$B_VALUE_CLASS]  = DSC$K_CLASS_Z;
3            ADDR = .VAL_DESC[DBG$L_VALUE_POINTER];
3            IF DBG$IS_IT_ENTRY(.ADDR)
3            THEN
4                BEGIN
4                VAL_DESC[DBG$B_VALUE_DTYPE]  = DSC$K_DTYPE_ZEM;
4                VAL_DESC[DBG$W_VALUE_LENGTH] = 2;
4                END
4
3            ELSE
4                BEGIN
4                VAL_DESC[DBG$B_VALUE_DTYPE]  = DSC$K_DTYPE_ZI;
4                VAL_DESC[DBG$W_VALUE_LENGTH] =
4                                DBG$INS_DECODE(.ADDR, FALSE, FALSE) - .ADDR;
3                END;
3
2            END;
2
2
2        ! Handle the /PACKED qualifier.
2        !
2        [DSC$K_DTYPE_P]:
3            BEGIN
3            VAL_DESC[DBG$B_VALUE_CLASS]  = DSC$K_CLASS_Z;
3            VAL_DESC[DBG$B_VALUE_DTYPE]  = .NEW_TYPE;
3            IF .NEW_SIZE NEQ %X'0000FFFF'
3            THEN
3                VAL_DESC[DBG$W_VALUE_LENGTH] = .NEW_SIZE
3
3            ELSE
4                BEGIN
4
4                FIELD
4                    PACKED_FIELDS =
4                        SET
4                        SIGN_NIBBLE = [0,0,4,0]
4                        TES;
4                BIND
4                    PACKED_DATA = .VAL_DESC[DBG$L_VALUE_POINTER]:
4                                BLOCKVECTOR[16,1,BYTE] FIELD(PACKED_FIELDS);
4
4                INCR I FROM 0 TO 15 DO
5                    BEGIN
5                    IF .PACKED_DATA[.I, SIGN_NIBBLE] GTR 9
5                    THEN
6                        BEGIN
6                        VAL_DESC[DBG$W_VALUE_LENGTH] = (.I*2) + 1;
6                        EXITLOOP;
```

```
;  299        0430  5                     END;
;  300        0431  5
;  301        0432  4                 END;
;  302        0433  4
;  303        0434  3             END;
;  304        0435  3
;  305        0436  2         END;
;  306        0437  2
;  307        0438  2
;  308        0439  2     ! Handle any other data type.
;  309        0440  2     !
;  310        0441  2     [OTHERWISE]:
;  311        0442  3         BEGIN
;  312        0443  3         VAL_DESC[DBG$B_VALUE_CLASS]  = DSC$K_CLASS_Z;
;  313        0444  3         VAL_DESC[DBG$B_VALUE_DTYPE]  = .NEW_TYPE;
;  314        0445  3         VAL_DESC[DBG$W_VALUE_LENGTH] = .NEW_SIZE;
;  315        0446  2         END;
;  316        0447  2
;  317        0448  2     TES;
;  318        0449  2
;  319        0450  2 RETURN .VAL_DESC;
;  320        0451  1 END;


                                                            .TITLE  DBGLEVEL3
                                                            .IDENT  \V04-000\

                                                            .PSECT  DBG$OWN,NOEXE,  PIC,2

                                            00000 PAGE_LIST:
                                                            .BLKB   4

                                                            .EXTRN  DBG$GL_CURRENT_PRIMARY
                                                            .EXTRN  DBG$REG_VALUES, DBG$GL_CURLOC_VMSDESC
                                                            .EXTRN  DBG$GL_DEPOSIT_TOKEN
                                                            .EXTRN  DBG$GL_IDENTITY_TOKEN
                                                            .EXTRN  DBG$GL_DFLTTYP, DBG$GW_DFLTLENG
                                                            .EXTRN  DBG$GL_SIGN_FLAG
                                                            .EXTRN  DBG$BUILD_PRIMARY_SUBNODE
                                                            .EXTRN  DBG$DATA_LENGTH
                                                            .EXTRN  DBG$DO_MAPPING, DBG$EVAL_LANG_OPERATOR
                                                            .EXTRN  DBG$FLUSHBUF, DBG$GET_TEMPMEM
                                                            .EXTRN  DBG$IS_IT_ENTRY
                                                            .EXTRN  DBG$INS_DECODE, DBG$MAKE_VAL_DESC
                                                            .EXTRN  DBG$NGET_PAGES, DBG$PC_TO_LINE_LOOKUP
                                                            .EXTRN  DBG$PC_TO_SYMID
                                                            .EXTRN  DBG$PRIM_TO_VAL
                                                            .EXTRN  DBG$PRINT, DBG$PRINT_AGGREGATE
                                                            .EXTRN  DBG$PRINT_FIELD_REF
                                                            .EXTRN  DBG$PRINT_IDENTIFIER
                                                            .EXTRN  DBG$PRINT_VALUE
                                                            .EXTRN  DBG$PRINT_VALUE_AS_INTEGER
                                                            .EXTRN  DBG$PUSH_TEMPMEM
                                                            .EXTRN  DBG$POP_TEMPMEM
                                                            .EXTRN  DBG$NEWLINE, DBG$SAVE_LOC
                                                            .EXTRN  DBG$SAVE_VAL, DBG$SET_PAGE_PROT
                                                            .EXTRN  DBG$SRC_TYPE_PC_SOURCE
```

```
                                                         .EXTRN  DBG$STA_ADDRESS_TO_REGDESCR
                                                         .EXTRN  DBG$STA_REGISTER_NAME
                                                         .EXTRN  DBG$STA_SETREGISTERS
                                                         .EXTRN  DBG$STA_SETCONTEXT
                                                         .EXTRN  DBG$STA_SYMKIND
                                                         .EXTRN  DBG$STA_SYMNAME
                                                         .EXTRN  DBG$STA_SYMSIZE
                                                         .EXTRN  DBG$STA_SYMTYPE
                                                         .EXTRN  DBG$STA_TYP_RECORD
                                                         .EXTRN  DBG$STA_VARIANT_SELECT
                                                         .EXTRN  DBG$TYPEID_FOR_ATOMIC
                                                         .EXTRN  DBG$UPDATE_WATCHPOINTS
                                                         .EXTRN  LIB$SIGNAL

                                                         .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                       00FC 00000        .ENTRY  DBG$CHANGE_DTYPE, Save R2,R3,R4,R5,R6,R7          ; 0259
                   57 00000000G   00   9E 00002          MOVAB   LIB$SIGNAL, R7                                    ;
                   5E             04   C2 00009          SUBL2   #4, SP                                            ;
                   56        04   AC   D0 0000C          MOVL    PRM_DESC, R6                                      ; 0285
             79    8F        02   A6   91 00010          CMPB    2(R6), #121                                       ; 0291
                                  11   12 00015          BNEQ    1$                                                ;
                                  5E   DD 00017          PUSHL   SP                                                ; 0292
                   7E        83   8F   9A 00019          MOVZBL  #131, -(SP)                                       ;
                                  56   DD 0001D          PUSHL   R6                                                ;
      00000000G    00             03   FB 0001F          CALLS   #3, DBG$PRIM_TO_VAL                               ;
                                  3B   11 00026          BRB     3$                                                ;
             83    8F        02   A6   91 00028 1$:      CMPB    2(R6), #131                                       ; 0297
                                  2B   12 0002D          BNEQ    2$                                                ;
                   52             66   3C 0002F          MOVZWL  (R6), SIZE                                        ; 0299
                   50        03   A2   9E 00032          MOVAB   3(R2), R0                                         ; 0300
         7E        50             04   C7 00036          DIVL3   #4, R0, -(SP)                                     ;
      00000000G    00             01   FB 0003A          CALLS   #1, DBG$GET_TEMPMEM                               ;
                   6E             50   D0 00041          MOVL    R0, VAL_DESC                                      ;
  00     BE        66             52   28 00044          MOVC3   SIZE, (R6), @VAL_DESC                             ; 0301
                             0C   A6   D5 00049          TSTL    12(R6)                                            ; 0302
                                  15   13 0004C          BEQL    3$                                                ;
                             0C   A6   DD 0004E          PUSHL   12(R6)                                            ; 0304
      00000000G    00             01   FB 00051          CALLS   #1, DBG$STA_SETCONTEXT                            ;
                                  09   11 00058          BRB     3$                                                ; 0285
                        000287D8  8F   DD 0005A 2$:      PUSHL   #165848                                           ; 0313
                   67             01   FB 00060          CALLS   #1, LIB$SIGNAL                                    ;
                   54        08   AC   D0 00063 3$:      MOVL    NEW_TYPE, R4                                      ; 0320
      00000080     8F             54   D1 00067          CMPL    R4, #128                                          ;
                                  04   12 0006E          BNEQ    4$                                                ;
                   50             6E   D0 00070          MOVL    VAL_DESC, R0                                      ;
                                       04 00073          RET                                                       ;
                   52             6E   D0 00074 4$:      MOVL    VAL_DESC, R2                                      ; 0326
             06    A2             03   90 00077          MOVB    #3, 6(R2)                                         ;
             04    A2        80   8F   88 0007B          BISB2   #128, 4(R2)                                       ; 0327
                   25             54   D1 00080          CMPL    R4, #37                                           ; 0335
                                  2C   19 00083          BLSS    6$                                                ;
                   27             54   D1 00085          CMPL    R4, #39                                           ;
                                  27   14 00088          BGTR    6$                                                ;
                   53        14   A2   9E 0008A          MOVAB   20(R2), R3                                        ; 0339
                   0D        03   A3   91 0008E          CMPB    3(R3), #13                                        ;
                                  09   12 00092          BNEQ    5$                                                ;
```

```
                                   00028D08   8F    DD 00094          PUSHL   #167176                          ; 0341
                            67                01    FB 0009A          CALLS   #1, LIB$SIGNAL                   ;
                   03       A3                0B    90 0009D  5$:     MOVB    #11, 3(R3)                       ; 0343
                   02       A3                54    90 000A1          MOVB    R4, 2(R3)                        ; 0344
                                              53    DD 000A5          PUSHL   R3                               ; 0346
               0000V        CF                01    FB 000A7          CALLS   #1, FIX_UP_LENGTH                ;
                            63                50    B0 000AC          MOVW    R0, (R3)                         ;
                                              7C    11 000AF          BRB     12$                              ; 0328
                            38                54    D1 000B1  6$:     CMPL    R4, #56                          ; 0352
                                              15    12 000B4          BNEQ    7$                               ;
                                              52    DD 000B6          PUSHL   R2                               ; 0354
               0000V        CF                01    FB 000B8          CALLS   #1, CHECK_TEXT_DESCRIPTOR        ;
                            6D                50    E8 000BD          BLBS    R0, 12$                          ;
                                   00028F50   8F    DD 000C0          PUSHL   #167760                          ; 0356
                            67                01    FB 000C6          CALLS   #1, LIB$SIGNAL                   ;
                                              62    11 000C9          BRB     12$                              ; 0328
                            0E                54    D1 000CB  7$:     CMPL    R4, #14                          ; 0363
                                              25    12 000CE          BNEQ    10$                              ;
                            53         14     A2    9E 000D0          MOVAB   20(R2), R3                       ; 0367
                                       0C     AC    D5 000D4          TSTL    NEW_SIZE                         ; 0365
                                              06    13 000D7          BEQL    8$                               ;
                            63         0C     AC    B0 000D9          MOVW    NEW_SIZE, (R3)                   ; 0367
                                              10    11 000DD          BRB     9$                               ;
                                              53    DD 000DF  8$:     PUSHL   R3                               ; 0371
           00000000G        00                01    FB 000E1          CALLS   #1, DBG$DATA_LENGTH              ;
                   51       50                08    C7 000E8          DIVL3   #8, R0, R1                       ;
                            63                51    B0 000EC          MOVW    R1, (R3)                         ;
                   02       A3                0E    B0 000EF  9$:     MOVW    #14, 2(R3)                       ; 0374
                                              72    11 000F3          BRB     18$                              ; 0328
                            16                54    D1 000F5  10$:    CMPL    R4, #22                          ; 0380
                                              35    12 000F8          BNEQ    13$                              ;
                            53         14     A2    9E 000FA          MOVAB   20(R2), R3                       ; 0382
                                       03     A3    94 000FE          CLRB    3(R3)                            ;
                            55         18     A2    D0 00101          MOVL    24(R2), ADDR                     ; 0383
                                              55    DD 00105          PUSHL   ADDR                             ; 0384
           00000000G        00                01    FB 00107          CALLS   #1, DBG$IS_IT_ENTRY              ;
                            09                50    E9 0010E          BLBC    R0, 11$                          ;
                   02       A3                17    90 00111          MOVB    #23, 2(R3)                       ; 0387
                            63                02    B0 00115          MOVW    #2, (R3)                         ; 0388
                                              4D    11 00118          BRB     18$                              ; 0384
                   02       A3                16    90 0011A  11$:    MOVB    #22, 2(R3)                       ; 0393
                                              7E    7C 0011E          CLRQ    -(SP)                            ; 0395
                                              55    DD 00120          PUSHL   ADDR                             ;
           00000000G        00                03    FB 00122          CALLS   #3, DBG$INS_DECODE               ;
                   63       50                55    A3 00129          SUBW3   ADDR, R0, (R3)                   ;
                                              38    11 0012D  12$:    BRB     18$                              ; 0328
                            15                54    D1 0012F  13$:    CMPL    R4, #21                          ; 0403
                                              2A    12 00132          BNEQ    16$                              ;
                   16       A2                54    9B 00134          MOVZBW  R4, 22(R2)                       ; 0406
            0000FFFF        8F         0C     AC    D1 00138          CMPL    NEW_SIZE, #65535                 ; 0407
                                              20    12 00140          BNEQ    17$                              ;
                                              50    D4 00142          CLRL    I                                ; 0423
   09           18 B240     04                00    ED 00144  14$:    CMPZV   #0, #4, @24(R2)[I], #9           ; 0425
                                              0B    15 0014B          BLEQ    15$                              ;
                   51       50                01    78 0014D          ASHL    #1, I, R1                        ; 0428
          14       A2       51                01    A1 00151          ADDW3   #1, R1, 20(R2)                   ;
                                              0F    11 00156          BRB     18$                              ; 0427
```

```
                 E8           50           0F  F3 00158 15$:    AOBLEQ  #15, I, 14$                               ; 0423
                                           09  11 0015C         BRB     18$                                       ; 0328
                      16      A2           54  9B 0015E 16$:    MOVZBW  R4, 22(R2)                                ; 0444
                      14      A2     0C    AC  B0 00162 17$:    MOVW    NEW_SIZE, 20(R2)                          ; 0445
                              50           52  D0 00167 18$:    MOVL    R2, R0                                    ; 0450
                                               04 0016A         RET                                               ; 0451

; Routine Size:  363 bytes,    Routine Base:  DBG$CODE + 0000
```

```
1 GLOBAL ROUTINE DBG$COLLECT(PRM_DESC) : NOVALUE =
1 !
1 ! FUNCTION
1 !          ----------------------------
1 !
1 ! INPUTS
1 !          ----------------------------
1 !
1 ! OUTPUTS
1 !          ----------------------------
1 !
1
2     BEGIN
2
2     MAP
2         PRM_DESC: REF DBG$PRIMARY;          ! Pointer to Primary Descriptor
2
2     BUILTIN
2         REMQUE;                             ! Remove queue entry from list
2
2     LOCAL
2         XXXXXXX;                            !<<------------------------------------
2
2
2
2     ! --------------
2     !
2     IF (.PRM_DESC NEQA 0) THEN
2      IF (.PRM_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_PRIMARY_DESC) THEN
2       IF .PRM_DESC[DBG$V_DHDR_AGGR] THEN
3         BEGIN
3         LOCAL SUB_NODE : REF DBG$PRIM_NODE;
3
3         SUB_NODE = .PRM_DESC[DBG$L_PRIM_BLINK];
3
4         IF (.SUB_NODE[DBG$B_PNODE_FCODE] EQL RST$K_TYPE_ARRAY)
3                                 AND
4            (.SUB_NODE[DBG$B_PNARR_DTYPE] EQL DSC$K_DTYPE_T)
3                                 AND
4            (.SUB_NODE[DBG$W_PNARR_LENGTH] EQL 1)
3         THEN
4             BEGIN
4             BIND S_VECTOR = SUB_NODE[DBG$A_PNARR_SVECTOR] : DBG$PRIM_NODE_SUBS;
4             LOCAL DIMS,SIZE,BASE,TYPEID,SYMID;
4             DIMS = .SUB_NODE[DBG$B_PNARR_DIMCNT] - 1;
4             IF .S_VECTOR[.DIMS,DBG$L_PNSUB_STRIDE] NEQ 1 THEN RETURN;
4             IF .S_VECTOR[.DIMS,DBG$L_PNSUB_TYPEID] NEQ 0 THEN RETURN;
4
4             BASE =  .S_VECTOR[.DIMS,DBG$L_PNSUB_LBOUND];
4             SIZE = (.S_VECTOR[.DIMS,DBG$L_PNSUB_UBOUND] - .BASE) + 1;
4             PRM_DESC[DBG$W_PRIM_OFFSET] = .BASE;
4             PRM_DESC[DBG$W_PRIM_LENGTH] = .SIZE;
4             PRM_DESC[DBG$V_DHDR_SUBREF] = TRUE;
4             PRM_DESC[DBG$V_DHDR_TMPREF] = TRUE;
4             TYPEID = DBG$TYPEID_FOR_ATOMIC(DSC$K_DTYPE_T,.SIZE*%BPUNIT,FALSE);
4             IF .DIMS GTR 0
4             THEN
```

```
;   379        0509 5               BEGIN
;   380        0510 5               SUB_NODE[DBG$B_PNARR_DIMCNT]   = .DIMS;
;   381        0511 5               SUB_NODE[DBG$L_PNARR_CELLTYPE] = .TYPEID;
;   382        0512 5               END
;   383        0513 4           ELSE
;   384        0514 5               BEGIN
;   385        0515 5               SYMID = .SUB_NODE[DBG$L_PNODE_SYMID];
;   386        0516 5               REMQUE(.SUB_NODE,SUB_NODE);
;   387        0517 5               DBG$BUILD_PRIMARY_SUBNODE(.PRM_DESC,RST$K_DATA,.SYMID,
;   388        0518 5                                         RST$K_TYPE_ATOMIC,.TYPEID,0);
;   389        0519 5               PRM_DESC[DBG$V_DHDR_AGGR] = FALSE;
;   390        0520 5               SUB_NODE = .PRM_DESC[DBG$L_PRIM_BLINK];
;   391        0521 5               SUB_NODE[DBG$L_PNODE_RELOC] = -.BASE;
;   392        0522 4               END;
;   393        0523 3           END;
;   394        0524 2       END;
;   395        0525 1   END;                            ! End of dbg$collect



                                      001C 00000        .ENTRY  DBG$COLLECT, Save R2,R3,R4                    ; 0452
                     51         04  AC  D0 00002        MOVL    PRM_DESC, R1                                  ; 0479
                                    01  12 00006        BNEQ    1$                                            ;
                                        04 00008        RET                                                   ;
               79    8F         02  A1  91 00009 1$:    CMPB    2(R1), #121                                   ; 0480
                                    34  12 0000E        BNEQ    3$                                            ;
                     01         04  A1  E8 00010        BLBS    4(R1), 2$                                     ; 0481
                                        04 00014        RET                                                   ;
                     52         18  A1  D0 00015 2$:    MOVL    24(R1), SUB_NODE                              ; 0485
                     01         09  A2  91 00019        CMPB    9(SUB_NODE), #1                               ; 0487
                                    25  12 0001D        BNEQ    3$                                            ;
                     0E         1A  A2  91 0001F        CMPB    26(SUB_NODE), #14                             ; 0489
                                    1F  12 00023        BNEQ    3$                                            ;
                     01         1C  A2  B1 00025        CMPW    28(SUB_NODE), #1                              ; 0491
                                    19  12 00029        BNEQ    3$                                            ;
                     53         1B  A2  9A 0002B        MOVZBL  27(SUB_NODE), DIMS                            ; 0496
                                    53  D7 0002F        DECL    DIMS                                          ;
          50         53             14  C5 00031        MULL3   #20, DIMS, R0                                 ; 0497
                             2C A240 9F 00035           PUSHAB  44(SUB_NODE)[R0]                              ;
                     01             9E  D1 00039        CMPL    @(SP)+, #1                                    ;
                                    6D  12 0003C        BNEQ    5$                                            ;
                             38 A240 9F 0003E           PUSHAB  56(SUB_NODE)[R0]                              ; 0498
                                    9E  D5 00042        TSTL    @(SP)+                                        ;
                                    65  12 00044 3$:    BNEQ    5$                                            ;
                             30 A240 9F 00046           PUSHAB  48(SUB_NODE)[R0]                              ; 0500
                     54             9E  D0 0004A        MOVL    @(SP)+, BASE                                  ;
                             34 A240 9F 0004D           PUSHAB  52(SUB_NODE)[R0]                              ; 0501
          50         9E             54  C3 00051        SUBL3   BASE, @(SP)+, R0                              ;
                                    50  D6 00055        INCL    SIZE                                          ;
               10    A1             54  B0 00057        MOVW    BASE, 16(R1)                                  ; 0502
               12    A1             50  B0 0005B        MOVW    SIZE, 18(R1)                                  ; 0503
               04    A1       0102  8F  A8 0005F        BISW2   #258, 4(R1)                                   ; 0505
                                    7E  D4 00065        CLRL    -(SP)                                         ; 0506
          7E         50             03  78 00067        ASHL    #3, SIZE, -(SP)                               ;
                                    0E  DD 0006B        PUSHL   #14                                           ;
          00000000G  00          03  FB 0006D       CALLS   #3, DBG$TYPEID_FOR_ATOMIC               ;
                                 53  D5 00074       TSTL    DIMS                                    ; 0507
                                 09  15 00076       BLEQ    4$                                      ;
                 1B  A2          53  90 00078       MOVB    DIMS, 27(SUB_NODE)                      ; 0510
                 24  A2          50  D0 0007C       MOVL    TYPEID, 36(SUB_NODE)                    ; 0511
                                     04 00080       RET                                             ; 0507
                     51      10  A2  D0 00081 4$:   MOVL    16(SUB_NODE), SYMID                     ; 0515
                     52          62  0F 00085       REMQUE  (SUB_NODE), SUB_NODE                    ; 0516
                                 7E  D4 00088       CLRL    -(SP)                                   ; 0517
                                 50  DD 0008A       PUSHL   TYPEID                                  ; 0518
                                 02  DD 0008C       PUSHL   #2                                      ; 0517
                                 51  DD 0008E       PUSHL   SYMID                                   ;
                                 06  DD 00090       PUSHL   #6                                      ;
                     53      04  AC  D0 00092       MOVL    PRM_DESC, R3                            ;
                                 53  DD 00096       PUSHL   R3                                      ;
          00000000G  00          06  FB 00098       CALLS   #6, DBG$BUILD_PRIMARY_SUBNODE           ;
                 04  A3          01  8A 0009F       BICB2   #1, 4(R3)                               ; 0519
                     52      18  A3  D0 000A3       MOVL    24(R3), SUB_NODE                        ; 0520
                 14  A2          54  CE 000A7       MNEGL   BASE, 20(SUB_NODE)                      ; 0521
                                     04 000AB 5$:   RET                                             ; 0525

; Routine Size:  172 bytes,    Routine Base:  DBG$CODE + 016B
;  397        0526  1 ROUTINE DEPOSIT_HANDLER(SIGNAL_ARGS: REF BLOCK[,BYTE]) =
;  398        0527  1 !
;  399        0528  1 ! FUNCTION
;  400        0529  1 !       This routine is the handler for errors that are signalled during
;  401        0530  1 !       the processing of a DEPOSIT command. A handler is necessary so
;  402        0531  1 !       that we can restore page protections that we may have changed.
;  403        0532  1 !
;  404        0533  1 ! INPUTS
;  405        0534  1 !       NONE
;  406        0535  1 !
;  407        0536  1 ! OUTPUTS
;  408        0537  1 !       ----------------------------
;  409        0538  1 !
;  410        0539  1
;  411        0540  2     BEGIN
;  412        0541  2
;  413        0542  2     LOCAL
;  414        0543  2         MESSAGE_VECT;
;  415        0544  2
;  416        0545  2     ! If we get here the second time around (on the unwind from the
;  417        0546  2     ! final handler) then resignal the exception. Do not free up
;  418        0547  2     ! the page list again.
;  419        0548  2     !
;  420        0549  2     IF .SIGNAL_ARGS[CHF$L_SIG_NAME] EQL SS$_UNWIND
;  421        0550  2     THEN
;  422        0551  2         RETURN SS$_RESIGNAL;
;  423        0552  2
;  424        0553  2     IF .PAGE_LIST NEQ 0
;  425        0554  2     THEN
;  426        0555  2         DBG$SET_PAGE_PROT(PAGE_LIST,TRUE,MESSAGE_VECT);
;  427        0556  2
;  428        0557  2     RETURN SS$_RESIGNAL;
;  429        0558  1     END;



                                       0004 00000 DEPOSIT_HANDLER:
                                                             .WORD   Save R2                                  ; 0526
                52 00000000'   EF   9E 00002          MOVAB   PAGE_LIST, R2                            ;
                5E             04   C2 00009          SUBL2   #4, SP                                   ;
                50        04   AC   D0 0000C          MOVL    SIGNAL_ARGS, R0                          ; 0549
      00000920  8F        04   A0   D1 00010          CMPL    4(R0), #2336                             ;
                               11   13 00018          BEQL    1$                                       ;
                               62   D5 0001A          TSTL    PAGE_LIST                                ; 0553
                               0D   13 0001C          BEQL    1$                                       ;
                               5E   DD 0001E          PUSHL   SP                                       ; 0555
                               01   DD 00020          PUSHL   #1                                       ;
                               52   DD 00022          PUSHL   R2                                       ;
      00000000G 00             03   FB 00024          CALLS   #3, DBG$SET_PAGE_PROT                    ;
                50      0918   8F   3C 0002B 1$:      MOVZWL  #2328, R0                                ; 0557
                                    04 00030          RET                                              ; 0558

; Routine Size:  49 bytes,    Routine Base:  DBG$CODE + 0217
;  431         0559  1 GLOBAL ROUTINE DBG$DEPOSIT(VERB_NODE : REF DBG$VERB_NODE) : NOVALUE =
;  432         0560  1 !
;  433         0561  1 ! FUNCTION
;  434         0562  1 !        This routine accepts as input the command execution tree constructed
;  435         0563  1 !        by the parse network and performs the semantic actions corresponding to
;  436         0564  1 !        the parsed DEPOSIT command. If the command cannot be executed, a message
;  437         0565  1 !        argument vector is constructed and returned.
;  438         0566  1 !
;  439         0567  1 !        Upon entrance to this routine, the command has been classified as plain
;  440         0568  1 !        DEPOSIT or Instruction DEPOSIT, and all default and override types have
;  441         0569  1 !        been set up in the adverb nodes.
;  442         0570  1 !
;  443         0571  1 !        There should be two noun nodes. The first is the target of the deposit
;  444         0572  1 !        while the second represents the source (either a value descriptor or a
;  445         0573  1 !        pointer to a counted string for instruction DEPOSITS).
;  446         0574  1 !
;  447         0575  1 ! INPUTS
;  448         0576  1 !        VERB_NODE          - A longword containing the address of the verb (head)
;  449         0577  1 !                             node of the command execution tree
;  450         0578  1 !
;  451         0579  1 ! OUTPUTS
;  452         0580  1 !        -----------------------------
;  453         0581  1 !
;  454         0582  1
;  455         0583  2   BEGIN
;  456         0584  2
;  457         0585  2
;  458         0586  2   ROUTINE TEXT_LENGTH(VAL_DESC : REF DBG$VALDESC) =
;  459         0587  3        BEGIN
;  460         0588  3        LOCAL LENGTH;
;  461         0589  3        SELECTONE .VAL_DESC[DBG$B_VALUE_DTYPE] OF
;  462         0590  3            SET
;  463         0591  3            [DSC$K_DTYPE_T]:      LENGTH = .VAL_DESC[DBG$W_VALUE_LENGTH];
;  464         0592  3            [DSC$K_DTYPE_VT]:     LENGTH = .(.VAL_DESC[DBG$L_VALUE_POINTER])<0,16,0>;
;  465         0593  3            [DSC$K_DTYPE_AC]:     LENGTH = .(.VAL_DESC[DBG$L_VALUE_POINTER])<0, 8,0>;
;  466         0594  3            [OTHERWISE]:          SIGNAL(DBG$_ILLTYPE);
;  467         0595  3            TES;
;  468         0596  3        RETURN .LENGTH;
;  469         0597  2        END;



                                      0004 00000 TEXT_LENGTH:
                                                        .WORD    Save R2                              ; 0586
                 50        04   AC  D0 00002            MOVL     VAL_DESC, R0                         ; 0589
                 51        16   A0  9A 00006            MOVZBL   22(R0), R1
                 0E             51  91 0000A            CMPB     R1, #14                              ; 0591
                                06  12 0000D            BNEQ     1$                                   ;
                 52        14   A0  3C 0000F            MOVZWL   20(R0), LENGTH                       ;
                                23  11 00013            BRB      4$                                   ;
                 25             51  91 00015 1$:        CMPB     R1, #37                              ; 0592
                                06  12 00018            BNEQ     2$                                   ;
                 52        18   B0  3C 0001A            MOVZWL   @24(R0), LENGTH                      ;
                                18  11 0001E            BRB      4$                                   ;
                 26             51  91 00020 2$:        CMPB     R1, #38                              ; 0593
```

```
                                     06  12 00023          BNEQ    3$                                         ;
                   52          18    B0  9A 00025          MOVZBL  @24(R0), LENGTH                            ;
                                     0D  11 00029          BRB     4$                                         ;
                         000287D8    8F  DD 0002B 3$:      PUSHL   #165848                                    ; 0594
            00000000G  00            01  FB 00031          CALLS   #1, LIB$SIGNAL                             ;
                       50            52  D0 00038 4$:      MOVL    LENGTH, R0                                 ; 0596
                                         04 0003B          RET                                                ; 0597

; Routine Size:  60 bytes,    Routine Base:  DBG$CODE + 0248


;  470            0598  2
;  471            0599  2
;  472            0600  2       LOCAL
;  473            0601  2           SOURCE_NN       : REF DBG$NOUN_NODE,            ! Source of deposit
;  474            0602  2           TARGET_NN       : REF DBG$NOUN_NODE,            ! Target of deposit
;  475            0603  2           TYPE_NODE       : REF DBG$ADVERB_NODE,          ! Command qualifier
;  476            0604  2           PRIM_DESC       : REF DBG$PRIMARY,
;  477            0605  2           ADDR_DESC       : REF DBG$VALDESC,
;  478            0606  2           DATA_DESC       : REF DBG$VALDESC,
;  479            0607  2           MESSAGE_VECT;                                   ! Error message vector
;  480            0608  2
;  481            0609  2       BUILTIN CALLG;
;  482            0610  2
;  483            0611  2       ENABLE DEPOSIT_HANDLER;
;  484            0612  2
;  485            0613  2       TARGET_NN = .VERB_NODE[DBG$L_VERB_OBJECT_PTR];
;  486            0614  2       SOURCE_NN = .TARGET_NN[DBG$L_NOUN_LINK];
;  487            0615  2       PRIM_DESC = .TARGET_NN[DBG$L_NOUN_VALUE];
;  488            0616  2       DATA_DESC = .SOURCE_NN[DBG$L_NOUN_VALUE];
;  489            0617  2       PAGE_LIST = 0;
;  490            0618  2
;  491            0619  2       ! Convert both the source and the target to value descriptors.
;  492            0620  2       ! eval_lang_operator is used to convert the source because it
;  493            0621  2       ! is sensitive to any language-specific rules for converting
;  494            0622  2       ! primaries to values (e.g., in BLISS we do primary->address,
;  495            0623  2       ! in other languages we do primary->value).
;  496            0624  2       !
;  497            0625  2       IF .DATA_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_PRIMARY_DESC
;  498            0626  2       THEN
;  499            0627  2           DATA_DESC = DBG$EVAL_LANG_OPERATOR(DBG$GL_IDENTITY_TOKEN,
;  500            0628  2                                               .DATA_DESC, 0);
;  501            0629  2       DBG$PRIM_TO_VAL(.PRIM_DESC,DBG$K_V_VALUE_DESC,ADDR_DESC);
;  502            0630  2
;  503            0631  2       IF (TYPE_NODE = .VERB_NODE[DBG$L_VERB_ADVERB_PTR]) EQLA 0
;  504            0632  2         THEN DBG$SAVE_LOC(.PRIM_DESC)
;  505            0633  2         ELSE
;  506            0634  3           BEGIN
;  507            0635  3           LOCAL OVERRIDE_TYPE,OVERRIDE_SIZE;
;  508            0636  3           ADDR_DESC[DBG$B_DHDR_FCODE] = RST$K_TYPE_DESCR;
;  509            0637  3           ADDR_DESC[DBG$V_DHDR_OVERRIDE] = TRUE;
;  510            0638  3           OVERRIDE_TYPE = .TYPE_NODE[DBG$B_ADVERB_LITERAL];
;  511            0639  3           OVERRIDE_SIZE = .TYPE_NODE[DBG$L_ADVERB_VALUE];
;  512            0640  3           SELECTONE .OVERRIDE_TYPE OF
;  513            0641  3               SET
;  514            0642  3               [DSC$K_DTYPE_AZ,DSC$K_DTYPE_AC,DSC$K_DTYPE_VT]:
;  515            0643  4                   BEGIN
```

```
4                   IF .ADDR_DESC[DSC$B_CLASS] EQL DSC$K_CLASS_UBS THEN SIGNAL(DBG$_UNALIGNED);
4                   ADDR_DESC[DBG$B_VALUE_CLASS]  = DSC$K_CLASS_VS;
4                   ADDR_DESC[DBG$B_VALUE_DTYPE]  = .OVERRIDE_TYPE;
4                   ADDR_DESC[DBG$W_VALUE_LENGTH] = TEXT_LENGTH(.DATA_DESC);
3                   END;
3
3               [DBG$K_DTYPE_AD]:
3                   IF NOT CHECK_TEXT_DESCRIPTOR(.ADDR_DESC) THEN SIGNAL(DBG$_DESCNOTSET);
3
3               [OTHERWISE]:
4                   BEGIN
5                   IF (.ADDR_DESC[DBG$B_VALUE_CLASS] EQL DSC$K_CLASS_UBS)
4                   THEN
5                       BEGIN
6                       IF (.OVERRIDE_SIZE GTR 32)
6                       OR (.OVERRIDE_TYPE EQL DSC$K_DTYPE_ZI)
6                       OR (.OVERRIDE_TYPE EQL DSC$K_DTYPE_T)
5                       THEN
5                           SIGNAL(DBG$_UNALIGNED);
5                       END
4                   ELSE
4                       ADDR_DESC[DBG$B_VALUE_CLASS]  =  DSC$K_CLASS_Z;
4
5                   IF (.OVERRIDE_TYPE EQL DSC$K_DTYPE_T)
4                                   AND
5                      (.OVERRIDE_SIZE EQL 0)
4                   THEN OVERRIDE_SIZE = TEXT_LENGTH(.DATA_DESC);
4
4                   ADDR_DESC[DBG$B_VALUE_DTYPE]  = .OVERRIDE_TYPE;
4                   IF .OVERRIDE_TYPE EQL DSC$K_DTYPE_ZI
4                   THEN
5                       BEGIN
5                       LOCAL
5                           ADDR;
5
5                       ADDR = .ADDR_DESC[DBG$L_VALUE_POINTER];
5                       ADDR_DESC[DBG$W_VALUE_LENGTH] =
5                                   DBG$INS_DECODE(.ADDR, FALSE, FALSE ) - .ADDR;
5                       END
5
4                   ELSE
4                       ADDR_DESC[DBG$W_VALUE_LENGTH] = .OVERRIDE_SIZE;
4
3                   END;
3
3               TES;
3           DBG$SAVE_LOC(.PRIM_DESC,ADDR_DESC[DBG$A_VALUE_VMSDESC]);
2           END;
2
2       DBG$SAVE_VAL(.DATA_DESC);
2
2       IF NOT DBG$NGET_PAGES(.PRIM_DESC,PAGE_LIST,MESSAGE_VECT)
2        OR NOT DBG$SET_PAGE_PROT(PAGE_LIST,FALSE,MESSAGE_VECT)
2         THEN
3           BEGIN
3           PAGE_LIST = 0;
3           CALLG(.MESSAGE_VECT,LIB$SIGNAL);
```

```
;  573         0701  2          END;
;  574         0702  2
;  575         0703  2      DBG$EVAL_LANG_OPERATOR(DBG$GL_DEPOSIT_TOKEN,.DATA_DESC,.ADDR_DESC);
;  576         0704  2
;  577         0705  2      IF NOT DBG$SET_PAGE_PROT(PAGE_LIST,TRUE,MESSAGE_VECT)
;  578         0706  2      THEN
;  579         0707  3          BEGIN
;  580         0708  3          PAGE_LIST = 0;
;  581         0709  3          CALLG(.MESSAGE_VECT,LIB$SIGNAL);
;  582         0710  2          END;
;  583         0711  2
;  584         0712  2      ! Set registers context and return
;  585         0713  2      !
;  586         0714  2      DBG$STA_SETREGISTERS();
;  587         0715  2
;  588         0716  2      ! Update all watch point event entries after the DEPOSIT.
;  589         0717  2      !
;  590         0718  2      DBG$UPDATE_WATCHPOINTS();
;  591         0719  2
;  592         0720  1      END;



                                    0FFC 00000         .ENTRY   DBG$DEPOSIT, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0559
                                                                R10,R11                                      ;
              5B 00000000G    00  9E 00002         MOVAB    DBG$SAVE_LOC, R11                            ;
              5A 00000000G    00  9E 00009         MOVAB    DBG$EVAL_LANG_OPERATOR, R10                  ;
              59 00000000G    00  9E 00010         MOVAB    LIB$SIGNAL, R9                               ;
              58 00000000'    EF  9E 00017         MOVAB    PAGE_LIST, R8                                ;
              5E              08  C2 0001E         SUBL2    #8, SP                                       ;
              6D         016B CF  DE 00021         MOVAL    16$, (FP)                                    ; 0583
              52           04 AC  D0 00026         MOVL     VERB_NODE, R2                                ; 0613
              50           08 A2  D0 0002A         MOVL     8(R2), TARGET_NN                             ;
              51           08 A0  D0 0002E         MOVL     8(TARGET_NN), SOURCE_NN                      ; 0614
              57              60  D0 00032         MOVL     (TARGET_NN), PRIM_DESC                       ; 0615
              56              61  D0 00035         MOVL     (SOURCE_NN), DATA_DESC                       ; 0616
                              68  D4 00038         CLRL     PAGE_LIST                                    ; 0617
        79    8F           02 A6  91 0003A         CMPB     2(DATA_DESC), #121                           ; 0625
                              10  12 0003F         BNEQ     1$                                           ;
                              7E  D4 00041         CLRL     -(SP)                                        ; 0627
                              56  DD 00043         PUSHL    DATA_DESC                                    ; 0628
                 00000000G    00  9F 00045         PUSHAB   DBG$GL_IDENTITY_TOKEN                        ; 0627
              6A              03  FB 0004B         CALLS    #3, DBG$EVAL_LANG_OPERATOR                   ;
              56              50  D0 0004E         MOVL     R0, DATA_DESC                                ;
                              5E  DD 00051  1$:    PUSHL    SP                                           ; 0629
              7E           83 8F  9A 00053         MOVZBL   #131, -(SP)                                  ;
                              57  DD 00057         PUSHL    PRIM_DESC                                    ;
   00000000G  00              03  FB 00059         CALLS    #3, DBG$PRIM_TO_VAL                          ;
              53           04 A2  D0 00060         MOVL     4(R2), TYPE_NODE                             ; 0631
                              08  12 00064         BNEQ     2$                                           ;
                              57  DD 00066         PUSHL    PRIM_DESC                                    ; 0632
              6B              01  FB 00068         CALLS    #1, DBG$SAVE_LOC                             ;
                            00BE  31 0006B         BRW      12$                                          ;
              52              6E  D0 0006E  2$:    MOVL     ADDR_DESC, R2                                ; 0636
        06    A2              03  90 00071         MOVB     #3, 6(R2)                                    ;
        04  A2        80  8F  88 00075          BISB2   #128, 4(R2)                          ; 0637
            54            63  9A 0007A          MOVZBL  (TYPE_NODE), OVERRIDE_TYPE           ; 0638
            55        04  A3  D0 0007D          MOVL    4(TYPE_NODE), OVERRIDE_SIZE          ; 0639
            25            54  D1 00081          CMPL    OVERRIDE_TYPE, #37                   ; 0642
                          29  19 00084          BLSS    4$                                   ;
            27            54  D1 00086          CMPL    OVERRIDE_TYPE, #39                   ;
                          24  14 00089          BGTR    4$                                   ;
            0D        0C  A2  91 0008B          CMPB    12(R2), #13                          ; 0644
                          09  12 0008F          BNEQ    3$                                   ;
              00028D08    8F  DD 00091          PUSHL   #167176                              ;
            69            01  FB 00097          CALLS   #1, LIB$SIGNAL                       ;
        17  A2            0B  90 0009A 3$:      MOVB    #11, 23(R2)                          ; 0645
        16  A2            54  90 0009E          MOVB    OVERRIDE_TYPE, 22(R2)                ; 0646
                          56  DD 000A2          PUSHL   DATA_DESC                            ; 0647
      FF1B  CF            01  FB 000A4          CALLS   #1, TEXT_LENGTH                      ;
        14  A2            50  B0 000A9          MOVW    R0, 20(R2)                           ;
                          75  11 000AD          BRB     11$                                  ; 0640
            38            54  D1 000AF 4$:      CMPL    OVERRIDE_TYPE, #56                   ; 0650
                          15  12 000B2          BNEQ    5$                                   ;
                          52  DD 000B4          PUSHL   R2                                   ; 0651
     0000V  CF            01  FB 000B6          CALLS   #1, CHECK_TEXT_DESCRIPTOR            ;
            66            50  E8 000BB          BLBS    R0, 11$                              ;
              00028F50    8F  DD 000BE          PUSHL   #167760                              ;
            69            01  FB 000C4          CALLS   #1, LIB$SIGNAL                       ;
                          5B  11 000C7          BRB     11$                                  ;
            53        14  A2  9E 000C9 5$:      MOVAB   20(R2), R3                           ; 0655
            0D        03  A3  91 000CD          CMPB    3(R3), #13                           ;
                          1A  12 000D1          BNEQ    7$                                   ;
            20            55  D1 000D3          CMPL    OVERRIDE_SIZE, #32                   ; 0658
                          0A  14 000D6          BGTR    6$                                   ;
            16            54  D1 000D8          CMPL    OVERRIDE_TYPE, #22                   ; 0659
                          05  13 000DB          BEQL    6$                                   ;
            0E            54  D1 000DD          CMPL    OVERRIDE_TYPE, #14                   ; 0660
                          0E  12 000E0          BNEQ    8$                                   ;
              00028D08    8F  DD 000E2 6$:      PUSHL   #167176                              ; 0662
            69            01  FB 000E8          CALLS   #1, LIB$SIGNAL                       ;
                          03  11 000EB          BRB     8$                                   ; 0655
                      03  A3  94 000ED 7$:      CLRB    3(R3)                                ; 0665
            0E            54  D1 000F0 8$:      CMPL    OVERRIDE_TYPE, #14                   ; 0667
                          0E  12 000F3          BNEQ    9$                                   ;
                          55  D5 000F5          TSTL    OVERRIDE_SIZE                        ; 0669
                          0A  12 000F7          BNEQ    9$                                   ;
                          56  DD 000F9          PUSHL   DATA_DESC                            ; 0670
      FEC4  CF            01  FB 000FB          CALLS   #1, TEXT_LENGTH                      ;
            55            50  D0 00100          MOVL    R0, OVERRIDE_SIZE                    ;
        02  A3            54  90 00103 9$:      MOVB    OVERRIDE_TYPE, 2(R3)                 ; 0672
            16            54  D1 00107          CMPL    OVERRIDE_TYPE, #22                   ; 0673
                          15  12 0010A          BNEQ    10$                                  ;
            54        18  A2  D0 0010C          MOVL    24(R2), ADDR                         ; 0679
                          7E  7C 00110          CLRQ    -(SP)                                ; 0681
                          54  DD 00112          PUSHL   ADDR                                 ;
 00000000G  00            03  FB 00114          CALLS   #3, DBG$INS_DECODE                   ;
    63      50            54  A3 0011B          SUBW3   ADDR, R0, (R3)                       ;
                          03  11 0011F          BRB     11$                                  ; 0673
            63            55  B0 00121 10$:     MOVW    OVERRIDE_SIZE, (R3)                  ; 0685
                      14  A2  9F 00124 11$:     PUSHAB  20(R2)                               ; 0690
                          57  DD 00127          PUSHL   PRIM_DESC                            ;
```

```
                6B              02  FB 00129         CALLS   #2, DBG$SAVE_LOC                            ;
                                56  DD 0012C 12$:    PUSHL   DATA_DESC                                   ; 0693
      00000000G 00              01  FB 0012E         CALLS   #1, DBG$SAVE_VAL                            ;
                           04   AE  9F 00135         PUSHAB  MESSAGE_VECT                                ; 0695
                         0180   8F  BB 00138         PUSHR   #^M<R7,R8>                                  ;
      00000000G 00              03  FB 0013C         CALLS   #3, DBG$NGET_PAGES                          ;
                11              50  E9 00143         BLBC    R0, 13$                                     ;
                           04   AE  9F 00146         PUSHAB  MESSAGE_VECT                                ; 0696
                                7E  D4 00149         CLRL    -(SP)                                       ;
                                58  DD 0014B         PUSHL   R8                                          ;
      00000000G 00              03  FB 0014D         CALLS   #3, DBG$SET_PAGE_PROT                       ;
                06              50  E8 00154         BLBS    R0, 14$                                     ;
                                68  D4 00157 13$:    CLRL    PAGE_LIST                                   ; 0699
                69         04   BE  FA 00159         CALLG   @MESSAGE_VECT, LIB$SIGNAL                   ; 0700
                                6E  DD 0015D 14$:    PUSHL   ADDR_DESC                                   ; 0703
                                56  DD 0015F         PUSHL   DATA_DESC                                   ;
                    00000000G   00  9F 00161         PUSHAB  DBG$GL_DEPOSIT_TOKEN                        ;
                6A              03  FB 00167         CALLS   #3, DBG$EVAL_LANG_OPERATOR                  ;
                           04   AE  9F 0016A         PUSHAB  MESSAGE_VECT                                ; 0705
                                01  DD 0016D         PUSHL   #1                                          ;
                                58  DD 0016F         PUSHL   R8                                          ;
      00000000G 00              03  FB 00171         CALLS   #3, DBG$SET_PAGE_PROT                       ;
                06              50  E8 00178         BLBS    R0, 15$                                     ;
                                68  D4 0017B         CLRL    PAGE_LIST                                   ; 0708
                69         04   BE  FA 0017D         CALLG   @MESSAGE_VECT, LIB$SIGNAL                   ; 0709
      00000000G 00              00  FB 00181 15$:    CALLS   #0, DBG$STA_SETREGISTERS                    ; 0714
      00000000G 00              00  FB 00188         CALLS   #0, DBG$UPDATE_WATCHPOINTS                  ; 0718
                                    04 0018F         RET                                                 ; 0720
                                  0000 00190 16$:    .WORD   Save nothing                                ; 0583
                                7E  D4 00192         CLRL    -(SP)                                       ;
                                5E  DD 00194         PUSHL   SP                                          ;
                7E         04   AC  7D 00196         MOVQ    4(AP), -(SP)                                ;
           FDF4 CF              03  FB 0019A         CALLS   #3, DEPOSIT_HANDLER                         ;
                                    04 0019F         RET                                                 ;

; Routine Size:  416 bytes,    Routine Base:  DBG$CODE + 0284
```

```
1 GLOBAL ROUTINE DBG$EVALUATE(VERB_NODE): NOVALUE =
1 !
1 ! FUNCTION
1 !       This routine is the command execution network for the EVALUATE command.
1 !       Various semantic actions are performed which correspond to the arguments
1 !       and operands of the parsed input string.
1 !
1 !       EVALUATE sets last val '\', EVALUTATE/ADDRESS sets '.', current loc.
1 !
1 ! INPUTS
1 !       VERB_NODE                  - A longword containing the address of the head
1 !                                    node in the command execution tree
1 !
1 ! OUTPUTS
1 !       NONE
1 !
1
2     BEGIN
2
2     MAP
2         VERB_NODE: REF DBG$VERB_NODE;    ! Pointer to the input Verb Node
2
2     LOCAL
2         RADIX,
2         NOUN_NODE: REF DBG$NOUN_NODE,
2         BASE_NODE: REF DBG$ADVERB_NODE,
2         PRM_DESC: REF DBG$PRIMARY,
2         VAL_DESC: REF DBG$VALDESC;
2
2
2
2
2     ! Flush the current print buffer.  Then pick up the first Noun Node pointer,
2     ! the Adverb Node pointer, and the radix setting for this command.
2     !
2     DBG$FLUSHBUF();
2     NOUN_NODE = .VERB_NODE [DBG$L_VERB_OBJECT_PTR];
2     BASE_NODE = .VERB_NODE [DBG$L_VERB_ADVERB_PTR];
2     IF .BASE_NODE EQLA 0
2     THEN
2         RADIX = DBG$K_DEFAULT
2     ELSE
2         RADIX = .BASE_NODE[DBG$B_ADVERB_LITERAL];
2
2
2     ! Loop through all the Noun Nodes to process each expression on the
2     ! EVALUATE command.
2     !
2     WHILE .NOUN_NODE NEQ 0 DO
3         BEGIN
3         PRM_DESC = .NOUN_NODE[DBG$L_NOUN_VALUE];
3         DBG$COLLECT(.PRM_DESC);
3
3
3         ! Case on the kind of EVALUATE command Verb Node we have as determined
3         ! by the command qualifiers.
3         !
```

```
3         CASE .VERB_NODE[DBG$B_VERB_COMPOSITE] FROM EVALUATE TO EVALUATE_COND OF
3             SET
3
3
3             ! Handle the plain EVALUATE and the EVALUATE/CONDITION_VALUE com-
3             ! mands.
3             !
3             [EVALUATE,
3              EVALUATE_COND]:
4                 BEGIN
4                 IF .PRM_DESC[DBG$V_DHDR_AGGR] THEN SIGNAL(DBG$_NOVALUE);
4                 IF .VERB_NODE[DBG$B_VERB_COMPOSITE] EQL EVALUATE_COND
4                 THEN
4                     PRM_DESC[DBG$V_DHDR_FORMAT] = 1
4
4                 ELSE IF .RADIX NEQ DBG$K_DEFAULT
4                 THEN
4                     PRM_DESC[DBG$V_DHDR_FORMAT] = 0;
4
4                 DBG$PRINT_VALUE(.PRM_DESC,.RADIX, .DBG$GL_SIGN_FLAG);
4                 END;
3
3
3
3             ! Handle the EVALUATE/ADDRESS command.
3             !
3             [EVALUATE_ADDR]:
4                 BEGIN
4                 LOCAL
4                     NAMEPTR,
4                     REGDESCR,
4                     VMS_DESC: DBG$STG_DESC;
4
4                 DBG$SAVE_LOC(.PRM_DESC);
4                 DBG$PRIM_TO_VAL(.PRM_DESC,DBG$K_V_VALUE_DESC,VAL_DESC);
4
4
4                 ! Check whether the address is in the register save area.
4                 !
4                 REGDESCR = DBG$STA_ADDRESS_TO_REGDESCR(.VAL_DESC[DBG$L_VALUE_POINTER]);
4                 IF .REGDESCR NEQ 0 THEN
5                     BEGIN
5                     NAMEPTR = DBG$STA_REGISTER_NAME(.REGDESCR);
5                     DBG$PRINT(UPLIT BYTE(%ASCIC '!AC'), .NAMEPTR);
5                     END
5
4                 ELSE
5                     BEGIN
5                     VMS_DESC[DSC$B_CLASS] = DSC$K_CLASS_Z;
5                     VMS_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_LU;
5                     VMS_DESC[DSC$W_LENGTH] = 4;
5                     VMS_DESC[DSC$A_POINTER] = VAL_DESC[DBG$L_VALUE_POINTER];
5                     DBG$PRINT_VALUE_AS_INTEGER(VMS_DESC,.RADIX);
4                     END;
4
4
4                 ! If the address is a bit_field then also print the <p,s,e>.
4                 !
```

```
;  708          0835  4                     DBG$PRINT_FIELD_REF(.VAL_DESC,TRUE);
;  709          0836  3                     END;
;  710          0837  3
;  711          0838  3
;  712          0839  3               ! Any other kind of Verb Node should never occur.  If it does, we
;  713          0840  3               ! signal an internal DEBUG coding error.
;  714          0841  3               !
;  715          0842  3               [INRANGE,OUTRANGE]:
;  716          0843  3                   $DBG_ERROR('DBGLEVEL3\EVALUATE');
;  717          0844  3
;  718          0845  3               TES;
;  719          0846  3
;  720          0847  3
;  721          0848  3           ! Close out the current print line, link to the next Noun Node on the
;  722          0849  3           ! Noun Node list, and loop.
;  723          0850  3           !
;  724          0851  3           DBG$NEWLINE();
;  725          0852  3           NOUN_NODE = .NOUN_NODE[DBG$L_NOUN_LINK];
;  726          0853  3
;  727          0854  2           END;                            ! End of WHILE loop over expressions
;  728          0855  2
;  729          0856  2
;  730          0857  2       ! The EVALUATE command is processed.  Now return.
;  731          0858  2       !
;  732          0859  2       RETURN;
;  733          0860  2
;  734          0861  1       END;


                                                                          .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

                                            43  41  21  03  00000 P.AAA:  .ASCII  <3>\!AC\                                ;
4C  41  56  45  5C  33  4C  45  56  45  4C  47  42  44  12  00004 P.AAB:  .ASCII  <18>\DBGLEVEL3\<92>\EVALUATE\           ;
                                            45  54  41  55  00013                                                         ;


                                                                          .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                                  00FC 00000              .ENTRY  DBG$EVALUATE, Save R2,R3,R4,R5,R6,R7    ; 0721
                              57 00000000G  00  9E 00002              MOVAB   LIB$SIGNAL, R7                          ;
                              5E                10  C2 00009              SUBL2   #16, SP                                 ;
                    00000000G  00                00  FB 0000C              CALLS   #0, DBG$FLUSHBUF                        ; 0756
                              54          04  AC  D0 00013              MOVL    VERB_NODE, R4                           ; 0757
                              56          08  A4  D0 00017              MOVL    8(R4), NOUN_NODE                        ;
                              50          04  A4  D0 0001B              MOVL    4(R4), BASE_NODE                        ; 0758
                                              05  12 0001F              BNEQ    1$                                      ; 0759
                              55              01  D0 00021              MOVL    #1, RADIX                               ; 0761
                                              03  11 00024              BRB     2$                                      ;
                              55              60  9A 00026 1$:          MOVZBL  (BASE_NODE), RADIX                      ; 0763
                                              56  D5 00029 2$:          TSTL    NOUN_NODE                               ; 0769
                                              01  12 0002B              BNEQ    3$                                      ;
                                                  04 0002D              RET                                             ;
                              52              66  D0 0002E 3$:          MOVL    (NOUN_NODE), PRM_DESC                   ; 0771
                                              52  DD 00031              PUSHL   PRM_DESC                                ; 0772
                         FDOF  CF              01  FB 00033              CALLS   #1, DBG$COLLECT                         ;
```

```
                                                      H 8
DBGLEVEL3                                   16-Sep-1984 01:30:26    VAX-11 Bliss-32 V4.0-742           Page  24
V04-000                                     14-Sep-1984 12:17:02    [DEBUG.SRC]DBGLEVEL3.B32;1              (7)

              02          01        01  A4  8F 00038        CASEB   1(R4), #1, #2                        ; 0778
            0019        004F      0019      0003D 4$:       .WORD   5$-4$,-                              ;
                                                                    10$-4$,-                             ;
                                                                    5$-4$                                ;
                           00000000'    EF  9F 00043        PUSHAB  P.AAB                                ; 0843
                                        01  DD 00049        PUSHL   #1                                   ;
                            00028362    8F  DD 0004B        PUSHL   #164706                              ;
                          67            03  FB 00051        CALLS   #3, LIB$SIGNAL                       ;
                                        34  11 00054        BRB     9$                                   ;
                          09        04  A2  E9 00056 5$:    BLBC    4(PRM_DESC), 6$                      ; 0788
                            000287F8    8F  DD 0005A        PUSHL   #165880                              ;
                          67            01  FB 00060        CALLS   #1, LIB$SIGNAL                       ;
                          03        01  A4  91 00063 6$:    CMPB    1(R4), #3                            ; 0789
                                        08  12 00067        BNEQ    7$                                   ;
  05   A2          04     04            01  F0 00069        INSV    #1, #4, #4, 5(PRM_DESC)              ; 0791
                                        0A  11 0006F        BRB     8$                                   ;
                          01            55  D1 00071 7$:    CMPL    RADIX, #1                            ; 0793
                                        05  13 00074        BEQL    8$                                   ;
                    05    A2        F0  8F  8A 00076        BICB2   #240, 5(PRM_DESC)                    ; 0795
                           00000000G    00  DD 0007B 8$:    PUSHL   DBG$GL_SIGN_FLAG                     ; 0797
                                        24  BB 00081        PUSHR   #^M<R2,R5>                           ;
             00000000G    00            03  FB 00083        CALLS   #3, DBG$PRINT_VALUE                  ;
                                        67  11 0008A 9$:    BRB     13$                                  ; 0778
                                        52  DD 0008C 10$:   PUSHL   PRM_DESC                             ; 0810
             00000000G    00            01  FB 0008E        CALLS   #1, DBG$SAVE_LOC                     ;
                                        5E  DD 00095        PUSHL   SP                                   ; 0811
                          7E        83  8F  9A 00097        MOVZBL  #131, -(SP)                          ;
                                        52  DD 0009B        PUSHL   PRM_DESC                             ;
             00000000G    00            03  FB 0009D        CALLS   #3, DBG$PRIM_TO_VAL                  ;
                          53            6E  D0 000A4        MOVL    VAL_DESC, R3                         ; 0816
                                    18  A3  DD 000A7        PUSHL   24(R3)                               ;
             00000000G    00            01  FB 000AA        CALLS   #1, DBG$STA_ADDRESS_TO_REGDESCR      ;
                                        50  D5 000B1        TSTL    REGDESCR                             ; 0817
                                        1A  13 000B3        BEQL    11$                                  ;
                                        50  DD 000B5        PUSHL   REGDESCR                             ; 0819
             00000000G    00            01  FB 000B7        CALLS   #1, DBG$STA_REGISTER_NAME            ;
                                        50  DD 000BE        PUSHL   NAMEPTR                              ; 0820
                           00000000'    EF  9F 000C0        PUSHAB  P.AAA                                ;
             00000000G    00            02  FB 000C6        CALLS   #2, DBG$PRINT                        ;
                                        19  11 000CD        BRB     12$                                  ; 0817
                    04    AE  00040004  8F  D0 000CF 11$:   MOVL    #262148, VMS_DESC                    ; 0827
                    08    AE        18  A3  9E 000D7        MOVAB   24(R3), VMS_DESC+4                   ; 0828
                                        55  DD 000DC        PUSHL   RADIX                                ; 0829
                                    08  AE  9F 000DE        PUSHAB  VMS_DESC                             ;
             00000000G    00            02  FB 000E1        CALLS   #2, DBG$PRINT_VALUE_AS_INTEGER       ;
                                        01  DD 000E8 12$:   PUSHL   #1                                   ; 0835
                                        53  DD 000EA        PUSHL   R3                                   ;
             00000000G    00            02  FB 000EC        CALLS   #2, DBG$PRINT_FIELD_REF              ;
             00000000G    00            00  FB 000F3 13$:   CALLS   #0, DBG$NEWLINE                      ; 0851
                          56        08  A6  D0 000FA        MOVL    8(NOUN_NODE), NOUN_NODE              ; 0852
                                      FF28  31 000FE        BRW     2$                                   ; 0769
                                            04 00101        RET                                          ; 0861

; Routine Size:  258 bytes,    Routine Base:  DBG$CODE + 0424
```

```
1 GLOBAL ROUTINE DBG$EXAMINE(VERB_NODE: REF DBG$VERB_NODE): NOVALUE =
1 !
1 ! FUNCTION
1 !       This routine performs the action associated with EXAMINE xxx.
1 !       We always get three adverb nodes linked to the verb node. See the
1 !       routine header for DBG$NPARSE_EXAMINE in DBGNEXMNE.B32 for details.
1 !
1 ! INPUTS
1 !       VERB_NODE - A longword containing the address of the command
1 !                   execution tree verb (head) node.
1 !
1 ! OUTPUTS
1 !       NONE
1 !
1
2     BEGIN
2
2     LOCAL
2         NOUN_NODE        : REF DBG$NOUN_NODE,
2         TYPE_NODE        : REF DBG$ADVERB_NODE,
2         BASE_NODE        : REF DBG$ADVERB_NODE,
2         MODE_NODE        : REF DBG$ADVERB_NODE,
2         PRM_DESC         : REF DBG$PRIMARY,
2         END_DESC         : REF DBG$PRIMARY,
2         VAL_DESC         : REF DBG$VALDESC,
2         NEW_SIZE         : WORD,
2         NEW_TYPE         : BYTE,
2         RADIX            : BYTE,
2         FORMAT_ONE       : BYTE,
2         FORMAT_TWO       : BYTE;
2
2     NOUN_NODE = .VERB_NODE[DBG$L_VERB_OBJECT_PTR];
2     TYPE_NODE = .VERB_NODE[DBG$L_VERB_ADVERB_PTR];
2     BASE_NODE = .TYPE_NODE[DBG$L_ADVERB_LINK];
2     MODE_NODE = .BASE_NODE[DBG$L_ADVERB_LINK];
2
2     SELECTONE .VERB_NODE[DBG$B_VERB_COMPOSITE] OF
2         SET
2         [EXAMINE]:
3             BEGIN
3             NEW_TYPE   = .TYPE_NODE[DBG$B_ADVERB_LITERAL];
3             NEW_SIZE   = .TYPE_NODE[DBG$L_ADVERB_VALUE];
3             RADIX      = .BASE_NODE[DBG$B_ADVERB_LITERAL];
3             FORMAT_ONE = 0;
2             END;
2
2         [EXAMINE_SOURCE]:        0;
2
2         [EXAMINE_CONDITION_VALUE]:
3             BEGIN
3             NEW_TYPE   = DSC$K_DTYPE_LU;
3             NEW_SIZE   = 4;
3             RADIX      = DBG$K_DEFAULT;
3             FORMAT_ONE = 1;
2             END;
2
2         [EXAMINE_PSL]:
```

```
            BEGIN
            NEW_TYPE    = DSC$K_DTYPE_LU;
            NEW_SIZE    = 4;
            RADIX       = DBG$K_DEFAULT;
            FORMAT_ONE = 2;
            END;

        [EXAMINE_PSW]:
            BEGIN
            NEW_TYPE    = DSC$K_DTYPE_WU;
            NEW_SIZE    = 2;
            RADIX       = DBG$K_DEFAULT;
            FORMAT_ONE = 3;
            END;

        ! Any other kind of the Verb Node is invalid, so we signal an internal
        ! DEBUG coding error.
        !
        [OTHERWISE]:
            $DBG_ERROR('DBGLEVEL3\EXAMINE');

        TES;

    ! ------
    !
    DO  BEGIN
        DBG$FLUSHBUF();

        IF .VERB_NODE[DBG$B_VERB_COMPOSITE] EQL EXAMINE_SOURCE
        THEN
            BEGIN
            LOCAL
                VAL_DESC        : REF DBG$VALDESC,
                START_ADDRESS,
                FINAL_ADDRESS;

            DBG$PRIM_TO_VAL(.NOUN_NODE[DBG$L_NOUN_VALUE ],DBG$K_V_VALUE_DESC,VAL_DESC);
            START_ADDRESS = .VAL_DESC[DBG$L_VALUE_POINTER];
            DBG$PRIM_TO_VAL(.NOUN_NODE[DBG$L_NOUN_VALUE2],DBG$K_V_VALUE_DESC,VAL_DESC);
            FINAL_ADDRESS = .VAL_DESC[DBG$L_VALUE_POINTER];

            ! Output the source. The third parameter indicates that the
            ! module name is to be displayed.

            DBG$SRC_TYPE_PC_SOURCE(.START_ADDRESS,.FINAL_ADDRESS,TRUE,FALSE);

            PRM_DESC = .NOUN_NODE[DBG$L_NOUN_VALUE2];
!           Commented out because screen windown does EXAMINE/SOURCE and
!           we don't want to save dot there.
!
!           DBG$SAVE_LOC(.PRM_DESC);
            END         ! EXAMINE/SOURCE

        ! ------
```

```
4              !
3          ELSE
4              BEGIN       ! Data Examine
4              PRM_DESC = .NOUN_NODE[DBG$L_NOUN_VALUE];
4              END_DESC = .NOUN_NODE[DBG$L_NOUN_VALUE2];
4              DBG$COLLECT(.PRM_DESC);
4              DBG$COLLECT(.END_DESC);
4
5              IF (.END_DESC NEQ 0) AND (.PRM_DESC NEQ .END_DESC)
4              THEN
5                  BEGIN
5                  !+
5                  ! We have a ranged examine (EXAMINE <prm>:<end>)
5                  ! Check for the case where the two endpoints are part
5                  ! of th same structure. We have to ensure that a number
5                  ! of conditions are met, e.g., they are both primaries,
5                  ! they are not aggregates, and so on.
5                  !-
6                  IF      (.PRM_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_PRIMARY_DESC)
5                                          AND
6                          (.END_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_PRIMARY_DESC)
5                                          AND
6                          (.PRM_DESC[DBG$L_DHDR_SYMID0] EQL .END_DESC[DBG$L_DHDR_SYMID0])
5                                          AND
6                          (.NEW_TYPE EQL DBG$K_NOTYPE)
5                                          AND
6                          (NOT .PRM_DESC[DBG$V_DHDR_AGGR])
5                                          AND
6                          (NOT .END_DESC[DBG$V_DHDR_AGGR])
5                                          AND
6                          (NOT .PRM_DESC[DBG$V_DHDR_SUBREF])
5                                          AND
6                          (NOT .END_DESC[DBG$V_DHDR_SUBREF])
5                  THEN
6                      BEGIN
6                      !+
6                      ! The start and end of the ranged examine appear to be
6                      ! part of the same aggregate structure. Check that the
6                      ! start is earlier than the end
6                      !-
6                      IF PRIMARY_ORDER(.PRM_DESC,.END_DESC) GTR 0 THEN SIGNAL(DBG$_EXARANGE);
6                      WHILE TRUE DO
7                          BEGIN
7                          LOCAL MARK;
7                          MARK = DBG$PUSH_TEMPMEM();
7                          DBG$PRINT_IDENTIFIER(.PRM_DESC);
7                          DBG$PRINT(UPLIT BYTE(%ASCIC '!AD! '), 1, UPLIT BYTE(':'));
7                          DBG$PRIM_TO_VAL(.PRM_DESC,DBG$K_VALUE_DESC,VAL_DESC);
7                          IF .FORMAT_ONE NEQ 0 THEN VAL_DESC[DBG$V_DHDR_FORMAT] = .FORMAT_ONE;
7                          DBG$PRINT_VALUE(.VAL_DESC,.RADIX, .DBG$GL_SIGN_FLAG);
7                          DBG$NEWLINE();
7                          DBG$SAVE_LOC(.PRM_DESC);
7                          DBG$POP_TEMPMEM(.MARK);
7                          IF PRIMARY_ORDER(.PRM_DESC,.END_DESC) GEQ 0 THEN EXITLOOP;
7                          IF NOT MODIFY_PRIMARY(.PRM_DESC,0) THEN EXITLOOP;
6                          END;
6                      END
```

```
5              ELSE
6                  BEGIN
6                  ! +
6                  ! The start and end are NOT part of the same aggregate.
6                  ! -
6                  LOCAL
6                      MARK,
6                      LAST_ADDR,
6                      NEXT_ADDR,
6                      DESC_TYPE,
6                      ADDR_DESC           : REF DBG$VALDESC,
6                      RDESC_ONE           : DBG$REGDESCR,
6                      RDESC_TWO           : DBG$REGDESCR,
6                      LENGTH;
6
6                  ADDR_DESC = DBG$CHANGE_DTYPE(.END_DESC,.NEW_TYPE,.NEW_SIZE);
6                  RDESC_ONE = DBG$STA_ADDRESS_TO_REGDESCR(.ADDR_DESC[DBG$L_VALUE_POINTER]);
6                  LAST_ADDR = .ADDR_DESC[DBG$L_VALUE_POINTER];
6                  ADDR_DESC = DBG$CHANGE_DTYPE(.PRM_DESC,.NEW_TYPE,.NEW_SIZE);
6                  RDESC_TWO = DBG$STA_ADDRESS_TO_REGDESCR(.ADDR_DESC[DBG$L_VALUE_POINTER]);
6                  IF ((.RDESC_ONE XOR .RDESC_TWO) AND %X'FFFF00FC') NEQ 0
6                  THEN
6                      SIGNAL(DBG$_EXARANGE);
6
6                  IF .LAST_ADDR LSSA .ADDR_DESC[DBG$L_VALUE_POINTER]
6                  THEN
6                      SIGNAL(DBG$_EXARANGE);
6
7                  IF (.ADDR_DESC[DBG$B_VALUE_CLASS] EQL DSC$K_CLASS_UBS)
6                  THEN
6                      SIGNAL(DBG$_ILLTYPE);
6
6                  DESC_TYPE = DBG$K_VALUE_DESC;
6                  IF (.ADDR_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_ZEM) OR
7                     (.ADDR_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_ZI)
6                  THEN
6                      DESC_TYPE = DBG$K_V_VALUE_DESC;
6
6
6                  ! ------
6                  !
6                  WHILE TRUE DO
7                      BEGIN
7                      MARK = DBG$PUSH_TEMPMEM();
7                      DBG$PRINT_IDENTIFIER(.ADDR_DESC);
7                      DBG$PRINT(UPLIT BYTE(%ASCIC '!AD! '),1,UPLIT BYTE(':'));
7                      DBG$PRIM_TO_VAL(.ADDR_DESC,.DESC_TYPE,VAL_DESC);
7                      IF .FORMAT_ONE NEQ 0 THEN VAL_DESC[DBG$V_DHDR_FORMAT] = .FORMAT_ONE;
7                      DBG$PRINT_VALUE(.VAL_DESC,.RADIX, .DBG$GL_SIGN_FLAG);
7                      DBG$NEWLINE();
7                      DBG$POP_TEMPMEM(.MARK);
7
7                      ! Get the increment we will add to the address for
7                      ! the next line of the ranged examine. If the increment
7                      ! is zero then signal an informational and get out of the loop.
7                      !
7                      LENGTH = (DBG$DATA_LENGTH(ADDR_DESC[DBG$A_VALUE_VMSDESC]) + (%BPUNIT-1))/%BPUNIT;
```

```
7                 IF .LENGTH EQL 0
7                 THEN
8                     BEGIN
8                     SIGNAL(DBG$_ZEROINCR); ! Informational
8                     EXITLOOP;
7                     END;
7
7                 NEXT_ADDR = .ADDR_DESC[DBG$L_VALUE_POINTER] + .LENGTH;
7                 IF .NEXT_ADDR GTRA .LAST_ADDR THEN EXITLOOP;
7                 ADDR_DESC[DBG$L_VALUE_POINTER] = .NEXT_ADDR;
9                 IF ((.ADDR_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_ZEM)
8                  OR (.ADDR_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_ZI))
7                   THEN
8                     BEGIN
8                     IF DBG$IS_IT_ENTRY(.NEXT_ADDR)
8                       THEN ADDR_DESC[DBG$B_VALUE_DTYPE]  = DSC$K_DTYPE_ZEM
8                       ELSE ADDR_DESC[DBG$B_VALUE_DTYPE]  = DSC$K_DTYPE_ZI;
8                     ADDR_DESC[DBG$W_VALUE_LENGTH] = DBG$INS_DECODE(.NEXT_ADDR,FALSE) - .NEXT_ADDR;
8                     END
7                   ELSE
7                     ADDR_DESC[DBG$W_VALUE_LENGTH] = FIX_UP_LENGTH(ADDR_DESC[DBG$A_VALUE_VMSDESC]);
6                 END;
6             DBG$SAVE_LOC(.ADDR_DESC);
5                 END;
5         END
4 ELSE
5     BEGIN
5
5
5     ! In the case where prm_desc is a volatile value descriptor
5     ! representing an absolute address, the print_identifier
5     ! will attempt to symbolize this address to a primary. If
5     ! it succeeds, it will return the newly-constructed primary..
5     ! In all other cases, it just returns the descriptor we pass
5     ! into it, unchanged.
5     !
5     PRM_DESC = DBG$PRINT_IDENTIFIER(.PRM_DESC);
5     DBG$SAVE_LOC(.PRM_DESC);
5     IF .NEW_TYPE EQL DBG$K_NOTYPE AND .PRM_DESC[DBG$V_DHDR_AGGR]
5     THEN
5         DBG$PRINT_AGGREGATE(.PRM_DESC,.RADIX)
5     ELSE
6         BEGIN
6         DBG$PRINT(UPLIT BYTE(%ASCIC '!AD! '), 1, UPLIT BYTE(':'));
6         VAL_DESC = DBG$CHANGE_DTYPE(.PRM_DESC,.NEW_TYPE,.NEW_SIZE);
6         FORMAT_TWO = .FORMAT_ONE;
6         IF .NEW_TYPE NEQ DBG$K_NOTYPE
6         THEN
6             DBG$SAVE_LOC(.PRM_DESC,VAL_DESC[DBG$A_VALUE_VMSDESC])
7         ELSE IF (.FORMAT_ONE EQL 0) AND (.RADIX EQL DBG$K_DEFAULT)
7               AND (.VAL_DESC[DBG$B_VALUE_CLASS] NEQ DSC$K_CLASS_UBS)
7                AND (.VAL_DESC[DBG$L_VALUE_POINTER] EQLA DBG$REG_VALUES[16])
6                 THEN FORMAT_TWO = 2;
6
6         IF .VAL_DESC[DBG$B_VALUE_DTYPE] NEQ DSC$K_DTYPE_ZI
6         THEN
7             BEGIN
```

```
; 1021        1147  7                      DBG$PRIM_TO_VAL(.VAL_DESC,DBG$K_VALUE_DESC,VAL_DESC);
; 1022        1148  7                      DBG$DO_MAPPING(.VAL_DESC);
; 1023        1149  6                      END;
; 1024        1150  6
; 1025        1151  6                  IF .FORMAT_TWO NEQ 0 THEN VAL_DESC[DBG$V_DHDR_FORMAT] = .FORMAT_TWO;
; 1026        1152  6                  DBG$PRINT_VALUE(.VAL_DESC,.RADIX,.DBG$GL_SIGN_FLAG);
; 1027        1153  6                  DBG$NEWLINE();
; 1028        1154  5                  END;
; 1029        1155  4              END;
; 1030        1156  3          END;
; 1031        1157  2      END UNTIL (NOUN_NODE = .NOUN_NODE[DBG$L_NOUN_LINK]) EQL 0;
; 1032        1158  2
; 1033        1159  2  RETURN STS$K_SUCCESS;
; 1034        1160  1  END;



                                                                      .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

4D  41  58  45  5C  33  4C  45  56  45  4C  47  42  44  11  00017 P.AAC:  .ASCII  <17>\DBGLEVEL3\<92>\EXAMINE\      ;
                                                45  4E  49  00026                                                  ;
                                5F  21  44  41  21  05  00029 P.AAD:  .ASCII  <5>\!AD!_\                         ;
                                                    3A  0002F P.AAE:  .ASCII  \:\                                ;
                                5F  21  44  41  21  05  00030 P.AAF:  .ASCII  <5>\!AD!_\                         ;
                                                    3A  00036 P.AAG:  .ASCII  \:\                                ;
                                5F  21  44  41  21  05  00037 P.AAH:  .ASCII  <5>\!AD!_\                         ;
                                                    3A  0003D P.AAI:  .ASCII  \:\                                ;



                                                                      .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                           0FFC 00000                 .ENTRY  DBG$EXAMINE, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0862
                                                                              R10,R11                              ;
                          5E          20  C2 00002                    SUBL2   #32, SP                              ;
                          50     04   AC  D0 00005                    MOVL    VERB_NODE, R0                        ; 0893
                          55     08   A0  D0 00009                    MOVL    8(R0), NOUN_NODE                     ;
                          51     04   A0  D0 0000D                    MOVL    4(R0), TYPE_NODE                     ; 0894
                          52     08   A1  D0 00011                    MOVL    8(TYPE_NODE), BASE_NODE              ; 0895
                          53     08   A2  D0 00015                    MOVL    8(BASE_NODE), MODE_NODE              ; 0896
                          5B     01   A0  9A 00019                    MOVZBL  1(R0), R11                           ; 0898
                          01          5B  91 0001D                    CMPB    R11, #1                              ; 0900
                                      0E  12 00020                    BNEQ    1$                                   ;
                          59          61  90 00022                    MOVB    (TYPE_NODE), NEW_TYPE                ; 0902
                          6E     04   A1  B0 00025                    MOVW    4(TYPE_NODE), NEW_SIZE               ; 0903
                          5A          62  90 00029                    MOVB    (BASE_NODE), RADIX                   ; 0904
                                      58  94 0002C                    CLRB    FORMAT_ONE                           ; 0905
                                      53  11 0002E                    BRB     5$                                   ; 0898
                          04          5B  91 00030 1$:                CMPB    R11, #4                              ; 0908
                                      4E  13 00033                    BEQL    5$                                   ;
                          05          5B  91 00035                    CMPB    R11, #5                              ; 0910
                                      0E  12 00038                    BNEQ    2$                                   ;
                          59          04  90 0003A                    MOVB    #4, NEW_TYPE                         ; 0912
                          6E          04  B0 0003D                    MOVW    #4, NEW_SIZE                         ; 0913
                          5A          01  90 00040                    MOVB    #1, RADIX                            ; 0914
                          58          01  90 00043                    MOVB    #1, FORMAT_ONE                       ; 0915
                                      3B  11 00046                    BRB     5$                                   ; 0898
```

```
                 06           5B  91 00048 2$:     CMPB    R11, #6                                               ; 0918
                              0E  12 0004B         BNEQ    3$                                                    ;
                 59           04  90 0004D         MOVB    #4, NEW_TYPE                                          ; 0920
                 6E           04  B0 00050         MOVW    #4, NEW_SIZE                                          ; 0921
                 5A           01  90 00053         MOVB    #1, RADIX                                             ; 0922
                 58           02  90 00056         MOVB    #2, FORMAT_ONE                                        ; 0923
                              28  11 00059         BRB     5$                                                    ; 0898
                 07           5B  91 0005B 3$:     CMPB    R11, #7                                               ; 0926
                              0E  12 0005E         BNEQ    4$                                                    ;
                 59           03  90 00060         MOVB    #3, NEW_TYPE                                          ; 0928
                 6E           02  B0 00063         MOVW    #2, NEW_SIZE                                          ; 0929
                 5A           01  90 00066         MOVB    #1, RADIX                                             ; 0930
                 58           03  90 00069         MOVB    #3, FORMAT_ONE                                        ; 0931
                              15  11 0006C         BRB     5$                                                    ; 0898
                    00000000' EF  9F 0006E 4$:     PUSHAB  P.AAC                                                 ; 0939
                              01  DD 00074         PUSHL   #1                                                    ;
                    00028362  8F  DD 00076         PUSHL   #164706                                               ;
       00000000G 00           03  FB 0007C         CALLS   #3, LIB$SIGNAL                                        ;
       00000000G 00           00  FB 00083 5$:     CALLS   #0, DBG$FLUSHBUF                                      ; 0947
                 04           5B  91 0008A         CMPB    R11, #4                                               ; 0949
                              46  12 0008D         BNEQ    6$                                                    ;
                          18  AE  9F 0008F         PUSHAB  VAL_DESC                                              ; 0957
                 7E       83  8F  9A 00092         MOVZBL  #131, -(SP)                                           ;
                              65  DD 00096         PUSHL   (NOUN_NODE)                                           ;
       00000000G 00           03  FB 00098         CALLS   #3, DBG$PRIM_TO_VAL                                   ;
                 50       18  AE  D0 0009F         MOVL    VAL_DESC, R0                                          ; 0958
                 52       18  A0  D0 000A3         MOVL    24(R0), START_ADDRESS                                 ;
                          18  AE  9F 000A7         PUSHAB  VAL_DESC                                              ; 0959
                 7E       83  8F  9A 000AA         MOVZBL  #131, -(SP)                                           ;
                          0C  A5  DD 000AE         PUSHL   12(NOUN_NODE)                                         ;
       00000000G 00           03  FB 000B1         CALLS   #3, DBG$PRIM_TO_VAL                                   ;
                 50       18  AE  D0 000B8         MOVL    VAL_DESC, R0                                          ; 0960
                 50       18  A0  D0 000BC         MOVL    24(R0), FINAL_ADDRESS                                 ;
                 7E           01  7D 000C0         MOVQ    #1, -(SP)                                             ; 0965
                              50  DD 000C3         PUSHL   FINAL_ADDRESS                                         ;
                              52  DD 000C5         PUSHL   START_ADDRESS                                         ;
       00000000G 00           04  FB 000C7         CALLS   #4, DBG$SRC_TYPE_PC_SOURCE                            ;
                 53       0C  A5  D0 000CE         MOVL    12(NOUN_NODE), PRM_DESC                               ; 0967
                            035D  31 000D2         BRW     36$                                                   ; 0949
                 53           65  D0 000D5 6$:     MOVL    (NOUN_NODE), PRM_DESC                                 ; 0979
                 56       0C  A5  D0 000D8         MOVL    12(NOUN_NODE), END_DESC                               ; 0980
                              53  DD 000DC         PUSHL   PRM_DESC                                              ; 0981
            FB62 CF           01  FB 000DE         CALLS   #1, DBG$COLLECT                                       ;
                              56  DD 000E3         PUSHL   END_DESC                                              ; 0982
            FB5B CF           01  FB 000E5         CALLS   #1, DBG$COLLECT                                       ;
                              56  D5 000EA         TSTL    END_DESC                                              ; 0984
                              03  13 000EC         BEQL    7$                                                    ;
                 56           53  D1 000EE         CMPL    PRM_DESC, END_DESC                                    ;
                              03  12 000F1 7$:     BNEQ    8$                                                    ;
                            0258  31 000F3         BRW     29$                                                   ;
              79 8F       02  A3  91 000F6 8$:     CMPB    2(PRM_DESC), #121                                     ; 0994
                              18  12 000FB         BNEQ    9$                                                    ;
              79 8F       02  A6  91 000FD         CMPB    2(END_DESC), #121                                     ; 0996
                              11  12 00102         BNEQ    9$                                                    ;
              0C A6       0C  A3  D1 00104         CMPL    12(PRM_DESC), 12(END_DESC)                            ; 0998
                              0A  12 00109         BNEQ    9$                                                    ;
              80 8F           59  91 0010B         CMPB    NEW_TYPE, #128                                        ; 1000
```

```
                                         04  12 0010F          BNEQ    9$                                 ;
                       03          04    A3  E9 00111          BLBC    4(PRM_DESC), 10$                   ; 1002
                                       00B9  31 00115 9$:      BRW     15$                                ;
                       F9          04    A6  E8 00118 10$:     BLBS    4(END_DESC), 9$                    ; 1004
            F4         04    A3          01  E0 0011C          BBS     #1, 4(PRM_DESC), 9$                ; 1006
            EF         04    A6          01  E0 00121          BBS     #1, 4(END_DESC), 9$                ; 1008
                               0048      8F  BB 00126          PUSHR   #^M<R3,R6>                         ; 1016
                 0000V CF                02  FB 0012A          CALLS   #2, PRIMARY_ORDER                  ;
                                         50  D5 0012F          TSTL    R0                                 ;
                                         0D  15 00131          BLEQ    11$                                ;
                           00028190      8F  DD 00133          PUSHL   #164240                            ;
             00000000G 00                01  FB 00139          CALLS   #1, LIB$SIGNAL                     ;
             00000000G 00                00  FB 00140 11$:     CALLS   #0, DBG$PUSH_TEMPMEM               ; 1020
                       52                50  D0 00147          MOVL    R0, MARK                           ;
                                         53  DD 0014A          PUSHL   PRM_DESC                           ; 1021
             00000000G 00                01  FB 0014C          CALLS   #1, DBG$PRINT_IDENTIFIER           ;
                           00000000'     EF  9F 00153          PUSHAB  P.AAE                              ; 1022
                                         01  DD 00159          PUSHL   #1                                 ;
                           00000000'     EF  9F 0015B          PUSHAB  P.AAD                              ;
             00000000G 00                03  FB 00161          CALLS   #3, DBG$PRINT                      ;
                                   1C    AE  9F 00168          PUSHAB  VAL_DESC                           ; 1023
                       7E          7A    8F  9A 0016B          MOVZBL  #122, -(SP)                        ;
                                         53  DD 0016F          PUSHL   PRM_DESC                           ;
             00000000G 00                03  FB 00171          CALLS   #3, DBG$PRIM_TO_VAL                ;
                                         58  95 00178          TSTB    FORMAT_ONE                         ; 1024
                                         0A  13 0017A          BEQL    12$                                ;
                       50          1C    AE  D0 0017C          MOVL    VAL_DESC, R0                       ;
   05    A0            04          04    58  F0 00180          INSV    FORMAT_ONE, #4, #4, 5(R0)          ;
                           00000000G     00  DD 00186 12$:     PUSHL   DBG$GL_SIGN_FLAG                   ; 1025
                       7E                5A  9A 0018C          MOVZBL  RADIX, -(SP)                       ;
                                   24    AE  DD 0018F          PUSHL   VAL_DESC                           ;
             00000000G 00                03  FB 00192          CALLS   #3, DBG$PRINT_VALUE                ;
             00000000G 00                00  FB 00199          CALLS   #0, DBG$NEWLINE                    ; 1026
                                         53  DD 001A0          PUSHL   PRM_DESC                           ; 1027
             00000000G 00                01  FB 001A2          CALLS   #1, DBG$SAVE_LOC                   ;
                                         52  DD 001A9          PUSHL   MARK                               ; 1028
             00000000G 00                01  FB 001AB          CALLS   #1, DBG$POP_TEMPMEM                ;
                               0048      8F  BB 001B2          PUSHR   #^M<R3,R6>                         ; 1029
                 0000V CF                02  FB 001B6          CALLS   #2, PRIMARY_ORDER                  ;
                                         50  D5 001BB          TSTL    R0                                 ;
                                         03  19 001BD          BLSS    14$                                ;
                                       0270  31 001BF 13$:     BRW     36$                                ;
                                         7E  D4 001C2 14$:     CLRL    -(SP)                              ; 1030
                                         53  DD 001C4          PUSHL   PRM_DESC                           ;
                 0000V CF                02  FB 001C6          CALLS   #2, MODIFY_PRIMARY                 ;
                       F1                50  E9 001CB          BLBC    R0, 13$                            ;
                                       FF6F  31 001CE          BRW     11$                                ;
                       7E                6E  3C 001D1 15$:     MOVZWL  NEW_SIZE, -(SP)                    ; 1048
                       7E                59  9A 001D4          MOVZBL  NEW_TYPE, -(SP)                    ;
                                         56  DD 001D7          PUSHL   END_DESC                           ;
                  F8FC CF                03  FB 001D9          CALLS   #3, DBG$CHANGE_DTYPE               ;
                       52                50  D0 001DE          MOVL    R0, ADDR_DESC                      ;
                                   18    A2  DD 001E1          PUSHL   24(ADDR_DESC)                      ; 1049
             00000000G 00                01  FB 001E4          CALLS   #1, DBG$STA_ADDRESS_TO_REGDESCR    ;
                       54                50  D0 001EB          MOVL    R0, RDESC_ONE                      ;
                    0C AE          18    A2  D0 001EE          MOVL    24(ADDR_DESC), LAST_ADDR           ; 1050
                       7E                6E  3C 001F3          MOVZWL  NEW_SIZE, -(SP)                    ; 1051
```

```
                                     7E             59  9A 001F6         MOVZBL  NEW_TYPE, -(SP)                        ;
                                                    53  DD 001F9         PUSHL   PRM_DESC                               ;
                               F8DA  CF             03  FB 001FB         CALLS   #3, DBG$CHANGE_DTYPE                   ;
                                     52             50  D0 00200         MOVL    R0, ADDR_DESC                          ;
                                               18   A2  DD 00203         PUSHL   24(ADDR_DESC)                          ; 1052
                          00000000G  00             01  FB 00206         CALLS   #1, DBG$STA_ADDRESS_TO_REGDESCR        ;
                                     50             54  CC 0020D         XORL2   RDESC_ONE, R0                          ; 1053
                           FFFF00FC  8F             50  D3 00210         BITL    R0, #-65284                            ;
                                                    0D  13 00217         BEQL    16$                                    ;
                                         00028190   8F  DD 00219         PUSHL   #164240                                ; 1055
                          00000000G  00             01  FB 0021F         CALLS   #1, LIB$SIGNAL                         ;
                                 18  A2        0C   AE  D1 00226 16$:    CMPL    LAST_ADDR, 24(ADDR_DESC)               ; 1057
                                                    0D  1E 0022B         BGEQU   17$                                    ;
                                         00028190   8F  DD 0022D         PUSHL   #164240                                ; 1059
                          00000000G  00             01  FB 00233         CALLS   #1, LIB$SIGNAL                         ;
                                     54        14   A2  9E 0023A 17$:    MOVAB   20(ADDR_DESC), R4                      ; 1061
                                     0D        03   A4  91 0023E         CMPB    3(R4), #13                             ;
                                                    0D  12 00242         BNEQ    18$                                    ;
                                         000287D8   8F  DD 00244         PUSHL   #165848                                ; 1063
                          00000000G  00             01  FB 0024A         CALLS   #1, LIB$SIGNAL                         ;
                                 08  AE        7A   8F  9A 00251 18$:    MOVZBL  #122, DESC_TYPE                        ; 1065
                                     17        02   A4  91 00256         CMPB    2(R4), #23                             ; 1066
                                                    06  13 0025A         BEQL    19$                                    ;
                                     16        02   A4  91 0025C         CMPB    2(R4), #22                             ; 1067
                                                    05  12 00260         BNEQ    20$                                    ;
                                 08  AE        83   8F  9A 00262 19$:    MOVZBL  #131, DESC_TYPE                        ; 1069
                          00000000G  00             00  FB 00267 20$:    CALLS   #0, DBG$PUSH_TEMPMEM                   ; 1076
                                 14  AE             50  D0 0026E         MOVL    R0, MARK                               ;
                                                    52  DD 00272         PUSHL   ADDR_DESC                              ; 1077
                          00000000G  00             01  FB 00274         CALLS   #1, DBG$PRINT_IDENTIFIER               ;
                                        00000000'   EF  9F 0027B         PUSHAB  P.AAG                                  ; 1078
                                                    01  DD 00281         PUSHL   #1                                     ;
                                        00000000'   EF  9F 00283         PUSHAB  P.AAF                                  ;
                          00000000G  00             03  FB 00289         CALLS   #3, DBG$PRINT                          ;
                                               1C   AE  9F 00290         PUSHAB  VAL_DESC                               ; 1079
                                               0C   AE  DD 00293         PUSHL   DESC_TYPE                              ;
                                                    52  DD 00296         PUSHL   ADDR_DESC                              ;
                          00000000G  00             03  FB 00298         CALLS   #3, DBG$PRIM_TO_VAL                    ;
                                                    58  95 0029F         TSTB    FORMAT_ONE                             ; 1080
                                                    0A  13 002A1         BEQL    21$                                    ;
                                     50        1C   AE  D0 002A3         MOVL    VAL_DESC, R0                           ;
     05    A0           04           04             58  F0 002A7         INSV    FORMAT_ONE, #4, #4, 5(R0)              ;
                                        00000000G   00  DD 002AD 21$:    PUSHL   DBG$GL_SIGN_FLAG                       ; 1081
                                     7E             5A  9A 002B3         MOVZBL  RADIX, -(SP)                           ;
                                               24   AE  DD 002B6         PUSHL   VAL_DESC                               ;
                          00000000G  00             03  FB 002B9         CALLS   #3, DBG$PRINT_VALUE                    ;
                          00000000G  00             00  FB 002C0         CALLS   #0, DBG$NEWLINE                        ; 1082
                                               14   AE  DD 002C7         PUSHL   MARK                                   ; 1083
                          00000000G  00             01  FB 002CA         CALLS   #1, DBG$POP_TEMPMEM                    ;
                                                    54  DD 002D1         PUSHL   R4                                     ; 1089
                          00000000G  00             01  FB 002D3         CALLS   #1, DBG$DATA_LENGTH                    ;
                                     50             07  C0 002DA         ADDL2   #7, R0                                 ;
                   04    AE          50             08  C7 002DD         DIVL3   #8, R0, LENGTH                         ;
                                                    0F  12 002E2         BNEQ    22$                                    ; 1090
                                         000287B3   8F  DD 002E4         PUSHL   #165811                                ; 1093
                          00000000G  00             01  FB 002EA         CALLS   #1, LIB$SIGNAL                         ;
                                                    50  11 002F1         BRB     28$                                    ; 1092
```

```
          57        18  A2        04  AE  C1 002F3 22$:    ADDL3   LENGTH, 24(ADDR_DESC), NEXT_ADDR          ; 1097
                    0C  AE            57  D1 002F9         CMPL    NEXT_ADDR, LAST_ADDR                      ; 1098
                                      44  1A 002FD         BGTRU   28$                                       ;
                    18  A2            57  D0 002FF         MOVL    NEXT_ADDR, 24(ADDR_DESC)                  ; 1099
                        17        02  A4  91 00303         CMPB    2(R4), #23                                ; 1100
                                      06  13 00307         BEQL    23$                                       ;
                        16        02  A4  91 00309         CMPB    2(R4), #22                                ; 1101
                                      27  12 0030D         BNEQ    26$                                       ;
                                      57  DD 0030F 23$:    PUSHL   NEXT_ADDR                                 ; 1104
               00000000G  00          01  FB 00311         CALLS   #1, DBG$IS_IT_ENTRY                       ;
                          06          50  E9 00318         BLBC    R0, 24$                                   ;
                    02  A4            17  90 0031B         MOVB    #23, 2(R4)                                ; 1105
                                      04  11 0031F         BRB     25$                                       ;
                    02  A4            16  90 00321 24$:    MOVB    #22, 2(R4)                                ; 1106
                                      7E  D4 00325 25$:    CLRL    -(SP)                                     ; 1107
                                      57  DD 00327         PUSHL   NEXT_ADDR                                 ;
               00000000G  00          02  FB 00329         CALLS   #2, DBG$INS_DECODE                        ;
          64              50          57  A3 00330         SUBW3   NEXT_ADDR, R0, (R4)                       ;
                                      0A  11 00334         BRB     27$                                       ; 1100
                                      54  DD 00336 26$:    PUSHL   R4                                        ; 1110
                   0000V  CF          01  FB 00338         CALLS   #1, FIX_UP_LENGTH                         ;
                          64          50  B0 0033D         MOVW    R0, (R4)                                  ;
                                    FF24  31 00340 27$:    BRW     20$                                       ; 1074
                                      52  DD 00343 28$:    PUSHL   ADDR_DESC                                 ; 1112
               00000000G  00          01  FB 00345         CALLS   #1, DBG$SAVE_LOC                          ;
                                      2B  11 0034C         BRB     30$                                       ; 0984
                                      53  DD 0034E 29$:    PUSHL   PRM_DESC                                  ; 1126
               00000000G  00          01  FB 00350         CALLS   #1, DBG$PRINT_IDENTIFIER                  ;
                          53          50  D0 00357         MOVL    R0, PRM_DESC                              ;
                                      53  DD 0035A         PUSHL   PRM_DESC                                  ; 1127
               00000000G  00          01  FB 0035C         CALLS   #1, DBG$SAVE_LOC                          ;
                      80  8F          59  91 00363         CMPB    NEW_TYPE, #128                            ; 1128
                                      13  12 00367         BNEQ    31$                                       ;
                          0F      04  A3  E9 00369         BLBC    4(PRM_DESC), 31$                          ;
                          7E          5A  9A 0036D         MOVZBL  RADIX, -(SP)                              ; 1130
                                      53  DD 00370         PUSHL   PRM_DESC                                  ;
               00000000G  00          02  FB 00372         CALLS   #2, DBG$PRINT_AGGREGATE                   ;
                                    00B6  31 00379 30$:    BRW     36$                                       ;
                          00000000'   EF  9F 0037C 31$:    PUSHAB  P.AAI                                     ; 1133
                                      01  DD 00382         PUSHL   #1                                        ;
                          00000000'   EF  9F 00384         PUSHAB  P.AAH                                     ;
               00000000G  00          03  FB 0038A         CALLS   #3, DBG$PRINT                             ;
                          7E          6E  3C 00391         MOVZWL  NEW_SIZE, -(SP)                           ; 1134
                          7E          59  9A 00394         MOVZBL  NEW_TYPE, -(SP)                           ;
                                      53  DD 00397         PUSHL   PRM_DESC                                  ;
                    F73C  CF          03  FB 00399         CALLS   #3, DBG$CHANGE_DTYPE                      ;
                      1C  AE          50  D0 0039E         MOVL    R0, VAL_DESC                              ;
                      10  AE          58  90 003A2         MOVB    FORMAT_ONE, FORMAT_TWO                    ; 1135
                      80  8F          59  91 003A6         CMPB    NEW_TYPE, #128                            ; 1136
                                      10  13 003AA         BEQL    32$                                       ;
          7E          1C  AE          14  C1 003AC         ADDL3   #20, VAL_DESC, -(SP)                      ; 1138
                                      53  DD 003B1         PUSHL   PRM_DESC                                  ;
               00000000G  00          02  FB 003B3         CALLS   #2, DBG$SAVE_LOC                          ;
                                      28  11 003BA         BRB     33$                                       ;
                                      58  95 003BC 32$:    TSTB    FORMAT_ONE                                ; 1139
                                      24  12 003BE         BNEQ    33$                                       ;
                          01          5A  91 003C0         CMPB    RADIX, #1                                 ;
```

```
                                             1F  12 003C3        BNEQ    33$                                      :
                              50        1C   AE  D0 003C5        MOVL    VAL_DESC, R0                             : 1140
                              0D        17   A0  91 003C9        CMPB    23(R0), #13                              :
                                             15  13 003CD        BEQL    33$                                      :
                              50        1C   AE  D0 003CF        MOVL    VAL_DESC, R0                             : 1141
                              51 00000000G   00  9E 003D3        MOVAB   DBG$REG_VALUES+64, R1                    :
                              51        18   A0  D1 003DA        CMPL    24(R0), R1                               :
                                             04  12 003DE        BNEQ    33$                                      :
                          10  AE             02  90 003E0        MOVB    #2, FORMAT_TWO                           : 1142
                              50        1C   AE  D0 003E4 33$:   MOVL    VAL_DESC, R0                             : 1144
                              16        16   A0  91 003E8        CMPB    22(R0), #22                              :
                                             1A  13 003EC        BEQL    34$                                      :
                                        1C   AE  9F 003EE        PUSHAB  VAL_DESC                                 : 1147
                              7E        7A   8F  9A 003F1        MOVZBL  #122, -(SP)                              :
                                             50  DD 003F5        PUSHL   R0                                       :
                    00000000G 00             03  FB 003F7        CALLS   #3, DBG$PRIM_TO_VAL                      :
                                        1C   AE  DD 003FE        PUSHL   VAL_DESC                                 : 1148
                    00000000G 00             01  FB 00401        CALLS   #1, DBG$DO_MAPPING                       :
                                        10   AE  95 00408 34$:   TSTB    FORMAT_TWO                               : 1151
                                             0B  13 0040B        BEQL    35$                                      :
                              50        1C   AE  D0 0040D        MOVL    VAL_DESC, R0                             :
    05   A0            04     04        10   AE  F0 00411        INSV    FORMAT_TWO, #4, #4, 5(R0)                :
                                 00000000G   00  DD 00418 35$:   PUSHL   DBG$GL_SIGN_FLAG                         : 1152
                              7E             5A  9A 0041E        MOVZBL  RADIX, -(SP)                             :
                                        24   AE  DD 00421        PUSHL   VAL_DESC                                 :
                    00000000G 00             03  FB 00424        CALLS   #3, DBG$PRINT_VALUE                      :
                    00000000G 00             00  FB 0042B        CALLS   #0, DBG$NEWLINE                          : 1153
                              55        08   A5  D0 00432 36$:   MOVL    8(NOUN_NODE), NOUN_NODE                  : 1157
                                             03  13 00436        BEQL    37$                                      :
                                           FC48  31 00438        BRW     5$                                       :
                                                 04 0043B 37$:   RET                                              : 1160

; Routine Size:  1084 bytes,    Routine Base:  DBG$CODE + 0526
```

```
1 GLOBAL ROUTINE DBG$NEXTLOC(PRM_DESC) =
1 !
1 ! FUNCTION
1 !         ----------------------------
1 !
1 ! INPUTS
1 !         ----------------------------
1 !
1 ! OUTPUTS
1 !         ----------------------------
1 !
1
2     BEGIN
2
2     MAP
2         PRM_DESC: REF DBG$PRIMARY;        ! Pointer to Primary Descriptor
2
2     LOCAL
2         BYTE_OFFSET,
2         LENGTH,
2         REG_DESC: DBG$REGDESCR,
2         STATUS,
2         VAL_DESC: REF DBG$VALDESC;
2
2
2
2     ! ------
2     !
2     STATUS = MODIFY_PRIMARY(.PRM_DESC,0);
2     IF .STATUS THEN RETURN .PRM_DESC;
2     IF .DBG$GL_CURLOC_VMSDESC NEQ 0
2     THEN
3         BEGIN
3         VAL_DESC = DBG$MAKE_VAL_DESC(.DBG$GL_CURLOC_VMSDESC, DBG$K_V_VALUE_DESC);
3         VAL_DESC[DBG$B_DHDR_LANG]    = .PRM_DESC[DBG$B_DHDR_LANG];
3         VAL_DESC[DBG$L_DHDR_SYMID0] = .PRM_DESC[DBG$L_DHDR_SYMID0];
3         END
3
2     ELSE
3         BEGIN
3         IF .STATUS EQL 2 THEN SIGNAL(DBG$_NOSUCC);
3         DBG$PRIM_TO_VAL(.PRM_DESC, DBG$K_V_VALUE_DESC, VAL_DESC);
3         END;
2
3     IF (.VAL_DESC[DBG$B_VALUE_CLASS] EQL DSC$K_CLASS_UBS)
2     THEN
2         SIGNAL(DBG$_NOSUCC);
2
2     LENGTH = (DBG$DATA_LENGTH(VAL_DESC[DBG$A_VALUE_VMSDESC]) - 1)/%BPUNIT + 1;
2     REG_DESC = DBG$STA_ADDRESS_TO_REGDESCR(.VAL_DESC[DBG$L_VALUE_POINTER]);
2     IF .REG_DESC NEQ 0
2     THEN
3         BEGIN
3         BYTE_OFFSET = 4*.REG_DESC[DBG$B_REGD_REGNUM]
3                     + .REG_DESC[DBG$V_REGD_OFFSET]
3                     + .LENGTH + .DBG$GW_DFLTLENG;
3         IF (.BYTE_OFFSET GTR 16*%UPVAL) AND
```

```
4            ((.DBG$GW_DFLTLENG NEQ 2) OR (.DBG$GW_DFLTLENG NEQ 4) OR
4            (.BYTE_OFFSET NEQ (16*%UPVAL + .DBG$GW_DFLTLENG)))
3        THEN
3            SIGNAL(DBG$_NOSUCC);
3
2        END;
2
2
2    ! Initialize the Value Descriptor to VMS descriptor class Z (unknown) and
2    ! set the pointer to the next location to be the current location plus the
2    ! length of the current object.
2    !
2    VAL_DESC[DBG$B_VALUE_CLASS]   = DSC$K_CLASS_Z;
2    VAL_DESC[DBG$L_VALUE_POINTER] = .VAL_DESC[DBG$L_VALUE_POINTER] + .LENGTH;
2
2
2    ! If the data type is instruction or entry point, determine the type of the
2    ! next location by seeing if it is an instruction or entry mask.  Also com-
2    ! pute its length by interpreting the instruction at that location.
2    !
2    IF (.VAL_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_ZI) OR
3       (.VAL_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_ZEM)
2    THEN
3        BEGIN
3        IF DBG$IS_IT_ENTRY(.VAL_DESC[DBG$L_VALUE_POINTER])
3        THEN
3            VAL_DESC[DBG$B_VALUE_DTYPE]  = DSC$K_DTYPE_ZEM
3
3        ELSE
3            VAL_DESC[DBG$B_VALUE_DTYPE]  = DSC$K_DTYPE_ZI;
3
3        VAL_DESC[DBG$W_VALUE_LENGTH] =
3                        DBG$INS_DECODE(.VAL_DESC[DBG$L_VALUE_POINTER], FALSE) -
3                        .VAL_DESC[DBG$L_VALUE_POINTER];
3        END
3
3
3    ! The next location is not an instruction or entry mask.  It is thus a
3    ! data object and we set up the Value Descriptor accordingly.
3    !
2    ELSE
3        BEGIN
3        VAL_DESC[DBG$B_DHDR_FCODE]   =  RST$K_TYPE_DESCR;
3        VAL_DESC[DBG$V_DHDR_FORMAT]  =  0;
3        VAL_DESC[DBG$B_VALUE_CLASS]  =  DSC$K_CLASS_S;
3        VAL_DESC[DBG$B_VALUE_DTYPE]  = .DBG$GL_DFLTTYP;
3        VAL_DESC[DBG$W_VALUE_LENGTH] = .DBG$GW_DFLTLENG;
2        END;
2
2
2    ! Return a pointer to the Value Descriptor for the next location.
2    !
2    RETURN .VAL_DESC;
2
1    END;
```

```
                                            007C 00000       .ENTRY  DBG$NEXTLOC, Save R2,R3,R4,R5,R6   ; 1161
                          56  00000000G    00 9E 00002       MOVAB   DBG$GW_DFLTLENG, R6                ;
                          55  00000000G    00 9E 00009       MOVAB   LIB$SIGNAL, R5                     ;
                          5E               04 C2 00010       SUBL2   #4, SP                             ;
                                           7E D4 00013       CLRL    -(SP)                              ; 1189
                          52         04    AC D0 00015       MOVL    PRM_DESC, R2                       ;
                                           52 DD 00019       PUSHL   R2                                 ;
                   0000V  CF               02 FB 0001B       CALLS   #2, MODIFY_PRIMARY                 ;
                          03               50 E9 00020       BLBC    STATUS, 1$                         ; 1190
                                         010A 31 00023       BRW     12$                                ;
                          51  00000000G    00 D0 00026 1$:   MOVL    DBG$GL_CURLOC_VMSDESC, R1          ; 1191
                                           1C 13 0002D       BEQL    2$                                 ;
                          7E         83    8F 9A 0002F       MOVZBL  #131, -(SP)                        ; 1194
                                           51 DD 00033       PUSHL   R1                                 ;
               00000000G  00               02 FB 00035       CALLS   #2, DBG$MAKE_VAL_DESC              ;
                          6E               50 D0 0003C       MOVL    R0, VAL_DESC                       ;
                      03  A0         03    A2 90 0003F       MOVB    3(R2), 3(R0)                       ; 1195
                      0C  A0         0C    A2 D0 00044       MOVL    12(R2), 12(R0)                     ; 1196
                                           1D 11 00049       BRB     4$                                 ; 1191
                          02               50 D1 0004B 2$:   CMPL    STATUS, #2                         ; 1201
                                           09 12 0004E       BNEQ    3$                                 ;
                               00028818    8F DD 00050       PUSHL   #165912                            ;
                          65               01 FB 00056       CALLS   #1, LIB$SIGNAL                     ;
                                           5E DD 00059 3$:   PUSHL   SP                                 ; 1202
                          7E         83    8F 9A 0005B       MOVZBL  #131, -(SP)                        ;
                                           52 DD 0005F       PUSHL   R2                                 ;
               00000000G  00               03 FB 00061       CALLS   #3, DBG$PRIM_TO_VAL                ;
                          52               6E D0 00068 4$:   MOVL    VAL_DESC, R2                       ; 1205
                          53         14    A2 9E 0006B       MOVAB   20(R2), R3                         ;
                          0D         03    A3 91 0006F       CMPB    3(R3), #13                         ;
                                           09 12 00073       BNEQ    5$                                 ;
                               00028818    8F DD 00075       PUSHL   #165912                            ; 1207
                          65               01 FB 0007B       CALLS   #1, LIB$SIGNAL                     ;
                                           53 DD 0007E 5$:   PUSHL   R3                                 ; 1209
               00000000G  00               01 FB 00080       CALLS   #1, DBG$DATA_LENGTH                ;
                                           50 D7 00087       DECL    R0                                 ;
                          50               08 C6 00089       DIVL2   #8, R0                             ;
                          54         01    A0 9E 0008C       MOVAB   1(R0), LENGTH                      ;
                                     18    A2 DD 00090       PUSHL   24(R2)                             ; 1210
               00000000G  00               01 FB 00093       CALLS   #1, DBG$STA_ADDRESS_TO_REGDESCR    ;
                                           50 D5 0009A       TSTL    REG_DESC                           ; 1211
                                           3D 13 0009C       BEQL    7$                                 ;
51        50              08               08 EF 0009E       EXTZV   #8, #8, REG_DESC, R1               ; 1214
50        50              02               00 EF 000A3       EXTZV   #0, #2, REG_DESC, R0               ; 1215
                          50             6041 DE 000A8       MOVAL   (R0)[R1], R0                       ;
          51              50               54 C1 000AC       ADDL3   LENGTH, R0, R1                     ; 1216
                          50               66 3C 000B0       MOVZWL  DBG$GW_DFLTLENG, R0                ;
                          51               50 C0 000B3       ADDL2   R0, BYTE_OFFSET                    ;
                00000040  8F               51 D1 000B6       CMPL    BYTE_OFFSET, #64                   ; 1217
                                           1C 15 000BD       BLEQ    7$                                 ;
                          02               50 B1 000BF       CMPW    R0, #2                             ; 1218
                                           0E 12 000C2       BNEQ    6$                                 ;
                          04               50 B1 000C4       CMPW    R0, #4                             ;
                                           09 12 000C7       BNEQ    6$                                 ;
```

```
                              50        40   A0   9E 000C9          MOVAB   64(R0), R0                          ; 1219
                              50             51   D1 000CD          CMPL    BYTE_OFFSET, R0                     ;
                                             09   13 000D0          BEQL    7$                                  ;
                                  00028818   8F   DD 000D2 6$:      PUSHL   #165912                             ; 1221
                              65             01   FB 000D8          CALLS   #1, LIB$SIGNAL                      ;
                                        03   A3   94 000DB 7$:      CLRB    3(R3)                               ; 1230
                    18        A2             54   C0 000DE          ADDL2   LENGTH, 24(R2)                      ; 1231
                              16        02   A3   91 000E2          CMPB    2(R3), #22                          ; 1238
                                             06   13 000E6          BEQL    8$                                  ;
                              17        02   A3   91 000E8          CMPB    2(R3), #23                          ; 1239
                                             2A   12 000EC          BNEQ    11$                                 ;
                                        18   A2   DD 000EE 8$:      PUSHL   24(R2)                              ; 1242
               00000000G      00             01   FB 000F1          CALLS   #1, DBG$IS_IT_ENTRY                 ;
                              06             50   E9 000F8          BLBC    R0, 9$                              ;
                    02        A3             17   90 000FB          MOVB    #23, 2(R3)                          ; 1244
                                             04   11 000FF          BRB     10$                                 ;
                    02        A3             16   90 00101 9$:      MOVB    #22, 2(R3)                          ; 1247
                                             7E   D4 00105 10$:     CLRL    -(SP)                               ; 1250
                                        18   A2   DD 00107          PUSHL   24(R2)                              ;
               00000000G      00             02   FB 0010A          CALLS   #2, DBG$INS_DECODE                  ;
          63                  50        18   A2   A3 00111          SUBW3   24(R2), R0, (R3)                    ; 1251
                                             18   11 00116          BRB     12$                                 ; 1238
                    06        A2             03   90 00118 11$:     MOVB    #3, 6(R2)                           ; 1260
                    05        A2        F0   8F   8A 0011C          BICB2   #240, 5(R2)                         ; 1261
                    03        A3             01   90 00121          MOVB    #1, 3(R3)                           ; 1262
                    02        A3 00000000G   00   90 00125          MOVB    DBG$GL_DFLTTYP, 2(R3)               ; 1263
                              63             66   B0 0012D          MOVW    DBG$GW_DFLTLENG, (R3)               ; 1264
                              50             52   D0 00130 12$:     MOVL    R2, R0                              ; 1270
                                                  04 00133          RET                                         ; 1272

; Routine Size:  308 bytes,    Routine Base:  DBG$CODE + 0962
```

```
GLOBAL ROUTINE DBG$PREVLOC(PRM_DESC) =
!
! FUNCTION
!       This routine finds the "previous location", denoted in the command
!       language as %PREVLOC or ^.  It accepts a Primary Descriptor for the
!       current location as input and returns either a Primary Descriptor or
!       a Volatile Value Descriptor for the previous location as output.  If
!       the current location is a structured object of some sort (like an
!       array), MODIFY_PRIMARY is called to find the logical predecessor and
!       the modified Primary Descriptor is returned.  Otherwise, this routine
!       determines the previous instruction location or the previous data
!       location and returns a Volatile Value Descriptor for that location.
!
! INPUTS
!       PRM_DESC - A pointer to the input Primary Descriptor for the location
!                  whose logical predecessor is to be computed.
!
! OUTPUTS
!       A pointer to the Primary Descriptor or Volatile Value Descriptor for
!                  the logical predecessor location is returned as this
!                  routine's value.
!

    BEGIN

    MAP
        PRM_DESC: REF DBG$PRIMARY;          ! Pointer to Primary Descriptor

    LOCAL
        ADDRESS,                            ! Address of the current location - 1
        DUMMY,                              ! Dummy routine argument
        LENGTH,
        LINE,                               ! Line number of the last instruction
        NEW_ADDR,                           ! Addess of the current instruction
        OLD_ADDR,                           ! Address of previous instruction
        PC_BEG,                             ! Beginning PC of current source line
        PC_END,                             ! Ending PC of current source line
        STATUS,
        STMT,                               ! Statement number of last instruction
        SYMID: REF RST$ENTRY,               ! The SYMID of the nearest preceding
                                            !       symbol (used for instructions)
        REG_DESC: DBG$REGDESCR,
        VAL_DESC: REF DBG$VALDESC;          ! Pointer to returned Value Descriptor



    ! If the input Primary Descriptor describes a structure object, like an
    ! array or record, let MODIFY_PRIMARY modify the Primary Descriptor to
    ! describe the logical predecessor.  Then return that Primary.
    !
    STATUS = MODIFY_PRIMARY(.PRM_DESC, 1);
    IF .STATUS THEN RETURN .PRM_DESC;


    ! If there is a defined "current location" (%CURLOC), then use the Primary
    ! or Value Descriptor for that entity to set up a Volatile Value Descriptor
    ! for the previous location.
```

```
2        !
2        IF .DBG$GL_CURLOC_VMSDESC NEQ 0
2        THEN
3            BEGIN
3            VAL_DESC = DBG$MAKE_VAL_DESC(.DBG$GL_CURLOC_VMSDESC, DBG$K_V_VALUE_DESC);
3            VAL_DESC[DBG$B_DHDR_LANG]    = .PRM_DESC[DBG$B_DHDR_LANG];
3            VAL_DESC[DBG$L_DHDR_SYMID0] = .PRM_DESC[DBG$L_DHDR_SYMID0];
3            END
3
3
3        ! But if no current location is defined, give an error message or use the
3        ! input Primary Descriptor to set up the previous location descriptor.
3        !
2        ELSE
3            BEGIN
3            IF .STATUS EQL 2 THEN SIGNAL(DBG$_NOPRED);
3            DBG$PRIM_TO_VAL(.PRM_DESC, DBG$K_V_VALUE_DESC, VAL_DESC);
3            END;
2
2
2        ! There is no logical successor for an unaligned bit string, so for that
2        ! case we signal an error.
2        !
2        IF .VAL_DESC[DBG$B_VALUE_CLASS] EQL DSC$K_CLASS_UBS
2        THEN
2            SIGNAL(DBG$_NOPRED);
2
2
2        ! -----
2        !
2        LENGTH = (DBG$DATA_LENGTH(VAL_DESC[DBG$A_VALUE_VMSDESC]) - 1)/%BPUNIT + 1;
2        REG_DESC = DBG$STA_ADDRESS_TO_REGDESCR(.VAL_DESC[DBG$L_VALUE_POINTER]);
3        IF (.REG_DESC NEQ 0) AND (.REG_DESC<W_> LSSU (%X'00B4' + .DBG$GW_DFLTLENG))
2        THEN
2            SIGNAL(DBG$_NOPRED);
2
2
2        ! Initialize the DTYPE of the logical predecessor to be type Z (unknown)
2        ! and assume its address is one byte before the current location.  This
2        ! may get changed below if appropriate.
2        !
2        VAL_DESC[DBG$B_VALUE_CLASS] = DSC$K_CLASS_Z;
2        ADDRESS = .VAL_DESC[DBG$L_VALUE_POINTER] - 1;
2
2
2        ! If the type of the current object is instruction or entry mask, try to
2        ! locate the previous instruction.
2        !
2        IF (.VAL_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_ZI) OR
3           (.VAL_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_ZEM)
2        THEN
3            BEGIN
3            OLD_ADDR = .ADDRESS;
3
3
3            ! First try to symbolize the current location - 1 byte (the contents of
3            ! ADDRESS) to find the nearest routine, block, or label preceding the
```

```
! current instruction.  If no such symbol is found, we just leave the
! address as the current instruction address - 1; there is no way we can
! locate the true previous instruction.
!
STATUS = DBG$PC_TO_SYMID(.ADDRESS, SYMID);
IF .STATUS AND (.SYMID NEQ 0)
THEN
    BEGIN


    ! A symbol preceding the current instruction was found.  If this
    ! is an instruction symbol (a routine, block, or label), save its
    ! address for the forward scan to the desired instruction.  However,
    ! if a line number preceding the current instruction can be found,
    ! use that address instead for a shorter forward scan.
    !
    IF (.SYMID[RST$B_KIND] EQL RST$K_ROUTINE) OR
       (.SYMID[RST$B_KIND] EQL RST$K_BLOCK)   OR
       (.SYMID[RST$B_KIND] EQL RST$K_LABEL)
    THEN
        BEGIN
        OLD_ADDR = .SYMID[RST$L_STARTADDR];
        DUMMY = .SYMID;
        IF DBG$PC_TO_LINE_LOOKUP(.ADDRESS,
                                LINE, STMT, PC_BEG, PC_END, DUMMY)
        THEN
            OLD_ADDR = .PC_BEG;

        END;

    END;                            ! End of code if we found a symbolization


! We now some address where to start the forward scan that looks for
! the previous instruction.  Scan forward from that address until the
! desired previous instruction is found.
!
WHILE TRUE DO
    BEGIN
    NEW_ADDR = DBG$INS_DECODE(.OLD_ADDR, FALSE);
    IF .NEW_ADDR GTRA .ADDRESS THEN EXITLOOP;
    OLD_ADDR = .NEW_ADDR;
    END;


! Fill the address and length of the found previous instruction into
! the Value Descriptor.  Also determine if this location is an entry
! mask--if so, set the DTYPE to be ZEM instead of ZI.
!
VAL_DESC[DBG$L_VALUE_POINTER] = .OLD_ADDR;
VAL_DESC[DBG$W_VALUE_LENGTH]  = .NEW_ADDR - .OLD_ADDR;
VAL_DESC[DBG$B_VALUE_DTYPE] = DSC$K_DTYPE_ZI;
IF DBG$IS_IT_ENTRY(.OLD_ADDR)
THEN
    VAL_DESC[DBG$B_VALUE_DTYPE] = DSC$K_DTYPE_ZEM;

END                                 ! End of code for previous instruction
```

```
; 1320      1444  3
; 1321      1445  3
; 1322      1446  3          ! The current location is not an instruction location.  We therefore set
; 1323      1447  3          ! up the Value Descriptor for a data object.
; 1324      1448  3          !
; 1325      1449  2          ELSE
; 1326      1450  3              BEGIN
; 1327      1451  3              VAL_DESC[DBG$B_DHDR_FCODE]    =  RST$K_TYPE_DESCR;
; 1328      1452  3              VAL_DESC[DBG$V_DHDR_FORMAT]   =  0;
; 1329      1453  3              VAL_DESC[DBG$B_VALUE_CLASS]   =  DSC$K_CLASS_S;
; 1330      1454  3              VAL_DESC[DBG$B_VALUE_DTYPE]   = .DBG$GL_DFLTTYP;
; 1331      1455  3              VAL_DESC[DBG$W_VALUE_LENGTH]  = .DBG$GW_DFLTLENG;
; 1332      1456  3              VAL_DESC[DBG$L_VALUE_POINTER] = .VAL_DESC[DBG$L_VALUE_POINTER] - .LENGTH;
; 1333      1457  2              END;
; 1334      1458  2
; 1335      1459  2
; 1336      1460  2          ! Return a pointer to the Volatile Value Descriptor for the previous
; 1337      1461  2          ! location.
; 1338      1462  2          !
; 1339      1463  2          RETURN .VAL_DESC;
; 1340      1464  2
; 1341      1465  1          END;



                                      03FC 00000        .ENTRY  DBG$PREVLOC, Save R2,R3,R4,R5,R6,R7,R8,R9      ; 1273
                59 00000000G    00  9E 00002        MOVAB   DBG$GW_DFLTLENG, R9                            ;
                58 00000000G    00  9E 00009        MOVAB   LIB$SIGNAL, R8                                 ;
                5E              1C  C2 00010        SUBL2   #28, SP                                        ;
                                01  DD 00013        PUSHL   #1                                             ; 1323
                52         04   AC  D0 00015        MOVL    PRM_DESC, R2                                   ;
                                52  DD 00019        PUSHL   R2                                             ;
          0000V CF              02  FB 0001B        CALLS   #2, MODIFY_PRIMARY                             ;
                57              50  D0 00020        MOVL    R0, STATUS                                     ;
                03              57  E9 00023        BLBC    STATUS, 1$                                     ; 1324
                              014D  31 00026        BRW     12$                                            ;
                50 00000000G    00  D0 00029 1$:    MOVL    DBG$GL_CURLOC_VMSDESC, R0                      ; 1331
                                1C  13 00030        BEQL    2$                                             ;
                7E         83   8F  9A 00032        MOVZBL  #131, -(SP)                                    ; 1334
                                50  DD 00036        PUSHL   R0                                             ;
      00000000G 00              02  FB 00038        CALLS   #2, DBG$MAKE_VAL_DESC                          ;
                6E              50  D0 0003F        MOVL    R0, VAL_DESC                                   ;
             03 A0         03   A2  90 00042        MOVB    3(R2), 3(R0)                                   ; 1335
             0C A0         0C   A2  D0 00047        MOVL    12(R2), 12(R0)                                 ; 1336
                                1D  11 0004C        BRB     4$                                             ; 1331
                02              57  D1 0004E 2$:    CMPL    STATUS, #2                                     ; 1345
                                09  12 00051        BNEQ    3$                                             ;
                   00028810     8F  DD 00053        PUSHL   #165904                                        ;
                68              01  FB 00059        CALLS   #1, LIB$SIGNAL                                 ;
                                5E  DD 0005C 3$:    PUSHL   SP                                             ; 1346
                7E         83   8F  9A 0005E        MOVZBL  #131, -(SP)                                    ;
                                52  DD 00062        PUSHL   R2                                             ;
      00000000G 00              03  FB 00064        CALLS   #3, DBG$PRIM_TO_VAL                            ;
                52              6E  D0 0006B 4$:    MOVL    VAL_DESC, R2                                   ; 1353
                54         14   A2  9E 0006E        MOVAB   20(R2), R4                                     ;
```

```
                              0D      03   A4  91 00072            CMPB    3(R4), #13                              ;
                                           09  12 00076            BNEQ    5$                                      ;
                                     00028810  8F  DD 00078        PUSHL   #165904                                 ; 1355
                              68           01  FB 0007E            CALLS   #1, LIB$SIGNAL                          ;
                                           54  DD 00081  5$:       PUSHL   R4                                      ; 1360
                     00000000G  00         01  FB 00083            CALLS   #1, DBG$DATA_LENGTH                     ;
                                           50  D7 0008A            DECL    R0                                      ;
                              50           08  C6 0008C            DIVL2   #8, R0                                  ;
                              56      01   A0  9E 0008F            MOVAB   1(R0), LENGTH                           ;
                                      18   A2  DD 00093            PUSHL   24(R2)                                  ; 1361
                     00000000G  00         01  FB 00096            CALLS   #1, DBG$STA_ADDRESS_TO_REGDESCR         ;
                                           50  D5 0009D            TSTL    REG_DESC                                ; 1362
                                           18  13 0009F            BEQL    6$                                      ;
                              51           69  3C 000A1            MOVZWL  DBG$GW_DFLTLENG, R1                     ;
                              51     00B4  C1  9E 000A4            MOVAB   180(R1), R1                             ;
         51           50      10           00  ED 000A9            CMPZV   #0, #16, REG_DESC, R1                   ;
                                           09  1E 000AE            BGEQU   6$                                      ;
                                     00028810  8F  DD 000B0        PUSHL   #165904                                 ; 1364
                              68           01  FB 000B6            CALLS   #1, LIB$SIGNAL                          ;
                                      03   A4  94 000B9  6$:       CLRB    3(R4)                                   ; 1371
                      55      18      A2   01  C3 000BC            SUBL3   #1, 24(R2), ADDRESS                     ; 1372
                              16      02   A4  91 000C1            CMPB    2(R4), #22                              ; 1378
                                           09  13 000C5            BEQL    7$                                      ;
                              17      02   A4  91 000C7            CMPB    2(R4), #23                              ; 1379
                                           03  13 000CB            BEQL    7$                                      ;
                                         008A  31 000CD            BRW     11$                                     ;
                              53           55  D0 000D0  7$:       MOVL    ADDRESS, OLD_ADDR                       ; 1382
                                      04   AE  9F 000D3            PUSHAB  SYMID                                   ; 1391
                                           55  DD 000D6            PUSHL   ADDRESS                                 ;
                     00000000G  00         02  FB 000D8            CALLS   #2, DBG$PC_TO_SYMID                     ;
                              57           50  D0 000DF            MOVL    R0, STATUS                              ;
                              42           57  E9 000E2            BLBC    STATUS, 9$                              ; 1392
                                      04   AE  D5 000E5            TSTL    SYMID                                   ;
                                           3D  13 000E8            BEQL    9$                                      ;
                              50      04   AE  D0 000EA            MOVL    SYMID, R0                               ; 1403
                              02      14   A0  91 000EE            CMPB    20(R0), #2                              ;
                                           0C  13 000F2            BEQL    8$                                      ;
                              03      14   A0  91 000F4            CMPB    20(R0), #3                              ; 1404
                                           06  13 000F8            BEQL    8$                                      ;
                              04      14   A0  91 000FA            CMPB    20(R0), #4                              ; 1405
                                           27  12 000FE            BNEQ    9$                                      ;
                              53      18   A0  D0 00100  8$:       MOVL    24(R0), OLD_ADDR                        ; 1408
                              08      AE   50  D0 00104            MOVL    R0, DUMMY                               ; 1409
                                      08   AE  9F 00108            PUSHAB  DUMMY                                   ; 1410
                                      10   AE  9F 0010B            PUSHAB  PC_END                                  ;
                                      18   AE  9F 0010E            PUSHAB  PC_BEG                                  ;
                                      20   AE  9F 00111            PUSHAB  STMT                                    ;
                                      28   AE  9F 00114            PUSHAB  LINE                                    ;
                                           55  DD 00117            PUSHL   ADDRESS                                 ;
                     00000000G  00         06  FB 00119            CALLS   #6, DBG$PC_TO_LINE_LOOKUP               ;
                              04           50  E9 00120            BLBC    R0, 9$                                  ;
                              53      10   AE  D0 00123            MOVL    PC_BEG, OLD_ADDR                        ; 1413
                                           7E  D4 00127  9$:       CLRL    -(SP)                                   ; 1426
                                           53  DD 00129            PUSHL   OLD_ADDR                                ;
                     00000000G  00         02  FB 0012B            CALLS   #2, DBG$INS_DECODE                      ;
                              55           50  D1 00132            CMPL    NEW_ADDR, ADDRESS                       ; 1427
                                           05  1A 00135            BGTRU   10$                                     ;
```

```
                53          50  D0 00137          MOVL    NEW_ADDR, OLD_ADDR                          ; 1428
                            EB  11 0013A          BRB     9$                                          ; 1424
            18  A2          53  D0 0013C  10$:    MOVL    OLD_ADDR, 24(R2)                            ; 1436
    64          50          53  A3 00140          SUBW3   OLD_ADDR, NEW_ADDR, (R4)                    ; 1437
            02  A4          16  90 00144          MOVB    #22, 2(R4)                                  ; 1438
                            53  DD 00148          PUSHL   OLD_ADDR                                    ; 1439
     00000000G  00          01  FB 0014A          CALLS   #1, DBG$IS_IT_ENTRY                         ;
                22          50  E9 00151          BLBC    R0, 12$                                     ;
            02  A4          17  90 00154          MOVB    #23, 2(R4)                                  ; 1441
                            1C  11 00158          BRB     12$                                         ; 1378
            06  A2          03  90 0015A  11$:    MOVB    #3, 6(R2)                                   ; 1451
            05  A2      F0  8F  8A 0015E          BICB2   #240, 5(R2)                                 ; 1452
            03  A4          01  90 00163          MOVB    #1, 3(R4)                                   ; 1453
            02  A4 00000000G 00 90 00167          MOVB    DBG$GL_DFLTTYP, 2(R4)                       ; 1454
                64          69  B0 0016F          MOVW    DBG$GW_DFLTLENG, (R4)                       ; 1455
            18  A2          56  C2 00172          SUBL2   LENGTH, 24(R2)                              ; 1456
                50          52  D0 00176  12$:    MOVL    R2, R0                                      ; 1463
                                04 00179          RET                                                 ; 1465

; Routine Size:  378 bytes,    Routine Base:  DBG$CODE + 0A96
```

```
1 ROUTINE MODIFY_PRIMARY(PRM_DESC: REF DBG$PRIMARY, DIRECTION) =
1 !
1 ! FUNCTION
1 !       This routine takes a Primary Descriptor and attempts to find
1 !       the logical successor (DIRECTION=0) or logical predecessor
1 !       (DIRECTION=1). For example, the Primary for X(0) might
1 !       be modified to be a Primary for X(1).
1 !
1 ! INPUTS
1 !       PRM_DESC - points to a Primary Descriptor
1 !       DIRECTION - if 0, we want the logical successor.
1 !                   if 1, we want the logical predecessor.
1 !
1 ! OUTPUTS
1 !       1 is returned if this routine is successful, and
1 !       the Primary pointed to by PRM_DESC is modified.
1 !       If this routine fails, it returns 0 or 2.
1 !
1
2     BEGIN
2
2     MAP
2         PRM_DESC: REF DBG$PRIMARY;        ! Pointer to input Primary Descriptor
2
2     BUILTIN
2         INSQUE,
2         REMQUE;
2
2     LABEL
2         PASS,
2         SCAN;
2
2     LOCAL
2         COMP_FLAG: BYTE,
2         ERROR_STATUS,
2         DUMMY,
2         ROOT_ADR,
2         SUB_NODE: REF DBG$PRIM_NODE,
2         SYM_NAME: REF VECTOR[,BYTE];
2
2
2
2     ! Save away the "current" primary. This is used by our stack
2     ! machine code in RSTACCESS to evaluate the "push inner record
2     ! address" and "push outer record address" instructions.
2     !
2     DBG$GL_CURRENT_PRIMARY = .PRM_DESC;
2
2     ! Give up if the descriptor is not a Primary we can modify
2     ! to get logical successor or predecessor.
2     !
2     IF (.PRM_DESC[DBG$B_DHDR_TYPE] NEQ DBG$K_PRIMARY_DESC) OR
3        ((.PRM_DESC[DBG$B_DHDR_KIND] NEQ RST$K_DATA) AND
2         (.PRM_DESC[DBG$B_DHDR_KIND] NEQ RST$K_TYPCOMP)) OR
3        (.PRM_DESC[DBG$V_DHDR_SUBREF])
2     THEN
2         RETURN 0;
```

```
2
2     ERROR_STATUS = 0;
2
2     WHILE TRUE DO
3 PASS:   BEGIN
3
3
3         ! This loop steps to the last/next component in the data aggregate.
3         ! In most cases this loop is executed exactly once. However to deal
3         ! properly with variant records it is sometimes necessary to repeat
3         ! this entire process until we find a valid component.  Examples of
3         ! invalid components are :
3         !       An anonymous (zero-length) PASCAL Tag Field
3         !       A VARIANT SET with no TAG, or an out-of-range Tag Value
3         !
3         ROOT_ADR = SUB_NODE = PRM_DESC[DBG$A_PRIM_FLINK];
4 SCAN:       BEGIN
4             !+
4             ! Scan the entire primary backwards until we find a group
4             ! item (Array, Record or Variant) which has more components
4             ! left in it.
4             !-
4             WHILE (SUB_NODE = .SUB_NODE[DBG$L_PNODE_BLINK]) NEQ .ROOT_ADR
4              DO IF .SUB_NODE[DBG$V_PNODE_EVAL] THEN
4               SELECTONE .SUB_NODE[DBG$B_PNODE_FCODE] OF
4                 SET
4                 [RST$K_TYPE_ARRAY]:
5                     BEGIN
5                     LOCAL S,S_VECT       : REF DBG$PRIM_NODE_SUBS;
5
5                     ! The following code increments or decrements the
5                     ! subscript vector. This must be incremented like
5                     ! an odometer in a car; that is, normally we
5                     ! just increment the last one, but if that is
5                     ! at its upper limit then set it back to the lower
5                     ! limit and attempt to increment the next one, and
5                     ! so on. We also have to take into account arrays
5                     ! that are stored in column order (first subscript
5                     ! varies fastest).
5                     !
5                     ERROR_STATUS = 2;
5                     S_VECT = SUB_NODE[DBG$A_PNARR_SVECTOR];
5                     INCR DIMENSION FROM 1 TO .SUB_NODE[DBG$B_PNARR_DIMCNT] DO
6                         BEGIN
7                         S = (IF .SUB_NODE[DBG$V_PNARR_COLUMN]
7                                 THEN .DIMENSION - 1
6                                 ELSE .SUB_NODE[DBG$B_PNARR_DIMCNT] - .DIMENSION);
6
6                         IF .DIRECTION EQL 0                 ! 0 = NEXTLOC, 1 = PREVLOC
6                         THEN
6
6                             ! Logical successor.
6                             !
7                             BEGIN
7
7                             ! Check for being at the upper bound.
7                             !
```

```
7                                IF .S_VECT[.S,DBG$L_PNSUB_SVALUE] EQL .S_VECT[.S,DBG$L_PNSUB_UBOUND]
7                                THEN
8                                    BEGIN
8
8                                    ! If we have no more dimensions and we
8                                    ! are at the top subnode (i.e., there are
8                                    ! no "higher" levels at which we can
8                                    ! increment something) then go ahead
8                                    ! and increment it, (but giving a warning
8                                    ! that we are at the upper bound).
8                                    ! For example, if X is a one-dimensional
8                                    ! array from 1 to 3, and we want the logical
8                                    ! successor of X(3), we'll go ahead and return
8                                    ! X(4) but we'll give an informational saying
8                                    ! you have walked past the upper bound.
8                                    ! But if X were 2-dimensional, say 1:3 by 1:3,
8                                    ! and you want the successor of X(1,3),
8                                    ! then return X(2,1) and not X(1,4).
8                                    ! Or if X were a record of arrays,
8                                    ! and X.A(3) was the upper bound, then
8                                    ! you would want to go to the next
8                                    ! record component, say X.B, instead of
8                                    ! going to X.A(4).
8                                    ! That is the reason for the checks for
8                                    ! DIMENSION EQL DIMCNT and BLINK EQL
8                                    ! ROOT_ADR.
8                                    !
8                                    IF (.DIMENSION EQL .SUB_NODE[DBG$B_PNARR_DIMCNT]) AND
9                                       (.SUB_NODE[DBG$L_PNODE_BLINK] EQL .ROOT_ADR)
8                                    THEN
9                                        BEGIN
9                                        SIGNAL(DBG$_SUBSCRNG, 3, UPLIT BYTE(%ASCIC 'upper'), .DIMENSION, .S_VECT
9                                        S_VECT[.S,DBG$L_PNSUB_SVALUE]  =  .S_VECT[.S,DBG$L_PNSUB_SVALUE] + 1;
9                                        LEAVE SCAN;
9                                        END
8                                    ELSE
8
8                                        ! Set back to lower bound.
8                                        !
8                                        S_VECT[.S,DBG$L_PNSUB_SVALUE]  =  .S_VECT[.S,DBG$L_PNSUB_LBOUND]
8                                    END
7                                ELSE
8                                    BEGIN
8
8                                    ! Increment and leave loop.
8                                    !
8                                    S_VECT[.S,DBG$L_PNSUB_SVALUE]  =  .S_VECT[.S,DBG$L_PNSUB_SVALUE] + 1;
8                                    LEAVE SCAN;
7                                    END;
7                                END
6                            ELSE
6
6                                ! Logical predecessor.
6                                !
7                                BEGIN
7                                IF .S_VECT[.S,DBG$L_PNSUB_SVALUE] EQL .S_VECT[.S,DBG$L_PNSUB_LBOUND]
7                                THEN
```

```
8                 BEGIN
8
8                 ! If we have no more dimensions then go ahead
8                 ! and decrement it, (but giving a warning
8                 ! that we are at the lower bound).
8                 ! For example, if X is a one-dimensional
8                 ! array from 1 to 3, and we want the logical
8                 ! predecessor of X(1), we'll go ahead and return
8                 ! X(0) but we'll give an informational saying
8                 ! you have walked past the upper bound.
8                 ! But if X were 2-dimensional, say 1:3 by 1:3,
8                 ! and you want the predecessor of X(3,1),
8                 ! then return X(2,3) and not X(3,0).
8                 ! That is the reason for this check for
8                 ! DIMENSION EQL DIMCNT.
8                 !
8                 IF (.DIMENSION EQL .SUB_NODE[DBG$B_PNARR_DIMCNT]) AND
9                    (.SUB_NODE[DBG$L_PNODE_BLINK] EQL .ROOT_ADR)
8                 THEN
9                     BEGIN
9                     SIGNAL(DBG$_SUBSCRNG, 3, UPLIT BYTE(%ASCIC 'lower'), .DIMENSION, .S_VECT
9                     S_VECT[.S,DBG$L_PNSUB_SVALUE]  =  .S_VECT[.S,DBG$L_PNSUB_SVALUE] - 1;
9                     LEAVE SCAN;
9                     END
8                 ELSE
8
8                     ! Set back to upper bound.
8                     !
8                     S_VECT[.S,DBG$L_PNSUB_SVALUE]  =  .S_VECT[.S,DBG$L_PNSUB_UBOUND]
8                 END
7             ELSE
8                 BEGIN
8
8                 ! Decrement and leave loop.
8                 !
8                 S_VECT[.S,DBG$L_PNSUB_SVALUE]  =  .S_VECT[.S,DBG$L_PNSUB_SVALUE] - 1;
8                 LEAVE SCAN;
7                 END;
6             END;
5         END;
4     END;
4
5 [RST$K_TYPE_RECORD,RST$K_TYPE_VARIANT]:
5     BEGIN
5     ERROR_STATUS = 2;
5     IF .DIRECTION EQL 0          ! 0 = NEXTLOC, 1 = PREVLOC
5     THEN
5
5         ! Logical successor.
5         !
6         BEGIN
6
6         ! If we can go to the next component, do so and exit
6         ! the loop.
6         !
6         IF .SUB_NODE[DBG$W_PNREC_INDEX] LSSU .SUB_NODE[DBG$W_PNREC_NCOMPS]
6         THEN
```

```
                                BEGIN
                                SUB_NODE[DBG$W_PNREC_INDEX] = .SUB_NODE[DBG$W_PNREC_INDEX] + 1;
                                LEAVE SCAN;
                                END;
                            END
                    ELSE

                        ! Logical predecessor.
                        !
                        BEGIN

                        ! If we can get to the previous component, do so
                        ! and exit the loop.
                        !
                        IF .SUB_NODE[DBG$W_PNREC_INDEX] GTRU 1
                        THEN
                            BEGIN
                            SUB_NODE[DBG$W_PNREC_INDEX] = .SUB_NODE[DBG$W_PNREC_INDEX] - 1;
                            LEAVE SCAN;
                            END
                        END;
                    END;
                [OTHERWISE]:0;
                TES;

        ! If we fall through to here without succeeding in incrementing
        ! or decrementing anything, then error status will still be
        ! 0 or 2 and we return it, indicating we did not succeed.
        !
        RETURN .ERROR_STATUS;
        END;                            ! End of block scan

    !+
    ! The following test is a special case so that if the last item examined
    ! was an array element we just examine the next (or previous) element of
    ! the array, even if this is an aggregate (e.g. a record).  In all other
    ! cases we will step down to an individual component of the aggregate.
    !-
    IF (.SUB_NODE[DBG$L_PNODE_FLINK] EQLA .PRM_DESC[DBG$L_PRIM_BLINK])
                                AND
       (.SUB_NODE[DBG$B_PNODE_FCODE] EQL RST$K_TYPE_ARRAY)

      THEN RETURN 1;


    ! We have found the composite entry we are going to modify. Strip off
    ! all subsequent primary sub-nodes, and then add new sub_nodes to get
    ! a primary which describes a single data item.
    !
    WHILE .SUB_NODE[DBG$L_PNODE_FLINK] NEQA .ROOT_ADR DO
        REMQUE(.SUB_NODE[DBG$L_PNODE_FLINK],DUMMY);

    IF .PRM_DESC[DBG$V_DHDR_TMPREF] THEN
        BEGIN
        PRM_DESC[DBG$V_DHDR_TMPREF] = FALSE;
        PRM_DESC[DBG$V_DHDR_SUBREF] = FALSE;
        PRM_DESC[DBG$W_PRIM_OFFSET] = 0;
```

```
                PRM_DESC[DBG$W_PRIM_LENGTH] = 0;
                END;

        COMP_FLAG = FALSE;

        WHILE TRUE DO
                BEGIN
                LOCAL SYMID,TYPEID,FCODE,KIND;
                SELECTONE .SUB_NODE[DBG$B_PNODE_FCODE] OF
                    SET
                    [RST$K_TYPE_ARRAY]:
                        BEGIN
                        IF .COMP_FLAG THEN
                            BEGIN
                            LOCAL S_VECT : REF DBG$PRIM_NODE_SUBS;
                            S_VECT = SUB_NODE[DBG$A_PNARR_SVECTOR];
                            DECR S FROM .SUB_NODE[DBG$B_PNARR_DIMCNT]-1 TO 0 DO
                                S_VECT[.S,DBG$L_PNSUB_SVALUE] =  .S_VECT[.S,DBG$L_PNSUB_UBOUND];
                            END;
                        SYMID = 0;
                        KIND = RST$K_DATA;
                        DBG$STA_SYMTYPE(.SUB_NODE[DBG$L_PNARR_CELLTYPE],FCODE,TYPEID);
                        END;

                    [RST$K_TYPE_RECORD,RST$K_TYPE_VARIANT]:
                        BEGIN
                        LOCAL N_COMPS,S_VECT : REF VECTOR[,LONG];
                        IF .SUB_NODE[DBG$B_PNODE_FCODE] EQL RST$K_TYPE_RECORD
                          THEN DBG$STA_TYP_RECORD(.SUB_NODE[DBG$L_PNODE_TYPEID],N_COMPS,S_VECT,DUMMY)
                          ELSE
                            BEGIN
                            !+
                            ! We have a variant set.  If the TYPEID field is non-zero this is
                            ! a new variant sub node (added by us on the last pass), and so we
                            ! need to obtain the current value of the TAG field. A zero-length
                            ! tag field (or an illegal tag value) cause this entire variant to
                            ! be ignored.
                            !-
                            IF (TYPEID = .SUB_NODE[DBG$L_PNODE_TYPEID]) NEQ 0 THEN
                                BEGIN
                                LOCAL TAG,MARK,TYPE,CODE,
                                      VAL_DESC  : REF DBG$VALDESC,
                                      VARIANT   : REF RST$VAR_ENTRY;
                                MAP   TYPEID    : REF RST$ENTRY;

                                REMQUE(.SUB_NODE,SUB_NODE);
                                IF (TAG  = .TYPEID[RST$L_VARTAGPTR]) EQL 0 THEN LEAVE PASS;
                                DBG$STA_SYMNAME(.TAG,SYM_NAME);
                                IF .SYM_NAME[0] EQL 0 THEN LEAVE PASS;
                                !+
                                ! We now need to obtain the actual tag value. This is done
                                ! by stripping off the VARIANT sub-node,  and adding a new
                                ! primary sub-node describing the Tag field.  We then call
                                ! dbg$prim_to_val to get a Debug Value descriptor, extract
                                ! the tag value, and restore the state of the primary.
                                !-
                                MARK = DBG$PUSH_TEMPMEM();
```

```
7                               DBG$STA_SYMTYPE(.TAG,CODE,TYPE);
7                               DBG$BUILD_PRIMARY_SUBNODE(.PRM_DESC,RST$K_DATA,.TAG,.CODE,.TYPE,0);
7                               DBG$PRIM_TO_VAL(.PRM_DESC,DBG$K_VALUE_DESC,VAL_DESC);
7                               TAG = .VAL_DESC[DBG$L_VALUE_VALUE0];
7                               REMQUE(.PRM_DESC[DBG$L_PRIM_BLINK],DUMMY);
7                               DBG$POP_TEMPMEM(.MARK);
7                               IF (VARIANT = DBG$STA_VARIANT_SELECT(.TAG,.TYPEID)) EQL 0 THEN LEAVE PASS;
7                               SUB_NODE[DBG$V_PNVAR_VALID]    = TRUE;
7                               SUB_NODE[DBG$W_PNVAR_INDEX]    = 1;
7                               SUB_NODE[DBG$L_PNODE_TYPEID]   = 0;
7                               SUB_NODE[DBG$L_PNVAR_TAGID]    =  .TYPEID[RST$L_VARTAGPTR];
7                               SUB_NODE[DBG$W_PNVAR_NCOMPS]   = .VARIANT[RST$L_VAR_COMPCNT];
7                               SUB_NODE[DBG$L_PNVAR_COMPLST] =  VARIANT[RST$A_VAR_COMPLST];
7                               SUB_NODE[DBG$L_PNVAR_DSTPTR]   = .VARIANT[RST$L_VAR_DSTPTR];
7                               INSQUE(.SUB_NODE,.PRM_DESC[DBG$L_PRIM_BLINK]);
6                               END;
6                           N_COMPS = .SUB_NODE[DBG$W_PNVAR_NCOMPS];
6                           S_VECT  = .SUB_NODE[DBG$L_PNVAR_COMPLST];
5                           END;
5                       IF .COMP_FLAG THEN SUB_NODE[DBG$W_PNREC_INDEX] = .N_COMPS;
5                       SYMID = .S_VECT[.SUB_NODE[DBG$W_PNREC_INDEX]-1];
5                       DBG$STA_SYMKIND(.SYMID,KIND);
5                       IF .KIND EQL RST$K_VARIANT
5                         THEN
6                           BEGIN
6                           FCODE  = RST$K_TYPE_VARIANT;
6                           TYPEID = .SYMID;
6                           SYMID  = 0;
6                           END
5                         ELSE
4                           DBG$STA_SYMTYPE(.SYMID,FCODE,TYPEID);
4                       END;
4
4                   [OTHERWISE]:
4                       EXITLOOP;
4                   TES;
4               SUB_NODE[DBG$V_PNODE_EVAL] = TRUE;
4               DBG$BUILD_PRIMARY_SUBNODE(.PRM_DESC,.KIND,.SYMID,.FCODE,.TYPEID, 0);
4               SUB_NODE = .PRM_DESC[DBG$L_PRIM_BLINK];
3               COMP_FLAG = .DIRECTION;
3               END;
3
3           IF .SUB_NODE[DBG$L_PNODE_SYMID] EQL 0 THEN EXITLOOP;
3           DBG$STA_SYMNAME(.SUB_NODE[DBG$L_PNODE_SYMID],SYM_NAME);
2           IF .SYM_NAME[0] NEQ 0 THEN EXITLOOP;
2           END;
2
1       RETURN 1;
        END;                            ! End of modify_primary


                                                        .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

                72  65  70  70  75  05  0003E P.AAJ:    .ASCII  <5>\upper\                             ;
                72  65  77  6F  6C  05  00044 P.AAK:    .ASCII  <5>\lower\                             ;
```

```
                                                  .PSECT   DBG$CODE,NOWRT,  SHR,  PIC,0

                               0FFC 00000 MODIFY_PRIMARY:
                                                  .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11         ; 1466
              5B 00000000G   00  9E 00002         MOVAB    LIB$SIGNAL, R11                              ;
              5E             28  C2 00009         SUBL2    #40, SP                                      ;
              50        04   AC  D0 0000C         MOVL     PRM_DESC, R0                                 ; 1512
   00000000G  00             50  D0 00010         MOVL     R0, DBG$GL_CURRENT_PRIMARY                   ;
          79  8F        02   A0  91 00017         CMPB     2(R0), #12T                                  ; 1517
                             11  12 0001C         BNEQ     2$                                           ;
              06        07   A0  91 0001E         CMPB     7(R0), #6                                    ; 1518
                             06  13 00022         BEQL     1$                                           ;
              0A        07   A0  91 00024         CMPB     7(R0), #10                                   ; 1519
                             05  12 00028         BNEQ     2$                                           ;
03        04  A0             01  E1 0002A 1$:     BBC      #1, 4(R0), 3$                                ; 1520
                           02DC  31 0002F 2$:     BRW      45$                                          ;
                             59  D4 00032 3$:     CLRL     ERROR_STATUS                                 ; 1524
53        04  AC             14  C1 00034 4$:     ADDL3    #20, PRM_DESC, SUB_NODE                      ; 1538
              58             53  D0 00039         MOVL     SUB_NODE, ROOT_ADR                           ;
              53        04   A3  D0 0003C 5$:     MOVL     4(SUB_NODE), SUB_NODE                        ; 1545
              58             53  D1 00040         CMPL     SUB_NODE, ROOT_ADR                           ;
                             03  12 00043         BNEQ     6$                                           ;
                           00E3  31 00045         BRW      22$                                          ;
              F0        0A   A3  E9 00048 6$:     BLBC     10(SUB_NODE), 5$                             ; 1546
              50        09   A3  9A 0004C         MOVZBL   9(SUB_NODE), R0                              ; 1547
              01             50  91 00050         CMPB     R0, #T                                       ; 1549
                             03  13 00053         BEQL     7$                                           ;
                           00AA  31 00055         BRW      19$                                          ;
              59             02  D0 00058 7$:     MOVL     #2, ERROR_STATUS                             ; 1563
              52        28   A3  9E 0005B         MOVAB    40(R3), S_VECT                               ; 1564
              57        1B   A3  9A 0005F         MOVZBL   27(SUB_NODE), R7                             ; 1565
                             54  D4 00063         CLRL     DIMENSION                                    ; 1580
                           0091  31 00065         BRW      17$                                          ;
06        0A  A3             01  E1 00068 8$:     BBC      #1, 10(SUB_NODE), 9$                         ; 1567
              55        FF   A4  9E 0006D         MOVAB    -1(R4), S                                    ; 1568
                             04  11 00071         BRB      10$                                          ;
55            57             54  C3 00073 9$:     SUBL3    DIMENSION, R7, S                             ; 1569
50            55             14  C5 00077 10$:    MULL3    #20, S, R0                                   ; 1580
56            52             50  C1 0007B         ADDL3    R0, S_VECT, R6                               ;
                        08   AC  D5 0007F         TSTL     DIRECTION                                    ; 1571
                             39  12 00082         BNEQ     13$                                          ;
                        0C A042  9F 00084         PUSHAB   12(R0)[S_VECT]                               ; 1580
              9E             66  D1 00088         CMPL     (R6), @(SP)+                                 ;
                             2C  12 0008B         BNEQ     12$                                          ;
              57             54  D1 0008D         CMPL     DIMENSION, R7                                ; 1607
                             21  12 00090         BNEQ     11$                                          ;
              58        04   A3  D1 00092         CMPL     4(SUB_NODE), ROOT_ADR                        ; 1608
                             1B  12 00096         BNEQ     11$                                          ;
                        0C A042  9F 00098         PUSHAB   12(R0)[S_VECT]                               ; 1611
                             9E  DD 0009C         PUSHL    @(SP)+                                       ;
                             54  DD 0009E         PUSHL    DIMENSION                                    ;
                 00000000'   EF  9F 000A0         PUSHAB   P.AAJ                                        ;
                             03  DD 000A6         PUSHL    #3                                           ;
                  000287AB   8F  DD 000A8         PUSHL    #165803                                      ;
              6B             05  FB 000AE         CALLS    #5, LIB$SIGNAL                               ;
```

```
                                     06  11 000B1        BRB     12$                              ; 1612
                          08 A042    9F  000B3 11$:      PUSHAB  8(R0)[S_VECT]                    ; 1619
                                37   11  000B7           BRB     15$                              ;
                                66   D6  000B9 12$:      INCL    (R6)                             ; 1626
                                72   11  000BB           BRB     23$                              ; 1627
                          08 A042    9F  000BD 13$:      PUSHAB  8(R0)[S_VECT]                    ; 1635
                   9E           66   D1  000C1           CMPL    (R6), @(SP)+                     ;
                                2F   12  000C4           BNEQ    16$                              ;
                   57           54   D1  000C6           CMPL    DIMENSION, R7                    ; 1653
                                21   12  000C9           BNEQ    14$                              ;
                   58     04    A3   D1  000CB           CMPL    4(SUB_NODE), ROOT_ADR            ; 1654
                                1B   12  000CF           BNEQ    14$                              ;
                          0C A042    9F  000D1           PUSHAB  12(R0)[S_VECT]                   ; 1657
                                9E   DD  000D5           PUSHL   @(SP)+                           ;
                                54   DD  000D7           PUSHL   DIMENSION                        ;
                     00000000'  EF   9F  000D9           PUSHAB  P.AAK                            ;
                                03   DD  000DF           PUSHL   #3                               ;
                     000287AB   8F   DD  000E1           PUSHL   #165803                          ;
                   6B           05   FB  000E7           CALLS   #5, LIB$SIGNAL                   ;
                                09   11  000EA           BRB     16$                              ; 1658
                          0C A042    9F  000EC 14$:      PUSHAB  12(R0)[S_VECT]                   ; 1665
                   66           9E   D0  000F0 15$:      MOVL    @(SP)+, (R6)                     ;
                                04   11  000F3           BRB     17$                              ; 1637
                                66   D7  000F5 16$:      DECL    (R6)                             ; 1672
                                36   11  000F7           BRB     23$                              ; 1673
FF69         54    01           57   F1  000F9 17$:      ACBL    R7, #1, DIMENSION, 8$            ; 1565
                              FF3A   31  000FF 18$:      BRW     5$                               ; 1547
                   07           50   91  00102 19$:      CMPB    R0, #7                           ; 1679
                                05   13  00105           BEQL    20$                              ;
                   13           50   91  00107           CMPB    R0, #19                          ;
                                F3   12  0010A           BNEQ    18$                              ;
                   59           02   D0  0010C 20$:      MOVL    #2, ERROR_STATUS                 ; 1681
                   50     18    A3   9E  0010F           MOVAB   24(SUB_NODE), R0                 ; 1692
                          08    AC   D5  00113           TSTL    DIRECTION                        ; 1682
                                0A   12  00116           BNEQ    21$                              ;
              02   A0           60   B1  00118           CMPW    (R0), 2(R0)                      ; 1692
                                E1   1E  0011C           BGEQU   18$                              ;
                                60   B6  0011E           INCW    (R0)                             ; 1695
                                0D   11  00120           BRB     23$                              ; 1696
                   01           60   B1  00122 21$:      CMPW    (R0), #1                         ; 1708
                                D8   1B  00125           BLEQU   18$                              ;
                                60   B7  00127           DECW    (R0)                             ; 1711
                                04   11  00129           BRB     23$                              ; 1712
                   50           59   D0  0012B 22$:      MOVL    ERROR_STATUS, R0                 ; 1723
                                     04  0012E           RET                                      ;
                   50     04    AC   D0  0012F 23$:      MOVL    PRM_DESC, R0                     ; 1732
              18   A0           63   D1  00133           CMPL    (SUB_NODE), 24(R0)               ;
                                09   12  00137           BNEQ    24$                              ;
                   01     09    A3   91  00139           CMPB    9(SUB_NODE), #1                  ; 1734
                                03   12  0013D           BNEQ    24$                              ;
                              01C8   31  0013F           BRW     44$                              ;
                   58           63   D1  00142 24$:      CMPL    (SUB_NODE), ROOT_ADR             ; 1743
                                06   13  00145           BEQL    25$                              ;
                   6E     00    B3   0F  00147           REMQUE  @0(SUB_NODE), DUMMY              ; 1744
                                F5   11  0014B           BRB     24$                              ;
                   50     04    AC   D0  0014D 25$:      MOVL    PRM_DESC, R0                     ; 1746
                   09     05    A0   E9  00151           BLBC    5(R0), 26$                       ;
```

```
             04   A0     0102   8F   AA 00155           BICW2   #258, 4(R0)                    ; 1748
                           10   A0   D4 0015B           CLRL    16(R0)                         ; 1750
                                5A   94 0015E 26$:      CLRB    COMP_FLAG                      ; 1754
                  50       09   A3   9A 00160 27$:      MOVZBL  9(SUB_NODE), R0                ; 1759
                  01            50   91 00164           CMPB    R0, #1                         ; 1761
                                30   12 00167           BNEQ    31$                            ;
                  1B            5A   E9 00169           BLBC    COMP_FLAG, 30$                 ; 1763
                  52       28   A3   9E 0016C           MOVAB   40(R3), S_VECT                 ; 1766
                  50       1B   A3   9A 00170           MOVZBL  27(SUB_NODE), S                ; 1768
                                0E   11 00174           BRB     29$                            ;
     51           50            14   C5 00176 28$:      MULL3   #20, S, R1                     ;
                              6142   9F 0017A           PUSHAB  (R1)[S_VECT]                   ;
                           0C A142   9F 0017D           PUSHAB  12(R1)[S_VECT]                 ;
                  9E            9E   D0 00181           MOVL    @(SP)+, @(SP)+                 ;
                  EF            50   F4 00184 29$:      SOBGEQ  S, 28$                         ;
                                55   D4 00187 30$:      CLRL    SYMID                          ; 1770
             18   AE            06   D0 00189           MOVL    #6, KIND                       ; 1771
                           1C   AE   9F 0018D           PUSHAB  TYPEID                         ; 1772
                           24   AE   9F 00190           PUSHAB  FCODE                          ;
                           24   A3   DD 00193           PUSHL   36(SUB_NODE)                   ;
                              0127   31 00196           BRW     41$                            ;
                  07            50   91 00199 31$:      CMPB    R0, #7                         ; 1775
                                08   13 0019C           BEQL    32$                            ;
                  13            50   91 0019E           CMPB    R0, #19                        ;
                                03   13 001A1           BEQL    32$                            ;
                              014A   31 001A3           BRW     43$                            ;
                  07            50   91 001A6 32$:      CMPB    R0, #7                         ; 1778
                                15   12 001A9           BNEQ    33$                            ;
                                5E   DD 001AB           PUSHL   SP                             ; 1779
                           08   AE   9F 001AD           PUSHAB  S_VECT                         ;
                           10   AE   9F 001B0           PUSHAB  N_COMPS                        ;
                           0C   A3   DD 001B3           PUSHL   12(SUB_NODE)                   ;
      00000000G   00            04   FB 001B6           CALLS   #4, DBG$STA_TYP_RECORD         ;
                              00C5   31 001BD           BRW     38$                            ;
             1C   AE       0C   A3   D0 001C0 33$:      MOVL    12(SUB_NODE), TYPEID           ; 1789
                                03   12 001C5           BNEQ    34$                            ;
                              00B1   31 001C7           BRW     37$                            ;
                  53            63   0F 001CA 34$:      REMQUE  (SUB_NODE), SUB_NODE           ; 1796
                  50       1C   AE   D0 001CD           MOVL    TYPEID, R0                     ; 1797
                  54       10   A0   D0 001D1           MOVL    16(R0), TAG                    ;
                                79   13 001D5           BEQL    35$                            ;
                           24   AE   9F 001D7           PUSHAB  SYM_NAME                       ; 1798
                                54   DD 001DA           PUSHL   TAG                            ;
      00000000G   00            02   FB 001DC           CALLS   #2, DBG$STA_SYMNAME            ;
                           24   BE   95 001E3           TSTB    @SYM_NAME                      ; 1799
                                68   13 001E6           BEQL    35$                            ;
      00000000G   00            00   FB 001E8           CALLS   #0, DBG$PUSH_TEMPMEM           ; 1807
                  57            50   D0 001EF           MOVL    R0, MARK                       ;
                           0C   AE   9F 001F2           PUSHAB  TYPE                           ; 1808
                           14   AE   9F 001F5           PUSHAB  CODE                           ;
                                54   DD 001F8           PUSHL   TAG                            ;
      00000000G   00            03   FB 001FA           CALLS   #3, DBG$STA_SYMTYPE            ;
                                7E   D4 00201           CLRL    -(SP)                          ; 1809
                           10   AE   DD 00203           PUSHL   TYPE                           ;
                           18   AE   DD 00206           PUSHL   CODE                           ;
                                54   DD 00209           PUSHL   TAG                            ;
                                06   DD 0020B           PUSHL   #6                             ;
```

```
              52     04   AC   D0 0020D         MOVL    PRM_DESC, R2                                  ;
                          52   DD 00211         PUSHL   R2                                            ;
       00000000G  00      06   FB 00213         CALLS   #6, DBG$BUILD_PRIMARY_SUBNODE                 ;
                     14   AE   9F 0021A         PUSHAB  VAL_DESC                                      ; 1810
              7E     7A   8F   9A 0021D         MOVZBL  #122, -(SP)                                   ;
                          52   DD 00221         PUSHL   R2                                            ;
       00000000G  00      03   FB 00223         CALLS   #3, DBG$PRIM_TO_VAL                           ;
              50     14   AE   D0 0022A         MOVL    VAL_DESC, R0                                  ; 1811
              54     20   A0   D0 0022E         MOVL    32(R0), TAG                                   ;
              6E     18   B2   0F 00232         REMQUE  @24(R2), DUMMY                                ; 1812
                          57   DD 00236         PUSHL   MARK                                          ; 1813
       00000000G  00      01   FB 00238         CALLS   #1, DBG$POP_TEMPMEM                           ;
              52     1C   AE   D0 0023F         MOVL    TYPEID, R2                                    ; 1814
                          52   DD 00243         PUSHL   R2                                            ;
                          54   DD 00245         PUSHL   TAG                                           ;
       00000000G  00      02   FB 00247         CALLS   #2, DBG$STA_VARIANT_SELECT                    ;
                          50   D5 0024E         TSTL    VARIANT                                       ;
                          03   12 00250  35$:   BNEQ    36$                                           ;
                        FDDF   31 00252         BRW     4$                                            ;
          0A  A3          10   88 00255  36$:   BISB2   #16, 10(SUB_NODE)                             ; 1815
          18  A3          01   B0 00259         MOVW    #1, 24(SUB_NODE)                              ; 1816
                     0C   A3   D4 0025D         CLRL    12(SUB_NODE)                                  ; 1817
          1C  A3     10   A2   D0 00260         MOVL    16(R2), 28(SUB_NODE)                          ; 1818
          1A  A3     04   A0   B0 00265         MOVW    4(VARIANT), 26(SUB_NODE)                      ; 1819
          20  A3     08   A0   9E 0026A         MOVAB   8(R0), 32(SUB_NODE)                           ; 1820
          24  A3          60   D0 0026F         MOVL    (VARIANT), 36(SUB_NODE)                       ; 1821
              50     04   AC   D0 00273         MOVL    PRM_DESC, R0                                  ; 1822
          18  B0          63   0E 00277         INSQUE  (SUB_NODE), @24(R0)                           ;
          08  AE     1A   A3   3C 0027B  37$:   MOVZWL  26(SUB_NODE), N_COMPS                         ; 1824
          04  AE     20   A3   D0 00280         MOVL    32(SUB_NODE), S_VECT                          ; 1825
              05          5A   E9 00285  38$:   BLBC    COMP_FLAG, 39$                                ; 1827
          18  A3     08   AE   B0 00288         MOVW    N_COMPS, 24(SUB_NODE)                         ;
              50     18   A3   3C 0028D  39$:   MOVZWL  24(SUB_NODE), R0                              ; 1828
              50     04 BE40   DE 00291         MOVAL   @S_VECT[R0], R0                               ;
              55     FC   A0   D0 00296         MOVL    -4(R0), SYMID                                 ;
                     18   AE   9F 0029A         PUSHAB  KIND                                          ; 1829
                          55   DD 0029D         PUSHL   SYMID                                         ;
       00000000G  00      02   FB 0029F         CALLS   #2, DBG$STA_SYMKIND                           ;
              0B     18   AE   D1 002A6         CMPL    KIND, #11                                     ; 1830
                          0C   12 002AA         BNEQ    40$                                           ;
          20  AE          13   D0 002AC         MOVL    #19, FCODE                                    ; 1833
          1C  AE          55   D0 002B0         MOVL    SYMID, TYPEID                                 ; 1834
                          55   D4 002B4         CLRL    SYMID                                         ; 1835
                          0F   11 002B6         BRB     42$                                           ; 1830
                     1C   AE   9F 002B8  40$:   PUSHAB  TYPEID                                        ; 1838
                     24   AE   9F 002BB         PUSHAB  FCODE                                         ;
                          55   DD 002BE         PUSHL   SYMID                                         ;
       00000000G  00      03   FB 002C0  41$:   CALLS   #3, DBG$STA_SYMTYPE                           ;
          0A  A3          01   88 002C7  42$:   BISB2   #1, 10(SUB_NODE)                              ; 1844
                          7E   D4 002CB         CLRL    -(SP)                                         ; 1845
                     20   AE   DD 002CD         PUSHL   TYPEID                                        ;
                     28   AE   DD 002D0         PUSHL   FCODE                                         ;
                          55   DD 002D3         PUSHL   SYMID                                         ;
                     28   AE   DD 002D5         PUSHL   KIND                                          ;
              52     04   AC   D0 002D8         MOVL    PRM_DESC, R2                                  ;
                          52   DD 002DC         PUSHL   R2                                            ;
       00000000G  00      06   FB 002DE         CALLS   #6, DBG$BUILD_PRIMARY_SUBNODE                 ;
```

```
                  53      18    A2  D0 002E5         MOVL    24(R2), SUB_NODE                 ; 1846
                  5A      08    AC  90 002E9         MOVB    DIRECTION, COMP_FLAG             ; 1847
                              FE70  31 002ED         BRW     27$                              ; 1756
                          10    A3  D5 002F0 43$:    TSTL    16(SUB_NODE)                     ; 1850
                                15  13 002F3         BEQL    44$                              ;
                          24    AE  9F 002F5         PUSHAB  SYM_NAME                         ; 1851
                          10    A3  DD 002F8         PUSHL   16(SUB_NODE)                     ;
        00000000G  00           02  FB 002FB         CALLS   #2, DBG$STA_SYMNAME              ;
                          24    BE  95 00302         TSTB    @SYM_NAME                        ; 1852
                                03  12 00305         BNEQ    44$                              ;
                              FD2A  31 00307         BRW     4$                               ;
                  50            01  D0 0030A 44$:    MOVL    #1, R0                           ; 1855
                                    04 0030D         RET                                      ;
                                50  D4 0030E 45$:    CLRL    R0                               ; 1856
                                    04 00310         RET                                      ;

; Routine Size:  785 bytes,    Routine Base:  DBG$CODE + 0C10
```

```
1 ROUTINE PRIMARY_ORDER(PRIM_1: REF DBG$PRIMARY, PRIM_2: REF DBG$PRIMARY) =
1 !
1 ! FUNCTION
1 !      ---------------------------
1 !
1 ! INPUTS
1 !      ---------------------------
1 !
1 ! OUTPUTS
1 !      ---------------------------
1 !
1
2     BEGIN
2     LOCAL
2         NODE_1  : REF DBG$PRIM_NODE,
2         NODE_2  : REF DBG$PRIM_NODE,
2         VALUE_1,
2         VALUE_2;
2
2     NODE_1 = .PRIM_1[DBG$L_PRIM_FLINK];
2     NODE_2 = .PRIM_2[DBG$L_PRIM_FLINK];
2
2     WHILE TRUE DO
3         BEGIN
3
3         SELECTONE .NODE_1[DBG$B_PNODE_FCODE] OF
3             SET
3             [RST$K_TYPE_RECORD,RST$K_TYPE_VARIANT]:
4                 BEGIN
4                 IF .NODE_1[DBG$B_PNODE_FCODE] EQL RST$K_TYPE_VARIANT THEN
4                   IF .NODE_1[DBG$L_PNVAR_DSTPTR] NEQ .NODE_2[DBG$L_PNVAR_DSTPTR]
4                     THEN SIGNAL(DBG$_EXARANGE);
4                 VALUE_1 = .NODE_1[DBG$W_PNREC_INDEX];
4                 VALUE_2 = .NODE_2[DBG$W_PNREC_INDEX];
4                 IF .VALUE_1 LSS .VALUE_2 THEN RETURN -1;
4                 IF .VALUE_1 GTR .VALUE_2 THEN RETURN +1;
3                 END;
3
3             [RST$K_TYPE_ARRAY]:
4                 BEGIN
4                 LOCAL
4                     SUBS_1       : REF DBG$PRIM_NODE_SUBS,
4                     SUBS_2       : REF DBG$PRIM_NODE_SUBS;
4
4                 IF .NODE_1[DBG$B_PNARR_SUBCNT] NEQ .NODE_2[DBG$B_PNARR_SUBCNT]
4                   THEN SIGNAL(DBG$_EXARANGE);
4
4                 SUBS_1 = NODE_1[DBG$A_PNARR_SVECTOR];
4                 SUBS_2 = NODE_2[DBG$A_PNARR_SVECTOR];
4
4                 INCR DIMENSION FROM 0 TO .NODE_1[DBG$B_PNARR_DIMCNT]-1 DO
5                     BEGIN
5                     LOCAL D;
6                     D = (IF .NODE_1[DBG$V_PNARR_COLUMN]
6                             THEN .NODE_1[DBG$B_PNARR_DIMCNT] - .DIMENSION - 1
5                             ELSE .DIMENSION);
5
```

```
; 1792          1914  5                    VALUE_1 = .SUBS_1[.D,DBG$L_PNSUB_SVALUE] - .SUBS_1[.D,DBG$L_PNSUB_LBOUND];
; 1793          1915  5                    VALUE_2 = .SUBS_2[.D,DBG$L_PNSUB_SVALUE] - .SUBS_2[.D,DBG$L_PNSUB_LBOUND];
; 1794          1916  5                    IF .VALUE_1 LSS .VALUE_2 THEN RETURN -1;
; 1795          1917  5                    IF .VALUE_1 GTR .VALUE_2 THEN RETURN +1;
; 1796          1918  4                    END;
; 1797          1919  3                END;
; 1798          1920  3
; 1799          1921  3            [OTHERWISE]:
; 1800          1922  3                EXITLOOP;
; 1801          1923  3            TES;
; 1802          1924  3
; 1803          1925  3        NODE_1 = .NODE_1[DBG$L_PNODE_FLINK];
; 1804          1926  3        NODE_2 = .NODE_2[DBG$L_PNODE_FLINK];
; 1805          1927  2        END;
; 1806          1928  2    RETURN 0;
; 1807          1929  1    END;                        ! End of primary_order



                                                03FC 00000 PRIMARY_ORDER:
                                                                   .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9              ; 1857
                         59 00000000G   00  9E 00002               MOVAB    LIB$SIGNAL, R9                            ;
                         50        04   AC  D0 00009               MOVL     PRIM_1, R0                                ; 1876
                         52        14   A0  D0 0000D               MOVL     20(R0), NODE_1                            ;
                         50        08   AC  D0 00011               MOVL     PRIM_2, R0                                ; 1877
                         53        14   A0  D0 00015               MOVL     20(R0), NODE_2                            ;
                         50        09   A2  9A 00019 1$:           MOVZBL   9(NODE_1), R0                             ; 1882
                         07             50  91 0001D               CMPB     R0, #7                                    ; 1884
                                        05  13 00020               BEQL     2$                                        ;
                         13             50  91 00022               CMPB     R0, #19                                   ;
                                        26  12 00025               BNEQ     4$                                        ;
                         13             50  91 00027 2$:           CMPB     R0, #19                                   ; 1886
                                        10  12 0002A               BNEQ     3$                                        ;
                 24      A3        24   A2  D1 0002C               CMPL     36(NODE_1), 36(NODE_2)                    ; 1887
                                        09  13 00031               BEQL     3$                                        ;
                            00028190    8F  DD 00033               PUSHL    #164240                                   ; 1888
                         69             01  FB 00039               CALLS    #1, LIB$SIGNAL                            ;
                         57        18   A2  3C 0003C 3$:           MOVZWL   24(NODE_1), VALUE_1                       ; 1889
                         56        18   A3  3C 00040               MOVZWL   24(NODE_2), VALUE_2                       ; 1890
                         56             57  D1 00044               CMPL     VALUE_1, VALUE_2                          ; 1891
                                        5B  19 00047               BLSS     9$                                        ;
                                        67  15 00049               BLEQ     13$                                       ; 1892
                                        5D  11 0004B               BRB      11$                                       ;
                         01             50  91 0004D 4$:           CMPB     R0, #1                                    ; 1895
                                        69  12 00050               BNEQ     14$                                       ;
                 1F      A3        1F   A2  91 00052               CMPB     31(NODE_1), 31(NODE_2)                    ; 1901
                                        09  13 00057               BEQL     5$                                        ;
                            00028190    8F  DD 00059               PUSHL    #164240                                   ; 1902
                         69             01  FB 0005F               CALLS    #1, LIB$SIGNAL                            ;
                         51        28   A2  9E 00062 5$:           MOVAB    40(R2), SUBS_1                            ; 1904
                         50        28   A3  9E 00066               MOVAB    40(R3), SUBS_2                            ; 1905
                         58        1B   A2  9A 0006A               MOVZBL   27(NODE_1), R8                            ; 1907
                         55             01  CE 0006E               MNEGL    #1, DIMENSION                             ; 1915
                                        3B  11 00071               BRB      12$                                       ;
          0B      0A     A2             01  E1 00073 6$:           BBC      #1, 10(NODE_1), 7$                        ; 1910
```

```
                  54      1B    A2  9A 00078        MOVZBL  27(NODE_1), R4                     ; 1911
                  54            55  C2 0007C        SUBL2   DIMENSION, R4                      ;
                                54  D7 0007F        DECL    D                                  ;
                                03  11 00081        BRB     8$                                 ;
                  54            55  D0 00083 7$:    MOVL    DIMENSION, D                       ; 1912
                  54            14  C4 00086 8$:    MULL2   #20, R4                            ; 1914
                              6441  9F 00089        PUSHAB  (R4)[SUBS_1]                       ;
                           08 A441  9F 0008C        PUSHAB  8(R4)[SUBS_1]                      ;
        57                 9E   9E  C3 00090        SUBL3   @(SP)+, @(SP)+, VALUE_1            ;
                              6440  9F 00094        PUSHAB  (R4)[SUBS_2]                       ; 1915
                           08 A440  9F 00097        PUSHAB  8(R4)[SUBS_2]                      ;
        56                 9E   9E  C3 0009B        SUBL3   @(SP)+, @(SP)+, VALUE_2            ;
                  56            57  D1 0009F        CMPL    VALUE_1, VALUE_2                   ; 1916
                                04  18 000A2        BGEQ    10$                                ;
        50                      01  CE 000A4 9$:    MNEGL   #1, R0                             ;
                                    04 000A7        RET                                        ;
                                04  15 000A8 10$:   BLEQ    12$                                ; 1917
        50                      01  D0 000AA 11$:   MOVL    #1, R0                             ;
                                    04 000AD        RET                                        ;
        C1        55            58  F2 000AE 12$:   AOBLSS  R8, DIMENSION, 6$                  ; 1907
                  52            62  D0 000B2 13$:   MOVL    (NODE_1), NODE_1                   ; 1925
                  53            63  D0 000B5        MOVL    (NODE_2), NODE_2                   ; 1926
                              FF5E  31 000B8        BRW     1$                                 ; 1879
                                50  D4 000BB 14$:   CLRL    R0                                 ; 1928
                                    04 000BD        RET                                        ; 1929

; Routine Size:  190 bytes,    Routine Base:  DBG$CODE + 0F21
```

```
ROUTINE CHECK_TEXT_DESCRIPTOR(VAL_DESC: REF DBG$VALDESC) =
!
! FUNCTION
!       ----------------------------
!
! INPUTS
!       ----------------------------
!
! OUTPUTS
!       ----------------------------
!
!

    BEGIN

    BUILTIN
        PROBER;

    BIND VMS_DESC = VAL_DESC[DBG$A_VALUE_VMSDESC]: DBG$STG_DESC;



    IF (.VAL_DESC[DBG$B_VALUE_CLASS] EQL DSC$K_CLASS_UBS) THEN SIGNAL(DBG$_UNALIGNED);

    IF NOT PROBER(%REF(0),%REF(8) ,.VAL_DESC[DBG$L_VALUE_POINTER])
      THEN SIGNAL(DBG$_NOACCESSR,1,.VAL_DESC[DBG$L_VALUE_POINTER]);

    CH$MOVE(8,.VAL_DESC[DBG$L_VALUE_POINTER],VAL_DESC[DBG$A_VALUE_VMSDESC]);

    IF (.VMS_DESC[DSC$B_DTYPE] EQL 0) AND (.VMS_DESC[DSC$B_CLASS] EQL 0)
    THEN
        BEGIN
        VMS_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
        VMS_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
        RETURN TRUE;
        END;

    IF  (.VMS_DESC[DSC$B_CLASS] EQL DSC$K_CLASS_D)
     THEN VMS_DESC[DSC$B_CLASS]  =  DSC$K_CLASS_S;

    IF  (.VMS_DESC[DSC$B_CLASS] EQL DSC$K_CLASS_VS)
    AND (.VMS_DESC[DSC$B_DTYPE] EQL DSC$K_DTYPE_T)
     THEN VMS_DESC[DSC$B_DTYPE]  =  DSC$K_DTYPE_VT;

    IF  (.VMS_DESC[DSC$B_CLASS] NEQ DSC$K_CLASS_VS)
    AND (.VMS_DESC[DSC$B_CLASS] NEQ DSC$K_CLASS_S)  THEN RETURN FALSE;

    SELECTONE .VMS_DESC[DSC$B_DTYPE] OF
        SET
        [DSC$K_DTYPE_T] :          VMS_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;

        [DSC$K_DTYPE_AZ,
         DSC$K_DTYPE_AC,
         DSC$K_DTYPE_VT]:          VMS_DESC[DSC$B_CLASS] = DSC$K_CLASS_VS;

        [OTHERWISE]:               RETURN FALSE;

        TES;
```

```
; 1866          1987  2
; 1867          1988  2     RETURN TRUE;
; 1868          1989  1     END;                             ! End of routine check_text_descriptor



                                                 00FC 00000 CHECK_TEXT_DESCRIPTOR:
                                                                        .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1930
                           57 00000000G   00   9E 00002             MOVAB   LIB$SIGNAL, R7                          ;
                           52        04   AC   D0 00009             MOVL    VAL_DESC, R2                            ; 1947
                           56        14   A2   9E 0000D             MOVAB   20(R2), R6                              ;
                           0D        03   A6   91 00011             CMPB    3(R6), #13                              ; 1951
                                          09   12 00015             BNEQ    1$                                      ;
                              00028D08    8F   DD 00017             PUSHL   #167176                                 ;
                           67             01   FB 0001D             CALLS   #1, LIB$SIGNAL                          ;
           18   B2         08             00   0C 00020  1$:        PROBER  #0, #8, @24(R2)                         ; 1953
                                          0E   12 00025             BNEQ    2$                                      ;
                                     18   A2   DD 00027             PUSHL   24(R2)                                  ; 1954
                                          01   DD 0002A             PUSHL   #1                                      ;
                              00028228    8F   DD 0002C             PUSHL   #164392                                 ;
                           67             03   FB 00032             CALLS   #3, LIB$SIGNAL                          ;
                66     18  B2             08   28 00035  2$:        MOVC3   #8, @24(R2), (R6)                       ; 1956
                           51        02   A6   9E 0003A             MOVAB   2(R6), R1                               ; 1958
                                          61   95 0003E             TSTB    (R1)                                    ;
                                          0E   12 00040             BNEQ    3$                                      ;
                                     03   A6   95 00042             TSTB    3(R6)                                   ;
                                          09   12 00045             BNEQ    3$                                      ;
                           61             0E   90 00047             MOVB    #14, (R1)                               ; 1961
                       03  A6             01   90 0004A             MOVB    #1, 3(R6)                               ; 1962
                                          3A   11 0004E             BRB     8$                                      ; 1963
                           50        03   A6   9E 00050  3$:        MOVAB   3(R6), R0                               ; 1966
                           02             60   91 00054             CMPB    (R0), #2                                ;
                                          03   12 00057             BNEQ    4$                                      ;
                           60             01   90 00059             MOVB    #1, (R0)                                ; 1967
                           0B             60   91 0005C  4$:        CMPB    (R0), #11                               ; 1969
                                          08   12 0005F             BNEQ    5$                                      ;
                           0E             61   91 00061             CMPB    (R1), #14                               ; 1970
                                          03   12 00064             BNEQ    5$                                      ;
                           61             25   90 00066             MOVB    #37, (R1)                               ; 1971
                           0B             60   91 00069  5$:        CMPB    (R0), #11                               ; 1973
                                          05   13 0006C             BEQL    6$                                      ;
                           01             60   91 0006E             CMPB    (R0), #1                                ; 1974
                                          1B   12 00071             BNEQ    9$                                      ;
                           0E             61   91 00073  6$:        CMPB    (R1), #14                               ; 1978
                                          05   12 00076             BNEQ    7$                                      ;
                           60             01   90 00078             MOVB    #1, (R0)                                ;
                                          0D   11 0007B             BRB     8$                                      ;
                           25             61   91 0007D  7$:        CMPB    (R1), #37                               ; 1980
                                          0C   1F 00080             BLSSU   9$                                      ;
                           27             61   91 00082             CMPB    (R1), #39                               ;
                                          07   1A 00085             BGTRU   9$                                      ;
                           60             0B   90 00087             MOVB    #11, (R0)                               ; 1982
                           50             01   D0 0008A  8$:        MOVL    #1, R0                                  ; 1988
                                               04 0008D             RET                                             ;
                                          50   D4 0008E  9$:        CLRL    R0                                      ; 1989
```

```
                                                     04 00090       RET                                                      ;

; Routine Size:  145 bytes,    Routine Base:  DBG$CODE + 0FDF
```

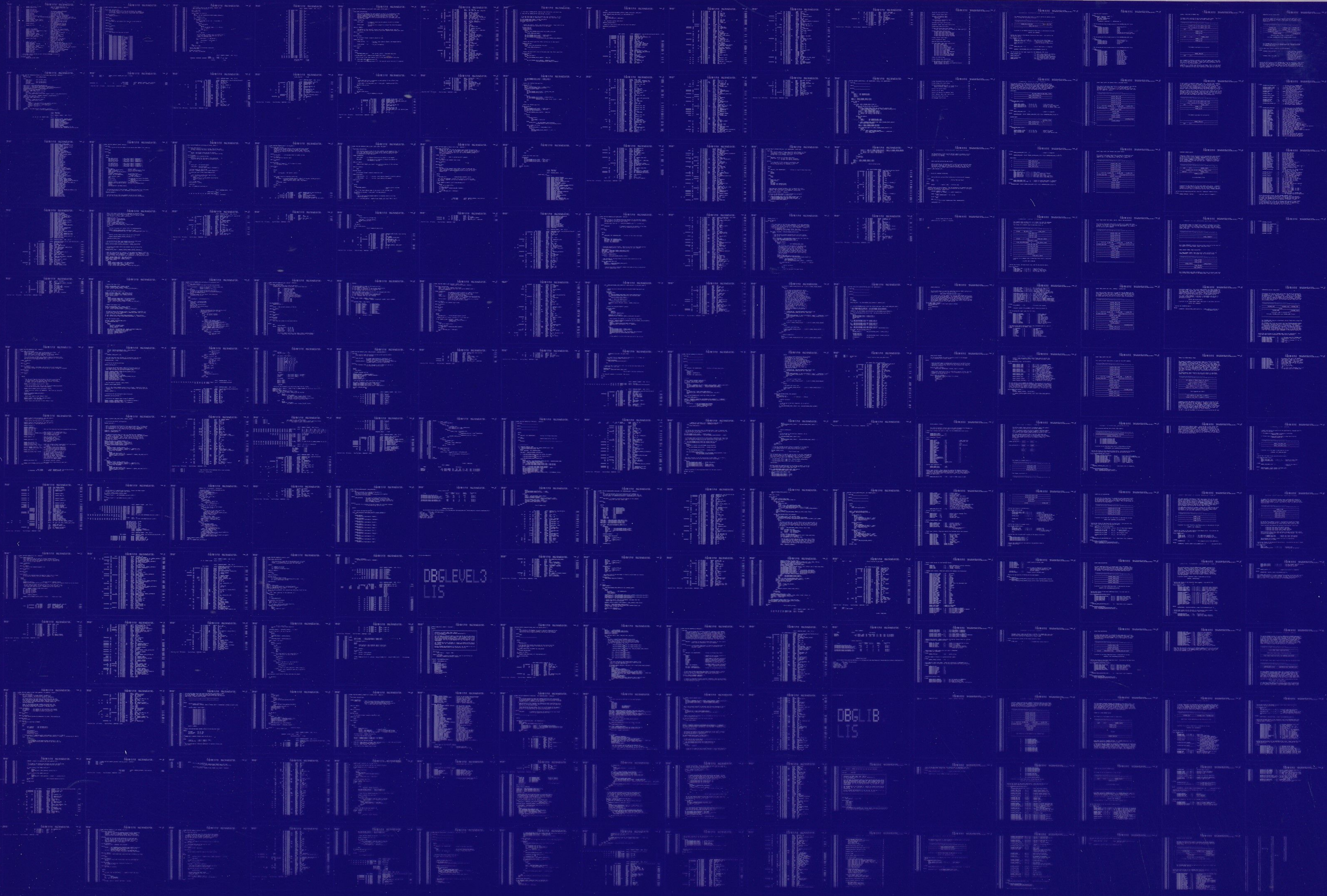
```
ROUTINE FIX_UP_LENGTH(VMS_DESC: REF DBG$STG_DESC) =
!
! FUNCTION
!           ----------------------------
!
! INPUTS
!           ----------------------------
!
! OUTPUTS
!           ----------------------------
!

    BEGIN

    LOCAL
        SIZE,
        BASE: REF BLOCK[,BYTE];


    BASE = .VMS_DESC[DSC$A_POINTER];
    SELECTONE .VMS_DESC[DSC$B_DTYPE] OF
        SET

        [DSC$K_DTYPE_VT]:
            BEGIN
            BUILTIN PROBER;
            IF NOT PROBER(%REF(0),%REF(2) ,.BASE)
            THEN SIGNAL(DBG$_NOACCESSR,1,.BASE);
            SIZE = .BASE[0,0,16,0];
            END;

        [DSC$K_DTYPE_AC]:
            BEGIN
            BUILTIN PROBER;
            IF NOT PROBER(%REF(0),%REF(1) ,.BASE)
            THEN SIGNAL(DBG$_NOACCESSR,1,.BASE);
            SIZE = .BASE[0,0, 8,0];
            END;

        [DSC$K_DTYPE_AZ]:
            BEGIN
            BUILTIN LOCC,PROBER;
            LOCAL ADDR;
            IF NOT PROBER(%REF(0),%REF(4) ,.BASE)
            THEN SIGNAL(DBG$_NOACCESSR,1,.BASE);
            LOCC(%REF(0), %REF(2048), .BASE; ,ADDR);
            SIZE = .ADDR - .BASE;
            END;

        [OTHERWISE]:
            SIZE = .VMS_DESC[DSC$W_LENGTH];

        TES;

    RETURN .SIZE;
```

```
; 1927          2047  1      END;                          ! End of routine fix_up_length



                                          001C 00000 FIX_UP_LENGTH:
                                                                   .WORD   Save R2,R3,R4                          ; 1990
                         54 00000000G   00   9E 00002              MOVAB   LIB$SIGNAL, R4
                         50        04   AC   D0 00009              MOVL    VMS_DESC, R0                           ; 2010
                         52        04   A0   D0 0000D              MOVL    4(R0), BASE
                         51        02   A0   9A 00011              MOVZBL  2(R0), R1                              ; 2011
                         25             51   91 00015              CMPB    R1, #37                                ; 2014
                                        18   12 00018              BNEQ    2$
             62          02             00   0C 0001A              PROBER  #0, #2, (BASE)                         ; 2017
                                        0D   12 0001E              BNEQ    1$
                                        52   DD 00020              PUSHL   BASE                                   ; 2018
                                        01   DD 00022              PUSHL   #1
                            00028228    8F   DD 00024              PUSHL   #164392
                         64             03   FB 0002A              CALLS   #3, LIB$SIGNAL
                         53             62   3C 0002D  1$:         MOVZWL  (BASE), SIZE                           ; 2019
                                        44   11 00030              BRB     7$                                     ; 2011
                         26             51   91 00032  2$:         CMPB    R1, #38                                ; 2022
                                        18   12 00035              BNEQ    4$
             62          01             00   0C 00037              PROBER  #0, #1, (BASE)                         ; 2025
                                        0D   12 0003B              BNEQ    3$
                                        52   DD 0003D              PUSHL   BASE                                   ; 2026
                                        01   DD 0003F              PUSHL   #1
                            00028228    8F   DD 00041              PUSHL   #164392
                         64             03   FB 00047              CALLS   #3, LIB$SIGNAL
                         53             62   9A 0004A  3$:         MOVZBL  (BASE), SIZE                           ; 2027
                                        27   11 0004D              BRB     7$                                     ; 2011
                         27             51   91 0004F  4$:         CMPB    R1, #39                                ; 2030
                                        1F   12 00052              BNEQ    6$
             62          04             00   0C 00054              PROBER  #0, #4, (BASE)                         ; 2034
                                        0D   12 00058              BNEQ    5$
                                        52   DD 0005A              PUSHL   BASE                                   ; 2035
                                        01   DD 0005C              PUSHL   #1
                            00028228    8F   DD 0005E              PUSHL   #164392
                         64             03   FB 00064              CALLS   #3, LIB$SIGNAL
             62       0800              8F   00 3A 00067  5$:      LOCC    #0, #2048, (BASE)                      ; 2036
             53                         51   52 C3 0006D           SUBL3   BASE, ADDR, SIZE                       ; 2037
                                        03   11 00071              BRB     7$                                     ; 2011
                         53             60   3C 00073  6$:         MOVZWL  (R0), SIZE                             ; 2041
                         50             53   D0 00076  7$:         MOVL    SIZE, R0                               ; 2045
                                             04 00079              RET                                            ; 2047

; Routine Size:  122 bytes,    Routine Base:  DBG$CODE + 1070


; 1928          2048  1
; 1929          2049  0 END ELUDOM



                                                                   .EXTRN  LIB$SIGNAL
```

```
;                                  PSECT SUMMARY
;
;       Name                        Bytes                    Attributes
;
;    DBG$OWN                            4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;    DBG$CODE                        4330  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(0)
;    DBG$PLIT                          74  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(0)




;                             Library Statistics
;
;                                                  -------- Symbols --------        Pages       Processing
;       File                                      Total    Loaded   Percent        Mapped      Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1                 18619        20         0          1000      00:01.9
;   _$255$DUA28:[DEBUG.OBJ]STRUCDEF.L32;1            32         1         3             7      00:00.1
;   _$255$DUA28:[DEBUG.OBJ]DBGLIB.L32;1            1545       190        12            97      00:02.0
;   _$255$DUA28:[DEBUG.OBJ]DSTRECRDS.L32;1
;                                                   418         2         0            31      00:00.3
;   _$255$DUA28:[DEBUG.OBJ]DBGMSG.L32;1             386        11         2            22      00:00.3
;   _$255$DUA28:[DEBUG.OBJ]DBGGEN.L32;1             150         0         0            12      00:00.3




;                                  COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:DBGLEVEL3/OBJ=OBJ$:DBGLEVEL3 MSRC$:DBGLEVEL3/UPDATE=(ENH$:DBGLEVEL3)

; Size:          4330 code + 78 data bytes
; Run Time:          01:11.8
; Elapsed Time:      03:41.8
; Lines/CPU Min:      1713
; Lexemes/CPU-Min: 14910
; Memory Used:  362 pages
; Compilation Complete
```

0085 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


DBGLEVEL3
LIS

DBGLIB
LIS