M 7

```
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
CCC              000         000  BBB         BBB  RRR         RRR        TTT        LLL
CCC              000         000  BBB         BBB  RRR         RRR        TTT        LLL
CCC              000         000  BBB         BBB  RRR         RRR        TTT        LLL
CCC              000         000  BBB         BBB  RRR         RRR        TTT        LLL
CCC              000         000  BBB         BBB  RRR         RRR        TTT        LLL
CCC              000         000  BBB         BBB  RRR         RRR        TTT        LLL
CCC              000         000  BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
CCC              000         000  BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
CCC              000         000  BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
CCC              000         000  BBB         BBB  RRR   RRR              TTT        LLL
CCC              000         000  BBB         BBB  RRR   RRR              TTT        LLL
CCC              000         000  BBB         BBB  RRR   RRR              TTT        LLL
CCC              000         000  BBB         BBB  RRR      RRR           TTT        LLL
CCC              000         000  BBB         BBB  RRR      RRR           TTT        LLL
CCC              000         000  BBB         BBB  RRR      RRR           TTT        LLL
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
                                                    F  1
**FILE**ID**COBDBEXCE




  CCCCCCCC      OOOOOO      BBBBBBBB    DDDDDDDD    BBBBBBBB    EEEEEEEEEE  XX      XX    CCCCCCCC  EEEEEEEEEE
  CCCCCCCC      OOOOOO      BBBBBBBB    DDDDDDDD    BBBBBBBB    EEEEEEEEEE  XX      XX    CCCCCCCC  EEEEEEEEEE
CC            OO      OO    BB      BB  DD      DD  BB      BB  EE          XX      XX  CC          EE
CC            OO      OO    BB      BB  DD      DD  BB      BB  EE          XX      XX  CC          EE
CC            OO      OO    BB      BB  DD      DD  BB      BB  EE            XX  XX    CC          EE
CC            OO      OO    BB      BB  DD      DD  BB      BB  EE            XX  XX    CC          EE
CC            OO      OO    BBBBBBBB    DD      DD  BBBBBBBB    EEEEEEEE        XX      CC          EEEEEEEE
CC            OO      OO    BBBBBBBB    DD      DD  BBBBBBBB    EEEEEEEE        XX      CC          EEEEEEEE
CC            OO      OO    BB      BB  DD      DD  BB      BB  EE            XX  XX    CC          EE
CC            OO      OO    BB      BB  DD      DD  BB      BB  EE            XX  XX    CC          EE
CC            OO      OO    BB      BB  DD      DD  BB      BB  EE          XX      XX  CC          EE              ....
CC            OO      OO    BB      BB  DD      DD  BB      BB  EE          XX      XX  CC          EE              ....
  CCCCCCCC      OOOOOO      BBBBBBBB    DDDDDDDD    BBBBBBBB    EEEEEEEEEE  XX      XX    CCCCCCCC  EEEEEEEEEE      ....
  CCCCCCCC      OOOOOO      BBBBBBBB    DDDDDDDD    BBBBBBBB    EEEEEEEEEE  XX      XX    CCCCCCCC  EEEEEEEEEE      ....

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

```
0 MODULE COB$DBEXCEPTION(
0                     IDENT = '1-011'          ! file:COBDBEXCE.B32 Edit:STAN1011
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 !++
1 ! FACILITY:     COBOL SUPPORT
1 !
1 ! ABSTRACT
1 !
1 !       This procedure is called from compiled code when a data base
1 !       exception condition occurs.  This procedure looks for an applicable
1 !       USE procedure to handle the data base exception condition.
1 !       If one is not found, it invokes LIB$STOP to handle the data base
1 !       exception condition.
1 !
1 !
1 ! ENVIRONMENT:  Vax-11 User Mode
1 !
1 ! AUTHOR: RKR , CREATION DATE: 11-FEB-1981
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original Skeleton.   RKR  11-FEB-1981
1 ! 1-002 - Actual code added.   LB   11-MAR-1981
1 ! 1-003 - Took out code that checked the DB code field (COB$B_USE CODE
1 !         equal to COB$K_DBUSE_CODE)  This code was moved to
1 !         COB$$HANDLER.  LB  16-MAR-81
1 ! 1-004 - Added diagrams of the data structures involved in COB$DBEXCEPTION.
1 !         Also compacted code and changed macro definition name to have a
1 !         DBMS prefix.  LB  19-MAR-81
1 ! 1-005 - Replaced arbitrary signalling value for no DB USE procedure found
1 !         with appropriate symbol name now defined in COBMSGDEF.  Added
1 !         corresponding entry in the EXTERNAL LITERAL declarations for
```

```
1 !             this module.  LB  24-MAR-81
1 ! 1-006 - Changed name of data base external literal to correspond to change
1 !             made in COBMSG.MDL.  Changed return status of SS$_CONTINUE to
1 !             SS$_NORMAL.  Changed calls to SIGNAL_STOP to be to LIB$STOP for
1 !             consistency reasons.  Changed calls to LIB$SIGNAL to now take an
1 !             FAO parameter (as is syntactically correct) even though the
1 !             parameter that is getting passed is not an FAO parameter (note that
1 !             the !+ directive in the message text will ignore it).  Added code
1 !             back into this routine that had previously been taken out (refer
1 !             to revision history 1-003); that code now resides in both places
1 !             (here and in COB$HANDLER).  Added heaps of comments and optimized
1 !             code by encasing calls to LIB$STOP in BEGIN-END blocks.  LB  16-APR-81
1 ! 1-007 - Changed search code for a DB USE procedure due to hidden design flaws
1 !             in the original search algorithm.  Code now saves the first match of
1 !             a USE procedure and continues searching through the USE list for a
1 !             locally defined one.  If one is found, then the local one is
1 !             invoked; otherwise, the saved procedure is invoked.  Note that this
1 !             code is executed only for the case where there is an ON OTHER clause
1 !             (meaning that COB$L_USE_LIT = 0) and when the looping count has not
1 !             reached its limit.  LB  21-APR-81
1 ! 1-008 - Changed check of looping count to check .I instead of
1 !             .DBUSE[COB$B_DBUSE_CNT].  Moved code that returned SS$_NORMAL to
1 !             follow directly after the call to LIB$SIGNAL instead of after the
1 !             END directive.  Added code to return SS$_NORMAL within the
1 !             BEGIN-END blocks at every occurrence of an invocation of a USE
1 !             procedure to correct flow problems.  LB  11-MAY-81.
1 ! 1-009 - Minor improvements to generated code.  PDG  9-AUG-81
1 ! 1-010 - Declare LIB$SIGNAL external.  SBL 2-Dec-1981
1 ! 1-011 - Remove informational errors. STAN 24-Jul-1984.
1 !
1 !--
1
1 !<BLF/PAGE>
```

```
!+
! SWITCHES:
!-

SWITCHES ADDRESSING_MODE
                (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);

!+
! LINKAGES:
!
!       NONE
!-

!+
! TABLE OF CONTENTS:
!-

FORWARD ROUTINE
        COB$DBEXCEPTION ;

!+
! INCLUDE FILES:
!
! Note that there is no require file for the data base
! exception codes.  These will be provided during link
! time as link-time constants.
!-


REQUIRE 'RTLIN:COBDEF';                         ! To find USE procedures
REQUIRE 'RTLIN:RTLPSECT' ;                      ! Macros for defining psects
LIBRARY 'RTLSTARLE';

!+
! MACROS
!-

MACRO

        DBMS_COND_VAL   =       4,0,32,0%;      ! Data Base Condition Value (status code)
```

```
!+
! EQUATED SYMBOLS
!
!       NONE
!-

!+
!
! PSECT DECLARATIONS:
!-

DECLARE_PSECTS (COB) ;                          ! Declare psects for COB$ facility

!+
! EXTERNAL REFERENCES:
!-

EXTERNAL ROUTINE
        LIB$SIGNAL,                             ! Signal
        LIB$STOP,                               ! Signal_stop
        COB$$INVOKE_USE : NOVALUE;              ! Invoke USE procedure


EXTERNAL LITERAL

!+
!  Condition codes we need
!
!  Note that the !+ directive for the data base literal indicates to ignore
!  the next parameter in the LIB$SIGNAL parameter list.  This is needed in
!  the case where the call to LIB$SIGNAL passes a condition value and an
!  address of a USE procedure which gets checked in COB$HANDLER.  The address
!  of the USE procedure is not a FAO parameter, but an FAO count must still
!  be included (as is syntactically correct) in the call to LIB$SIGNAL in
!  the case that COB$HANDLER resorts to re-signalling the error, causing an
!  error message to be printed at user level.  At that point, the address of
!  the USE procedure parameter must not be treated as an FAO parameter -
!  therefore the !+ takes care of it (by ignoring the parameter).
!
!-

    COB$_LSTHNDLDB,     ! Lost handler for data base exception - environment corrupted !+
    OTS$_FATINTERR;     ! Fatal Internal Error
```

```
GLOBAL ROUTINE COB$DBEXCEPTION (

        DBMS_STAT                                       !\Ptr to a block which contains
                                                        !/the condition value & error msg.
                                ) =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This procedure is called from compiled code when a data base
!       exception occurs when accessing the data base.  This procedure
!       looks for an applicable USE procedure to handle the error; if
!       it can't find one, it LIB$STOP's.  If it finds one defined in
!       the local program, it invokes the USE procedure; if it is not
!       defined in the local program, it signals the data base exception
!       condition which should then get processed in COB$HANDLER.
!
!
! CALLING SEQUENCE:
!
!       COB$DBEXCEPTION (dbms_stat.rr.r)
!
! FORMAL PARAMETERS:
!
!       DBMS_STAT.rr.r          Ptr to a block which contains the error
!                               and the condition value to be LIB$SIGNAL'ed
!                               (i.e. a message vector)
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       An error message may be signalled.
!
! COMPLETION CODE:
!
!       Returns SS$_NORMAL if success (LSB = 1)
!       Otherwise, returns a zero.
!
! SIDE EFFECTS:
!
!       Calls LIB$STOP if input parameters are insufficient or invalid.
!       Also invokes LIB$STOP if the frame pointer equals 0 (indicating
!       serious problems).
!
! NOTES:
!
!       For more information on the message vector, refer to
!       the $PUTMSG system service documentation.
!
!--
```

```
1 !+
1 !
1 ! Note that the DB USE list is structured as follows:
1 !
1 !       *****************************************************
1 !       *                                    !COB$B_USE_CODE *
1 !       *---------------------------------------------------*
1 !       *               COB$A_DBUSE_PNC                     *
1 !       *---------------------------------------------------*
1 !       *                    !COB$B_GDBUSE_CNT!COB$B_DBUSE_CNT*
1 !       *---------------------------------------------------*
1 !       *               COB$A_USE_PROC                      * BASE OF 1ST DATA BASE ENTRY
1 !       *---------------------------------------------------*
1 !       *               COB$A_USE_EOPR                      *
1 !       *---------------------------------------------------*
1 !       *               COB$L_USE_LIT                       *
1 !       *****************************************************
1 !
1 !-
1
1 !+
1 !
1 ! Note that the above fields are defined as follows:
1 !
1 !       COB$B_USE_CODE - generic code indicating that the USE list
1 !                        pertains to data base exceptions.
1 !       COB$A_DBUSE_PNC -address of Perform Nest Counter for declaring
1 !                        program.
1 !       COB$B_DBUSE_CNT -number of data base USE procedures defined in
1 !                        this program.  This includes both local and
1 !                        global procedures defined in both this program
1 !                        and containing programs.
1 !       COB$B_GDBUSE_CNT-number of global data base USE procedures
1 !                        defined in the local program.
1 !       COB$A_USE_PROC - address of data base USE procedure.
1 !       COB$A_USE_EOPR - pointer to the end of the Perform Range Block
1 !                        for the USE procedure if the entry was defined
1 !                        in this program or 0 if it was defined in a
1 !                        containing program.
1 !       COB$L_USE_LIT  - a data base exception literal or 0 for "ON OTHER"
1 !                        or no "ON".
1 !-
```

```
1
1 !+
1 ! Note that this is the DBMS_STAT block (referred to as
1 ! a message vector) that is the input parameter to
1 ! COB$DBEXCEPTION.  Note that this particular example
1 ! is for one that contains FAO parameters.  There can
1 ! be other formats, where the condition code is either
1 ! an RMS status or a system service status, which in
1 ! both cases, do not take any FAO parameters.  Also note
1 ! that the numbers in parentheses are only meaningful for
1 ! this sample block.  The # of longwords in a DBMS_STAT
1 ! block can be up to 255.  Also, the # of FAO arguments
1 ! that can be passed in this block can be up to 16 for
1 ! DBMS.
1 !
1 !
1 !       *****************************************
1 !       *       # OF LONGWORDS IN BLOCK (6)     *
1 !       *---------------------------------------*
1 !       *     DBMS CONDITION CODE (STATUS)      *
1 !       *---------------------------------------*
1 !       *         0        !  # OF FAO ARGS (4) *
1 !       *---------------------------------------*
1 !       *              FAO ARG #1               *
1 !       *---------------------------------------*
1 !       *              FAO ARG #2               *
1 !       *---------------------------------------*
1 !       *              FAO ARG #3               *
1 !       *---------------------------------------*
1 !       *              FAO ARG #4               *
1 !       *****************************************
1 !-
```

```
2 BEGIN
2
2       BUILTIN
2           CALLG,
2           ACTUALPARAMETER,
2           ACTUALCOUNT,
2           FP;
2
2       MAP
2           DBMS_STAT:          REF BLOCK[,BYTE],
2           FP:                 REF BLOCK[,BYTE];
2
2       LOCAL
2           SFP:                REF BLOCK[,BYTE],          ! Saved FP
2           DBUSE:              REF BLOCK[,BYTE],          ! Pointer to DB USE list
2           DBUSE_ENT:          REF BLOCK[,BYTE],          ! Pointer to DB USE list entry
2           SAVE_PNC:           VOLATILE,                  ! Saved addr of PNC
2           SAVE_EOPR:          VOLATILE,                  ! Saved addr of EOPR
2           SAVE_SAVED_AP:      VOLATILE,                  ! Saved saved AP
2           SAVE_ADDR_USELST: VOLATILE,                    ! Saved addr of USE list
2           SAVE_ADDR_USEPROC;                             ! Saved addr of USE procedure
2
2       !+
2       ! Ensure that the DBMS_STAT argument is present
2       !-
2
3       IF (IF ACTUALCOUNT() EQL 0 THEN 1 ELSE .DBMS_STAT EQL 0)
2         THEN
3           BEGIN
3           LIB$STOP (OTS$_FATINTERR);
3           RETURN 0;
2           END;
2
2       !+
2       ! Initialize the local storage for the
2       ! saved address of the USE procedure
2       ! for later use.
2       !-
2
2       SAVE_ADDR_USEPROC = 0;
2
2       !+
2       ! Search for an appropriate USE procedure.
2       !-
2
2       SFP = .FP[SF$L_SAVE_FP];                           ! Get saved FP
2       IF .SFP EQL 0                                      !\If FP equals zero,
2         THEN                                             ! then we have
3           BEGIN                                          !/serious problems
3           LIB$STOP (OTS$_FATINTERR);
3           RETURN 0;
3           END
2         ELSE
3           BEGIN
3           DBUSE = .SFP[COB$A_DB_USE];                    !\Fetch address at offset
3                                                          ! 8 from stack to obtain
3                                                          !/a ptr to a DB USE list
```

```
 3
 3      IF .DBUSE NEQ 0
 3        THEN
 4          BEGIN
 4
 4      !+
 4      ! The following check determines if this is a
 4      ! data base USE list.  The COB$B_USE_CODE field
 4      ! should contain the generic code for the class of
 4      ! data base exceptions (equal to COB$K_DBUSE_CODE).
 4      !-
 4
 4          IF .DBUSE[COB$B_USE_CODE] EQL COB$K_DBUSE_CODE
 4
 4            THEN
 5              BEGIN
 5              DBUSE_ENT = DBUSE[COB$A_DBUSE_ENT];     ! Point to 1st DB USE entry
 5
 5
 5              !+
 5              ! Find an applicable USE procedure.  A USE procedure
 5              ! is applicable if COB$L_USE_LIT equals the data base
 5              ! exception or if COB$L_USE_LIT equals zero.
 5              !-
 5
 5              DECR I FROM .DBUSE[COB$B_DBUSE_CNT] - 1 TO 0 DO
 6                BEGIN
 6                IF .DBUSE_ENT[COB$L_USE_LIT] EQL 0 OR
 6                .DBUSE_ENT[COB$L_USE_LIT] EQL .DBMS_STAT[DBMS_COND_VAL]
 6                  THEN
 7                    BEGIN
 7
 7                  !+
 7                  ! If EOPR (ptr to end of perform range block)
 7                  ! not equal to zero, then we know that
 7                  ! the USE procedure is local, and we can invoke
 7                  ! it immediately.  Otherwise, the data base
 7                  ! error should be LIB$SIGNAL'ed.
 7                  !-
 7
 7                    IF .DBUSE_ENT[COB$A_USE_EOPR] NEQ 0
 7                      THEN
 8                        BEGIN
 8                        IF .DBUSE_ENT[COB$L_USE_LIT] EQL 0
 8                          THEN
 9                            BEGIN
 9
 9                          !+
 9                          ! If the looping count equals zero, (meaning
 9                          ! there aren't any more USE procedures in the
 9                          ! list), then we can invoke this USE procedure.
 9                          !-
 9
 9                            IF .I EQL 0
 9                              THEN
10                                BEGIN
10                                COB$$INVOKE_USE (
;  421        0950 10                              .DBUSE_ENT[COB$A_USE_PROC],  ! Addr of USE procedure
;  422        0951 10                              .DBUSE,                      ! Addr of USE list
;  423        0952 10                              .FP[SF$L_SAVE_AP],
;  424        0953 10                              .DBUSE_ENT[COB$A_USE_EOPR],  ! EOPR
;  425        0954 10                              .DBUSE[COB$A_DBUSE_PNC]);    ! Perform Nest Ctr
;  426        0955 10                           RETURN SS$_NORMAL;
;  427        0956 10                           END
;  428        0957  9                        ELSE
;  429        0958  9
;  430        0959  9                        !+
;  431        0960  9                        ! At this point, we have found a USE procedure,
;  432        0961  9                        ! but rather than invoking it immediately, we
;  433        0962  9                        ! save its address and all other context required
;  434        0963  9                        ! to invoke the USE procedure, and keep on
;  435        0964  9                        ! searching for an additional match.  This is
;  436        0965  9                        ! due to the fact that the USE list is arranged
;  437        0966  9                        ! in the order of global USE procedures defined
;  438        0967  9                        ! in the local program, followed by local USE
;  439        0968  9                        ! procedures defined in the local program.  This
;  440        0969  9                        ! method of saving the context of the 1st matched
;  441        0970  9                        ! USE procedure now ensures that if a local USE
;  442        0971  9                        ! procedure defined in the local program exists
;  443        0972  9                        ! that it will be invoked rather than a globally
;  444        0973  9                        ! defined one.
;  445        0974  9                        !-
;  446        0975  9
;  447        0976 10                           BEGIN
;  448        0977 10                           SAVE_ADDR_USEPROC = .DBUSE_ENT[COB$A_USE_PROC];
;  449        0978 10                           SAVE_ADDR_USELST = .DBUSE;
;  450        0979 10                           SAVE_SAVED_AP = .FP[SF$L_SAVE_AP];
;  451        0980 10                           SAVE_EOPR = .DBUSE_ENT[COB$A_USE_EOPR];
;  452        0981 10                           SAVE_PNC = .DBUSE[COB$A_DBUSE_PNC];
;  453        0982  9                           END;
;  454        0983  9                     END
;  455        0984  8                  ELSE
;  456        0985  8
;  457        0986  8                  !+
;  458        0987  8                  ! Here we know that there is no "ON OTHER"
;  459        0988  8                  ! clause (meaning that COB$A_USE_LIT is not
;  460        0989  8                  ! equal to zero), so we can just invoke the
;  461        0990  8                  ! found USE procedure.
;  462        0991  8                  !-
;  463        0992  8
;  464        0993  9                     BEGIN
;  465        0994  9                     COB$$INVOKE_USE (
;  466        0995  9                        .DBUSE_ENT[COB$A_USE_PROC],      ! Addr of USE procedure
;  467        0996  9                        .DBUSE,                          ! Addr of USE list
;  468        0997  9                        .FP[SF$L_SAVE_AP],
;  469        0998  9                        .DBUSE_ENT[COB$A_USE_EOPR],      ! EOPR
;  470        0999  9                        .DBUSE[COB$A_DBUSE_PNC]);        ! Perform Nest Ctr
;  471        1000  9                     RETURN SS$_NORMAL;
;  472        1001  8                     END;
;  473        1002  8               END
;  474        1003  7            ELSE
;  475        1004  8               BEGIN
;  476        1005  8
;  477        1006  8            !+
8                       ! At this point, if "SAVE_ADDR_USEPROC" not
8                       ! equal to zero, then we had previously found
8                       ! a USE procedure which is a globally defined
8                       ! one in the local program, and saved its context.
8                       ! We can now invoke it.
8                       !-
8
8                         IF .SAVE_ADDR_USEPROC NEQ 0
8                           THEN
8                             COB$$INVOKE_USE (
8                               .SAVE_ADDR_USEPROC,          ! Addr of USE procedure
8                               .SAVE_ADDR_USELST,           ! Addr of USE list
8                               .SAVE_SAVED_AP,              ! Saved AP
8                               .SAVE_EOPR,                  ! Addr of EOPR
8                               .SAVE_PNC)                   ! Addr of PNC
8                           ELSE
8
8                       !+
8                       ! Signal the error here indicating that the USE
8                       ! procedure is a global one that is in a containing
8                       ! program.  Need to get the FP of the containing
8                       ! program and check the defined global USE
8                       ! procedures of the local program there.  If found,
8                       ! then invoke the USE procedure at that level;
8                       ! otherwise, the error is re-signalled.
8                       !-
8
8                             LIB$SIGNAL (COB$_LSTHNDLDB,1,.DBUSE_ENT[COB$A_USE_PROC]);
8                         RETURN SS$_NORMAL;
7                         END;
6                     END;
6                 DBUSE_ENT = .DBUSE_ENT + COB$S_DBUSE;        ! Step to next entry
5                 END;                                         ! End of DECR loop
5
5           !+
5           ! At this point, we know that there isn't an
5           ! applicable USE procedure, so the only recourse
5           ! we have is to LIB$STOP.  The reason for this is that
5           ! the COB$B_USE_CODE did not match with the code
5           ! provided in COB$K_DBUSE_CODE.  We also could have
5           ! arrived here if DBUSE was equal to zero, indicating
5           ! there was no DB USE list to begin with.
5           !--
5
4                 END;                                         !\End of code block that
4                                                              ! checks for a COB$K_DBUSE_CODE
4                                                              !/match
3               END;                                           !\End of code dealing with
3                                                              !/a valid ptr to a DB USE list
3             CALLG (.DBMS_STAT,LIB$STOP);
2           END;
2     RETURN 0                                                 ! Never gets here
2                                                              !\End of global routine
1     END;                                                     !/COB$DBEXCEPTION

                                                  .TITLE  COB$DBEXCEPTION
```

```
                                                       .IDENT   \1-011\

                                                       .EXTRN   LIB$SIGNAL, LIB$STOP
                                                       .EXTRN   COB$$INVOKE_USE
                                                       .EXTRN   COB$_LSTHNDLDB, OTS$_FATINTERR

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

                               00FC 00000              .ENTRY   COB$DBEXCEPTION, Save R2,R3,R4,R5,R6,R7        ; 0709
          57 00000000G   00     9E 00002               MOVAB    LIB$STOP, R7                                   ;
          5E             10     C2 00009               SUBL2    #16, SP                                        ;
                         6C     95 0000C               TSTB     (AP)                                           ; 0862
                         0D     13 0000E               BEQL     1$                                             ;
                   04    AC     D5 00010               TSTL     DBMS_STAT                                      ;
                         08     13 00013               BEQL     1$                                             ;
                         56     D4 00015               CLRL     SAVE_ADDR_USEPROC                              ; 0875
          52       1C    AE     D0 00017               MOVL     28(FP), SFP                                    ; 0881
                         0C     12 0001B               BNEQ     2$                                             ; 0882
             00000000G   8F     DD 0001D 1$:           PUSHL    #OTS$_FATINTERR                                ; 0885
          67             01     FB 00023               CALLS    #1, LIB$STOP                                   ;
                       0099     31 00026               BRW      16$                                            ; 0886
          52       F8    A2     D0 00029 2$:           MOVL     -8(SFP), DBUSE                                 ; 0890
                         03     12 0002D               BNEQ     4$                                             ; 0894
                       008C     31 0002F 3$:           BRW      15$                                            ;
          01             62     91 00032 4$:           CMPB     (DBUSE), #1                                    ; 0905
                         F8     12 00035               BNEQ     3$                                             ;
          53       0C    A2     9E 00037               MOVAB    12(R2), DBUSE_ENT                              ; 0909
          55       08    A2     9A 0003B               MOVZBL   8(DBUSE), I                                    ; 0918
                         7A     11 0003F               BRB      14$                                            ;
                         51     D4 00041 5$:           CLRL     R1                                             ; 0920
                   08    A3     D5 00043               TSTL     8(DBUSE_ENT)                                   ;
                         04     12 00046               BNEQ     6$                                             ;
                         51     D6 00048               INCL     R1                                             ;
                         0B     11 0004A               BRB      7$                                             ;
          50       04    AC     D0 0004C 6$:           MOVL     DBMS_STAT, R0                                  ; 0921
    04    A0       08    A3     D1 00050               CMPL     8(DBUSE_ENT), 4(R0)                            ;
                         61     12 00055               BNEQ     13$                                            ;
          54       04    A3     D0 00057 7$:           MOVL     4(DBUSE_ENT), R4                               ; 0933
                         2B     13 0005B               BEQL     9$                                             ;
          1A             51     E9 0005D               BLBC     R1, 8$                                         ; 0952
                         55     D5 00060               TSTL     I                                              ; 0946
                         16     13 00062               BEQL     8$                                             ;
          56             63     D0 00064               MOVL     (DBUSE_ENT), SAVE_ADDR_USEPROC                 ; 0977
          6E             52     D0 00067               MOVL     DBUSE, SAVE_ADDR_USELST                        ; 0978
    04    AE       18    AE     D0 0006A               MOVL     24(FP), SAVE_SAVED_AP                          ; 0979
    08    AE             54     D0 0006F               MOVL     R4, SAVE_EOPR                                  ; 0980
    0C    AE       04    A2     D0 00073               MOVL     4(DBUSE), SAVE_PNC                             ; 0981
                         3E     11 00078               BRB      13$                                            ; 0936
                   04    A2     DD 0007A 8$:           PUSHL    4(DBUSE)                                       ; 0999
                         54     DD 0007D               PUSHL    R4                                             ; 0998
                   20    AE     DD 0007F               PUSHL    32(FP)                                         ; 0997
                         52     DD 00082               PUSHL    DBUSE                                          ; 0996
                         63     DD 00084               PUSHL    (DBUSE_ENT)                                    ; 0995
                         12     11 00086               BRB      10$                                            ;
                         56     D5 00088 9$:           TSTL     SAVE_ADDR_USEPROC                              ; 1014
                         17     13 0008A               BEQL     11$                                            ;
                   0C    AE     DD 0008C               PUSHL    SAVE_PNC                                       ; 1021
```

```
                              0C   AE  DD 0008F         PUSHL   SAVE_EOPR                        ; 1020
                              0C   AE  DD 00092         PUSHL   SAVE_SAVED_AP                    ; 1019
                              0C   AE  DD 00095         PUSHL   SAVE_ADDR_USELST                 ; 1018
                                   56  DD 00098         PUSHL   SAVE_ADDR_USEPROC                ; 1017
        00000000G  00              05  FB 0009A 10$:    CALLS   #5, COB$$INVOKE_USE              ;
                                   11  11 000A1         BRB     12$                              ; 1016
                                   63  DD 000A3 11$:    PUSHL   (DBUSE_ENT)                      ; 1034
                                   01  DD 000A5         PUSHL   #1                               ;
                       00000000G   8F  DD 000A7         PUSHL   #COB$_LSTHNDLDB                  ;
        00000000G  00              03  FB 000AD         CALLS   #3, LIB$SIGNAL                   ;
                   50              01  D0 000B4 12$:    MOVL    #1, R0                           ; 1035
                                       04 000B7         RET                                      ;
                   53              0C  C0 000B8 13$:    ADDL2   #12, DBUSE_ENT                   ; 1038
                   83              55  F4 000BB 14$:    SOBGEQ  I, 5$                            ; 0918
                   67         04   BC  FA 000BE 15$:    CALLG   @DBMS_STAT, LIB$STOP             ; 1056
                                   50  D4 000C2 16$:    CLRL    R0                               ; 1060
                                       04 000C4         RET                                      ;

; Routine Size:  197 bytes,    Routine Base:  _COB$CODE + 0000


;  532          1061  1 END
;  533          1062  0 ELUDOM




;                              PSECT SUMMARY
;
;     Name                    Bytes                      Attributes
;
;  _COB$CODE                      197  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                        Library Statistics
;
;                                          -------- Symbols --------       Pages      Processing
;     File                                 Total    Loaded   Percent       Mapped     Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1        9776         3         0         581        00:00.8




;                              COMMAND QUALIFIERS

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

; Size:        197 code + 0 data bytes
; Run Time:          00:06.7
; Elapsed Time:      00:30.6
; Lines/CPU Min:      9581
; Lexemes/CPU-Min:  25181
; Memory Used:  115 pages
; Compilation Complete
```

0062 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

COBDIVQ
LIS

COBFINDNA
LIS

COBDBEXCE
LIS

COBEXPI
LIS

COBDEEDIT
LIS

COBDISPLA
LIS

COBESCGEN
LIS

COBERROR
LIS

COBDHANDL
LIS