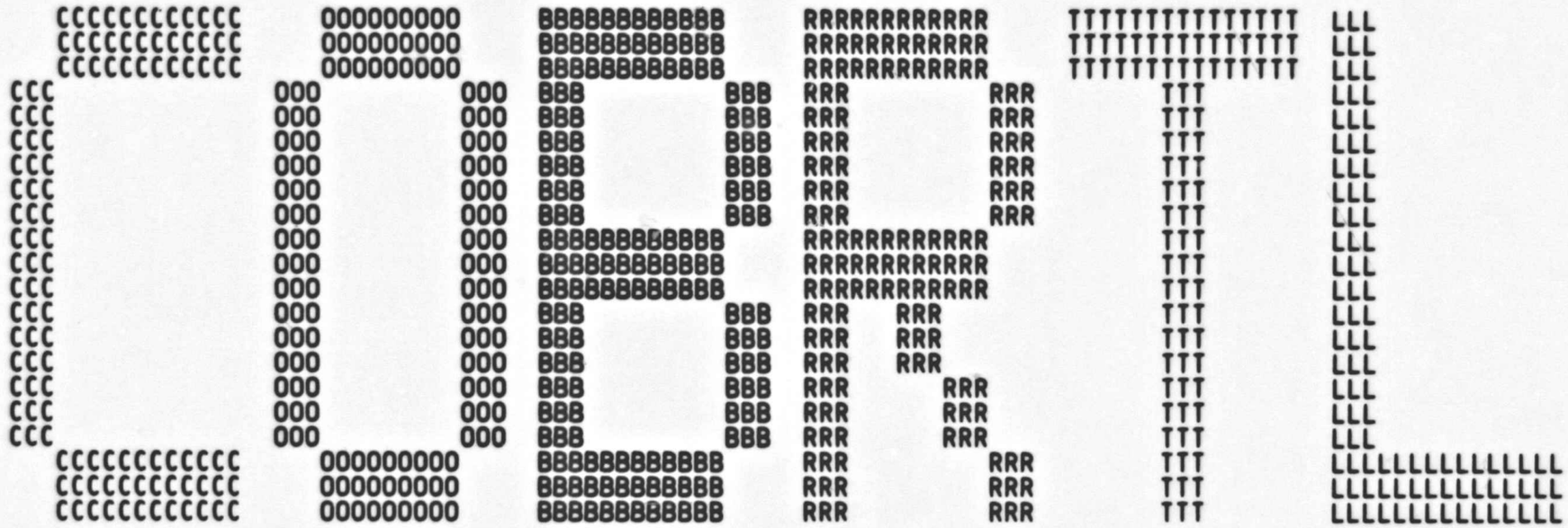
COBRA

```
  CCCCCCCC    OOOOOO    BBBBBBBB    HH      HH    AAAAAA    NN      NN  DDDDDDDD    LL          EEEEEEEEEE
  CCCCCCCC    OOOOOO    BBBBBBBB    HH      HH    AAAAAA    NN      NN  DDDDDDDD    LL          EEEEEEEEEE
CC          OO      OO  BB      BB  HH      HH  AA      AA  NN      NN  DD      DD  LL          EE
CC          OO      OO  BB      BB  HH      HH  AA      AA  NN      NN  DD      DD  LL          EE
CC          OO      OO  BB      BB  HH      HH  AA      AA  NNNN    NN  DD      DD  LL          EE
CC          OO      OO  BB      BB  HH      HH  AA      AA  NNNN    NN  DD      DD  LL          EE
CC          OO      OO  BBBBBBBB    HHHHHHHHHH  AA      AA  NN  NN  NN  DD      DD  LL          EEEEEEEE
CC          OO      OO  BBBBBBBB    HHHHHHHHHH  AA      AA  NN  NN  NN  DD      DD  LL          EEEEEEEE
CC          OO      OO  BB      BB  HH      HH  AAAAAAAAAA  NN    NNNN  DD      DD  LL          EE
CC          OO      OO  BB      BB  HH      HH  AAAAAAAAAA  NN    NNNN  DD      DD  LL          EE
CC          OO      OO  BB      BB  HH      HH  AA      AA  NN      NN  DD      DD  LL          EE                ....
CC          OO      OO  BB      BB  HH      HH  AA      AA  NN      NN  DD      DD  LL          EE                ....
  CCCCCCCC    OOOOOO    BBBBBBBB    HH      HH  AA      AA  NN      NN  DDDDDDDD    LLLLLLLLLL  EEEEEEEEEE        ....
  CCCCCCCC    OOOOOO    BBBBBBBB    HH      HH  AA      AA  NN      NN  DDDDDDDD    LLLLLLLLLL  EEEEEEEEEE        ....

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

```
;     1          0001  0 MODULE COB$$HANDLER (
;     2          0002  0                 IDENT = '1-022'         ! FILE: COBHANDLE.B32 EDIT:PDG1022
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !****************************************************************************
;     7          0007  1 !*                                                                          *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                     *
;    11          0011  1 !*                                                                          *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  1 !*  TRANSFERRED.                                                             *
;    18          0018  1 !*                                                                          *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  1 !*  CORPORATION.                                                             *
;    22          0022  1 !*                                                                          *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    25          0025  1 !*                                                                          *
;    26          0026  1 !*                                                                          *
;    27          0027  1 !****************************************************************************
;    28          0028  1 !
;    29          0029  1 !
;    30          0030  1 !++
;    31          0031  1 ! FACILITY:     COBOL SUPPORT
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:     This procedure is the error handler for COBOL error
;    34          0034  1 !               conditions.  It gets invoked as a result of a call
;    35          0035  1 !               to LIB$SIGNAL.
;    36          0036  1 !
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  Vax-11 User Mode
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: MLJ , CREATION DATE: 03-MAY-1979
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original. MLJ  03-MAY-1979
;    45          0045  1 ! 1-002 - Added boilerplate and comments. RKR 18-JULY-1979
;    46          0046  1 ! 1-003 - Declare psects via library macro. RKR 23-AUG-1979
;    47          0047  1 ! 1-004 - Change symbolic name of LIBRARY file. RKR 1-OCT-79
;    48          0048  1 ! 1-005 - Change name of entry point to COB$$HANDLER. RKR 18-OCT-79
;    49          0049  1 ! 1-006 - Remove definition of COB$_USE_EXIT. Cosmetic changes.
;    50          0050  1 !         RKR 20-OCT-79
;    51          0051  1 ! 1-007 - But forgot to declare it as EXTERNAL LITERAL.
;    52          0052  1 !         RKR 20-OCT-79
;    53          0053  1 ! 1-008 - Check for COB$_USE_EXIT by using LIB$MATCH_COND.
;    54          0054  1 !         RKR 22-OCT-79
;    55          0055  1 ! 1-009 - Make arguments to LIBMATCHCOND be by REF.  RKR 31-OCT-79
;    56          0056  1 ! 1-010 - Add code to resignal a COBOL-specific error message if
;    57          0057  1 !         signal is SS$_ROPRAND and associated opcode was
```

```
;    58          0058  1 !            CVTTP or CVTSP.  RKR 21-NOV-79
;    59          0059  1 !    1-011 - Correct resignaling code.  RKR 27-NOV-79
;    60          0060  1 !    1-012 - Added abstract, functional description, comments and made
;    61          0061  1 !            cosmetic changes.  Added code that breaks up the CASE code
;    62          0062  1 !            for a USE procedure condition value into the appropriate types -
;    63          0063  1 !            file specific and mode specific.  This addition of code
;    64          0064  1 !            also involved introducing two new condition values and
;    65          0065  1 !            symbolic names for those values.  LB  3-MAR-81
;    66          0066  1 !    1-013 - Added comments.  LB 09-MAR-81
;    67          0067  1 !    1-014 - Added code for handling a data base USE procedure condition
;    68          0068  1 !            code (as a result of the new routine COB$DBEXCEPTION).  Also
;    69          0069  1 !            changed the range of the CASE statement from 0-3 to 0-5 to
;    70          0070  1 !            account for new error conditions.  And added yet more
;    71          0071  1 !            comments.  LB  12-MAR-81
;    72          0072  1 !    1-015 - Added code to check for the DB code (check that COB$B_USE_CODE
;    73          0073  1 !            equals COB$K_DBUSE_CODE) before searching for the DB entry.
;    74          0074  1 !            This code used to reside in routine COB$DBEXCEPTION.  LB  16-MAR-81
;    75          0075  1 !    1-016 - Replaced arbitrary signalled values for USE procedure checking
;    76          0076  1 !            code with appropriate symbol names which are now defined in
;    77          0077  1 !            COBMSGDEF.  Added corresponding entries in the EXTERNAL LITERAL
;    78          0078  1 !            declarations for this module.  LB  24-MAR-81
;    79          0079  1 !    1-017 - Changed names of the external literals to correspond to changes
;    80          0080  1 !            made in COBMSG.MDL.  Deleted call to LIB$MATCH_COND and changed
;    81          0081  1 !            the CASE stmt to a SELECTONE stmt.  Changed labels in the
;    82          0082  1 !            SELECTONE stmt (used to be a CASE) to be mnemonics instead of
;    83          0083  1 !            numbers.  Added comments.  LB  16-APR-81
;    84          0084  1 !    1-018 - Deleted the external literals COB$_LSTHNDLOP and LSTHNDLFL and
;    85          0085  1 !            added LSTHNDUSE.  This was done as a result of a change made
;    86          0086  1 !            in COBOL regarding the scoping rules for USE procedures.  Also
;    87          0087  1 !            changed the macro name for the signalling arguments in the signal
;    88          0088  1 !            array to reflect changes made to COBDEF (the reference had been
;    89          0089  1 !            [COB$A_CHK_PROC] which has been changed and extended to the fields
;    90          0090  1 !            [COB$A_OPN_PROC] and [COB$A_FIL_PROC]).  LB  21-APR-81
;    91          0091  1 !    1-019 - Entry point changed to COB$$HANDLER.  For some reason, it had
;    92          0092  1 !            remained a single $ entry point.  Resolves duplicate symbol
;    93          0093  1 !            problem with COBDHANDL.  LB 3-AUG-81
;    94          0094  1 !    1-020 - Added external routine declaration for COB$HANDLER.  LB 4-AUG-81
;    95          0095  1 !
;    96          0096  1 !    1-021 - Added handling of SORT/MERGE signalled errors.  Currently
;    97          0097  1 !            using literal SORT_FAC_CODE until the literal SORT$_FACILITY
;    98          0098  1 !            is put into STARLET.  ER 16-MAR-84
;    99          0099  1 !    1-022 - Move handling of SORT/MERGE errors to end of SELECTONE, and
;   100          0100  1 !            resignal the errors prefixed with the COB$_ERRDURSOR message.
;   101          0101  1 !            Remove unreferenced variables.  Change indentation, and reword the
;   102          0102  1 !            checks that validate the addresses of the USE lists.  Add comments.
;   103          0103  1 !            PDG 9-Apr-84
;   104          0104  1 !--
;   105          0105  1
;   106          0106  1 !<BLF/PAGE>
```

```
;  108          0107  1 !+
;  109          0108  1 ! SWITCHES
;  110          0109  1 !-
;  111          0110  1
;  112          0111  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;  113          0112  1
;  114          0113  1 !+
;  115          0114  1 ! LINKAGES
;  116          0115  1 !       NONE
;  117          0116  1 !-
;  118          0117  1
;  119          0118  1 !+
;  120          0119  1 ! TABLE OF CONTENTS:
;  121          0120  1 !-
;  122          0121  1
;  123          0122  1 FORWARD ROUTINE
;  124          0123  1
;  125          0124  1 ! changed name to cob$$handler
;  126          0125  1         COB$$HANDLER;
;  127          0126  1
;  128          0127  1 !+
;  129          0128  1 ! INCLUDE FILES
;  130          0129  1 !-
;  131          0130  1
;  132          0131  1 REQUIRE 'RTLIN:RTLPSECT';               ! Macros for declaring psects
;  133          0226  1 LIBRARY 'RTLSTARLE';                    ! RTL routines
;  134          0227  1 REQUIRE 'RTLIN:COBDEF';                 ! COBOL specific RTL macros and literals
;  135          0669  1
;  136          0670  1 !+
;  137          0671  1 ! MACROS
;  138          0672  1 !       NONE
;  139          0673  1 !-
;  140          0674  1
;  141          0675  1 !+
;  142          0676  1 ! EQUATED SYMBOLS
;  143          0677  1 !-
;  144          0678  1
;  145          0679  1 LITERAL
;  146          0680  1         CVTTP_OPCODE = %X'26',          ! Opcode value for CVTTP instruction
;  147          0681  1         CVTSP_OPCODE = %X'09';          ! Opcode value for CVTSP instruction
;  148          0682  1
;  149          0683  1 !+
;  150          0684  1 ! PSECT DECLARATIONS:
;  151          0685  1 !-
;  152          0686  1
;  153          0687  1 DECLARE_PSECTS (COB);                   ! Psects for COB$ facility
;  154          0688  1
;  155          0689  1 !+
;  156          0690  1 ! EXTERNAL REFERENCES
;  157          0691  1 !-
;  158          0692  1
;  159          0693  1 EXTERNAL ROUTINE
;  160          0694  1         LIB$STOP,
;  161          0695  1         LIB$SIGNAL,
;  162          0696  1         COB$$INVOKE_USE: NOVALUE,       ! Invoke the USE procedure
;  163          0697  1         COB$HANDLER;
;  164          0698  1
```

```
;  165          0699  1 EXTERNAL LITERAL
;  166          0700  1          COB$_ERRDURSOR,
;  167          0701  1          COB$_NO_USEPRO,          ! No USE procedure available - error on file !AS
;  168          0702  1          COB$_LSTHNDUSE,          ! Lost handler for a USE procedure - environment corrupted !2(+)
;  169          0703  1          COB$_LSTHNDLDB,          ! Lost handler for data base exception - environment corrupted !+
;  170          0704  1          COB$_USE_EXIT,           ! Special signal from COB$$INVOKE_USE
;  171          0705  1          COB$_INVDECDAT;          ! Invalid decimal data signal
;  172          0706  1 LITERAL
;  173          0707  1          SORT$_FACILITY = 28;     ! Delete when SORT$_FACILITY is in STARLET 1-021 ER
```

```
;  175     0708  1 ! changed name to cob$$handler
;  176     0709  1 GLOBAL ROUTINE COB$$HANDLER(SIGNAL,MECHANISM)=
;  177     0710  1
;  178     0711  1
;  179     0712  1 !++
;  180     0713  1 !
;  181     0714  1 ! FUNCTIONAL DESCRIPTION:
;  182     0715  1 !
;  183     0716  1 !       This routine is the error handler for COBOL error conditions.
;  184     0717  1 !       It gets invoked as a result of a call to LIB$SIGNAL.
;  185     0718  1 !
;  186     0719  1 ! CALLING SEQUENCE:
;  187     0720  1 !
;  188     0721  1 !       COB$$HANDLER (signal.rr.r, mechanism.rr.r)
;  189     0722  1 !
;  190     0723  1 ! FORMAL PARAMETERS:
;  191     0724  1 !
;  192     0725  1 !       SIGNAL.rr.r     Address of vector of longwords indicating
;  193     0726  1 !                       nature of condition.
;  194     0727  1 !
;  195     0728  1 !       MECHANISM.rr.r  Address of vector of longwords indicating
;  196     0729  1 !                       the state of the process.
;  197     0730  1 !
;  198     0731  1 !
;  199     0732  1 ! IMPLICIT INPUTS:
;  200     0733  1 !
;  201     0734  1 !       NONE
;  202     0735  1 !
;  203     0736  1 ! IMPLICIT OUTPUTS:
;  204     0737  1 !
;  205     0738  1 !       NONE
;  206     0739  1 !
;  207     0740  1 ! ROUTINE VALUE:
;  208     0741  1 !
;  209     0742  1 !       NONE
;  210     0743  1 !
;  211     0744  1 ! COMPLETION CODES:
;  212     0745  1 !
;  213     0746  1 !       NONE
;  214     0747  1 !
;  215     0748  1 ! SIDE EFFECTS:
;  216     0749  1 !
;  217     0750  1 !       NONE
;  218     0751  1 !
;  219     0752  1 ! NOTES:
;  220     0753  1 !
;  221     0754  1 !       The macro field references beginning with 'CHF' refer to
;  222     0755  1 !       condition handling argument list offsets.
;  223     0756  1 !
;  224     0757  1 !--
```

```
;  226     0758  2     BEGIN
;  227     0759  2
;  228     0760  2     MAP
;  229     0761  2         MECHANISM:      REF BLOCK[,BYTE],
;  230     0762  2         SIGNAL:         REF BLOCK[,BYTE];
;  231     0763  2     LOCAL
;  232     0764  2         CONDITION;      ! Condition that was signalled
;  233     0765  2     LITERAL
;  234     0766  2         FALSE = 0,
;  235     0767  2         TRUE = 1;
;  236     0768  2 !+
;  237     0769  2 ! Find out if it is a signal of interest.
;  238     0770  2 !-
;  239     0771  2
;  240     0772  2
;  241     0773  2     CONDITION = .SIGNAL[CHF$L_SIG_NAME];        ! Fetch condition value from signal array
;  242     0774  2
;  243     0775  2
;  244     0776  2 !+
;  245     0777  2 ! Select appropriate action based on which one we have.
;  246     0778  2 !-
;  247     0779  2
;  248     0780  2     SELECTONE .CONDITION OF
;  249     0781  2         SET
;  250     0782  2
;  251     0783  2         [COB$_USE_EXIT]:        ! Special COBOL unwind signal
;  252     0784  2
;  253     0785  2             $UNWIND();          ! Just unwind
;  254     0786  2
;  255     0787  2         [SS$_ROPRAND]:          ! Was a SS$_ROPRAND
;  256     0788  3             BEGIN
;  257     0789  3
;  258     0790  3             LOCAL
;  259     0791  3                 USER_PC;        ! Program counter where exception took place
;  260     0792  3
;  261     0793  3             !+
;  262     0794  3             ! Using the signal argument vector, extract the
;  263     0795  3             ! program counter at the time the SS$_ROPRAND occurred.
;  264     0796  3             ! The PC is the second-to-last argument in the signal vector.
;  265     0797  3             ! Note that %BPVAL and %UPVAL are pre-declared BLISS literals
;  266     0798  3             ! defining bits per BLISS value and units per BLISS value.
;  267     0799  3             !-
;  268     0800  3
;  269     0801  3             USER_PC = .SIGNAL[ (.SIGNAL[CHF$L_SIG_ARGS]-1)*%UPVAL,0,%BPVAL,0 ];
;  270     0802  3
;  271     0803  3             !+
;  272     0804  3             ! Check to see if a CVTTP or a CVTSP instruction
;  273     0805  3             ! was the generator of the signal.  Note that the
;  274     0806  3             ! PC is pointing to the instruction that caused the fault.
;  275     0807  3             ! If the debugger had a breakpoint set on this instruction,
;  276     0808  3             ! this check will fail, but that's okay.
;  277     0809  3             !-
;  278     0810  3
;  279     0811  4             IF (  .(.USER_PC)<0,8> EQL CVTTP_OPCODE  OR
;  280     0812  4                   .(.USER_PC)<0,8> EQL CVTSP_OPCODE     )
;  281     0813  3             THEN
;  282     0814  3
```

```
;  283          0815  3                !+
;  284          0816  3                ! Set the first longword of the signal argument
;  285          0817  3                ! vector (the condition value field) to the condition
;  286          0818  3                ! name we want the user to see, invalid decimal data.
;  287          0819  3                !-
;  288          0820  3
;  289          0821  3                SIGNAL[CHF$L_SIG_NAME] = COB$_INVDECDAT;
;  290          0822  3            END;
;  291          0823  3
;  292          0824  3        !+
;  293          0825  3        ! The following code handles the case of file specific
;  294          0826  3        ! and open-mode specific GLOBAL USE procedure conditions.
;  295          0827  3        ! Search through the entries for a match.  If there is a match,
;  296          0828  3        ! then invoke the USE procedure and return SS$_CONTINUE;
;  297          0829  3        ! otherwise, re-signal the error (return SS$_RESIGNAL).
;  298          0830  3        !-
;  299          0831  3
;  300          0832  3        [COB$_LSTHNDUSE]:
;  301          0833  3            BEGIN
;  302          0834  3
;  303          0835  3            LOCAL
;  304          0836  3                FP:             REF BLOCK[,BYTE],
;  305          0837  3                SFP:            REF BLOCK[,BYTE];       ! Saved FP
;  306          0838  3            REGISTER
;  307          0839  3                USE = 2:        REF BLOCK[,BYTE],       ! Pointer to USE list
;  308          0840  3                USEENT = 3:     REF BLOCK[,BYTE];       ! Pointer to USE list entry
;  309          0841  3
;  310          0842  3            FP = .MECHANISM[CHF$L_MCH_FRAME];           ! Get FP of this program
;  311          0843  3            SFP = .FP[SF$L_SAVE_FP];                    ! Get FP of program we want to look at
;  312          0844  4            IF  BEGIN
;  313          0845  4                !+
;  314          0846  4                ! This check is to ensure that the only way
;  315          0847  4                ! you could get here is from a COBOL program.
;  316          0848  4                ! Note that we check for COB$HANDLER, rather than COB$$HANDLER;
;  317          0849  4                ! COB$HANDLER is the symbol that COBOL programs reference - it
;  318          0850  4                ! may be in a transfer vector or a fixup section; ie, the frame
;  319          0851  4                ! may not hold a direct reference to COB$$HANDLER.
;  320          0852  4                ! Also, get the USE list.
;  321          0853  4                ! Note that, if we get here, the USE list won't be zero.
;  322          0854  4                !-
;  323          0855  4                IF .SFP EQL 0 THEN FALSE
;  324          0856  4                ELSE IF .SFP[SF$A_HANDLER] NEQA COB$HANDLER THEN FALSE
;  325          0857  4                ELSE IF (USE = .SFP[COB$A_SF_USE]) EQL 0 THEN FALSE
;  326          0858  4                ELSE TRUE
;  327          0859  4                END
;  328          0860  3            THEN
;  329          0861  4                BEGIN
;  330          0862  4
;  331          0863  4                !+
;  332          0864  4                ! Search for a USE procedure declared for the specific file
;  333          0865  4                ! on which the exception occurred.  Note that the
;  334          0866  4                ! COB$A_USE_FILES reference is the base of the 1st file
;  335          0867  4                ! entry and COB$B_GUSE_COUNT is the count of global
;  336          0868  4                ! file entries.
;  337          0869  4                !-
;  338          0870  4
;  339          0871  4                USEENT = USE[COB$A_USE_FILES];  ! Point to first file entry
```

```
;  340     0872  4                DECR I FROM .USE[COB$B_GUSE_COUNT]-1 TO 0 DO
;  341     0873  5                    BEGIN                               ! Loop over files
;  342     0874  5                    IF .USEENT[COB$A_USE_PROC] EQLA .SIGNAL[COB$A_FIL_PROC]
;  343     0875  5                    THEN
;  344     0876  6                        BEGIN
;  345     0877  6                        COB$$INVOKE_USE(                    ! Invoke USE
;  346     0878  6                            .USEENT[COB$A_USE_PROC],        ! Addr of USE procedure
;  347     0879  6                            .USE,                           ! Ptr to USE list
;  348     0880  6                            .FP[SF$L_SAVE_AP],              ! Argument pointer
;  349     0881  6                            .USEENT[COB$A_USE_EOPR],        ! Addr of EOPR block
;  350     0882  6                            .USE[COB$A_USE_PN]);            ! Addr of Perform Nest Ctr
;  351     0883  6                        RETURN SS$_CONTINUE;
;  352     0884  5                        END;
;  353     0885  5                    USEENT = .USEENT + COB$S_USE_FILES; ! Step to next
;  354     0886  4                    END;
;  355     0887  4
;  356     0888  4
;  357     0889  4                !+
;  358     0890  4                ! Open Mode Only.
;  359     0891  4                !
;  360     0892  4                ! See if a procedure has been declared for the
;  361     0893  4                ! open mode.  Note that COB$A_USE_MODES refers to
;  362     0894  4                ! the base of the open mode entries.  There are four
;  363     0895  4                ! open modes, i.e. INPUT, OUTPUT, I-O, and EXTEND.
;  364     0896  4                !-
;  365     0897  4
;  366     0898  4                USEENT = USE[COB$A_USE_MODES];  ! Point to first mode entry
;  367     0899  4                DECR I FROM 3 TO 0 DO
;  368     0900  5                    BEGIN                               ! Loop over modes
;  369     0901  5
;  370     0902  5                    !+
;  371     0903  5                    ! The check here for EOPR not equal to zero is
;  372     0904  5                    ! to ensure that the program is a local one.  If
;  373     0905  5                    ! EOPR equals zero, then the USE procedure is an
;  374     0906  5                    ! up-level reference; then the original condition
;  375     0907  5                    ! that was signalled, should be re-signalled.
;  376     0908  5                    ! Else, if EOPR is not equal to zero and the USE
;  377     0909  5                    ! procedure has been found, then call COB$INV_USE.
;  378     0910  5                    !-
;  379     0911  5
;  380     0912  5                    IF .USEENT[COB$A_USE_PROC] EQLA .SIGNAL[COB$A_OPN_PROC]
;  381     0913  5                        AND .USEENT[COB$A_USE_EOPR] NEQ 0
;  382     0914  5                    THEN
;  383     0915  6                        BEGIN
;  384     0916  6                        COB$$INVOKE_USE(                    ! Invoke USE
;  385     0917  6                            .USEENT[COB$A_USE_PROC],        ! Addr of USE procedure
;  386     0918  6                            .USE,                           ! Ptr to USE list
;  387     0919  6                            .FP[SF$L_SAVE_AP],              ! Argument ptr
;  388     0920  6                            .USEENT[COB$A_USE_EOPR],        ! Addr of EOPR block
;  389     0921  6                            .USE[COB$A_USE_PN]);            ! Addr of Perform Nest Ctr
;  390     0922  6                        RETURN SS$_CONTINUE;
;  391     0923  5                        END;
;  392     0924  5                    USEENT = .USEENT + COB$S_USE_MODES; ! Step to next
;  393     0925  4                    END;
;  394     0926  3                END;
;  395     0927  2            END;
;  396     0928  2
```

```
;  397     0929  2          !+
;  398     0930  2          ! The following code handles the case of a Data Base
;  399     0931  2          ! GLOBAL USE procedure condition.  Search through the COB$GDBUSE_CNT
;  400     0932  2          ! entries and check for a match for the entry address with the
;  401     0933  2          ! address of the USE procedure passed to this handler in the
;  402     0934  2          ! signal argument vector.  If there is a match, then invoke the
;  403     0935  2          ! USE procedure and return SS$_CONTINUE; otherwise, re-signal the
;  404     0936  2          ! error (return SS$_RESIGNAL).
;  405     0937  2          !-
;  406     0938  2
;  407     0939  2          [COB$_LSTHNDLDB]:
;  408     0940  2              BEGIN
;  409     0941  2
;  410     0942  2              LOCAL
;  411     0943  2                  FP:             REF BLOCK[,BYTE],
;  412     0944  2                  SFP:            REF BLOCK[,BYTE];       ! Saved FP
;  413     0945  2              REGISTER
;  414     0946  2                  USE = 2:        REF BLOCK[,BYTE],       ! Ptr to Data Base USE list
;  415     0947  2                  USEENT = 3:     REF BLOCK[,BYTE];       ! Ptr to Data Base USE list entry
;  416     0948  2
;  417     0949  3              FP = .MECHANISM[CHF$L_MCH_FRAME];           ! Get FP of this program
;  418     0950  3              SFP = .FP[SF$L_SAVE_FP];                    ! Get FP of program we want to look at
;  419     0951  4              IF  BEGIN
;  420     0952  4                  !+
;  421     0953  4                  ! This check is to ensure that the only way
;  422     0954  4                  ! you could get here is from a COBOL program.
;  423     0955  4                  ! Also, get the DB USE list.
;  424     0956  4                  !
;  425     0957  4                  ! Check if this is a DB USE list.
;  426     0958  4                  ! The COB$B_USE_CODE field should contain
;  427     0959  4                  ! the generic code for the class of data base
;  428     0960  4                  ! exceptions (equal to COB$K_DBUSE_CODE).
;  429     0961  4                  ! This allows new kinds of USE procedures to be added,
;  430     0962  4                  ! without requiring more longwords on the COBOL stack frame.
;  431     0963  4                  !-
;  432     0964  4                  IF .SFP EQL 0 THEN FALSE
;  433     0965  4                  ELSE IF .SFP[SF$A_HANDLER] NEQA COB$HANDLER THEN FALSE
;  434     0966  4                  ELSE IF (USE = .SFP[COB$A_DB_USE]) EQL 0 THEN FALSE
;  435     0967  4                  ELSE IF .USE[COB$B_USE_CODE] NEQ COB$K_DBUSE_CODE THEN FALSE
;  436     0968  4                  ELSE TRUE
;  437     0969  4                  END
;  438     0970  3              THEN
;  439     0971  4                  BEGIN
;  440     0972  4
;  441     0973  4                  !+
;  442     0974  4                  ! Search for a USE procedure for the corresponding
;  443     0975  4                  ! Data Base exception.  Note that the COB$A_DB_USE
;  444     0976  4                  ! reference is the address of the data base entry while
;  445     0977  4                  ! COB$B_DBUSE_CNT is the count of global Data Base
;  446     0978  4                  ! USE procedures defined in the local program.
;  447     0979  4                  !-
;  448     0980  4
;  449     0981  4                  USEENT = USE[COB$A_DBUSE_ENT];  ! Point to 1st data base entry
;  450     0982  4                  DECR I FROM .USE[COB$B_GDBUSE_CNT] - 1 TO 0 DO
;  451     0983  5                      BEGIN
;  452     0984  5                      IF .USEENT[COB$A_USE_PROC] EQLA .SIGNAL[COB$A_DBCHK_PROC]
;  453     0985  5                      THEN
```

```
;  454          0986  6                    BEGIN
;  455          0987  6                    COB$$INVOKE_USE (                   ! Invoke the USE procedure
;  456          0988  6                        .USEENT[COB$A_USE_PROC],        ! Addr of DB USE procedure
;  457          0989  6                        .USE,                           ! Ptr to DB USE list
;  458          0990  6                        .FP[SF$L_SAVE_AP],              ! Argument pointer
;  459          0991  6                        .USEENT[COB$A_USE_EOPR],        ! Addr of EOPR block
;  460          0992  6                        .USE[COB$A_DBUSE_PNC]);         ! Addr of Perform Nest Ctr
;  461          0993  6                    RETURN SS$_CONTINUE;
;  462          0994  5                    END;
;  463          0995  5                USEENT = .USEENT + COB$S_DBUSE;         ! Step to next entry
;  464          0996  4                END;
;  465          0997  3            END;
;  466          0998  2        END;
;  467          0999  2
;  468          1000  2        !+
;  469          1001  2        ! Check for other errors that are handled specially.
;  470          1002  2        ! Currently, these only include errors from Sort/Merge.
;  471          1003  2        !-
;  472          1004  2
;  473          1005  2        [OTHERWISE]:                                    ! No match occurred
;  474          1006  3            BEGIN
;  475          1007  3            MAP
;  476          1008  3                CONDITION:      BLOCK[,BYTE];           ! Condition that was signalled
;  477          1009  3
;  478          1010  3            !+
;  479          1011  3            ! Is it a SORT/MERGE error signal?
;  480          1012  3            !-
;  481          1013  3            IF  .CONDITION[STS$V_FAC_NO] EQL SORT$_FACILITY
;  482          1014  3            THEN
;  483          1015  4                BEGIN
;  484          1016  4                IF .CONDITION[STS$V_SEVERITY] LSS STS$K_SEVERE
;  485          1017  4                THEN
;  486          1018  5                    BEGIN
;  487          1019  5                    !
;  488          1020  5                    ! These errors are continuable.
;  489          1021  5                    !
;  490          1022  5                    RETURN SS$_CONTINUE         ! Ignore the error
;  491          1023  5                    END
;  492          1024  4                ELSE
;  493          1025  5                    BEGIN
;  494          1026  5                    !
;  495          1027  5                    ! Resignal the error, prefixing the Cobol-specific
;  496          1028  5                    ! error message, and removing the PC and PSL.
;  497          1029  5                    !
;  498          1030  5                    ! Note that, although we don't need to increase the size
;  499          1031  5                    ! of the SIGNAL vector, we can't use it for the new signal,
;  500          1032  5                    ! since we musn't just mung the PC and PSL in this vector.
;  501          1033  5                    !
;  502          1034  5                    ! We assume that ARG_K_SIZE longwords suffice.  If not,
;  503          1035  5                    ! the displayed message will look tacky, that's all.
;  504          1036  5                    !
;  505          1037  5                    LITERAL ARG_K_SIZE = 12;            ! Should be large enough
;  506          1038  5                    LOCAL ARGS: VECTOR[ARG_K_SIZE];
;  507          1039  5                    BUILTIN CALLG;
;  508          1040  5
;  509          1041  5                    ARGS[0] = MINU(.SIGNAL[CHF$L_SIG_ARGS], ARG_K_SIZE-1);
;  510          1042  5                    ARGS[1] = COB$_ERRDURSOR;
```

```
;  511          1043  5                    ARGS[2] = 0;
;  512          1044  5
;  513          1045  5                    CH$MOVE(.ARGS[0] * %UPVAL   ! Everything ...
;  514          1046  5                            - 2*%UPVAL,         ! Less bytes for PC and PSL
;  515          1047  5                        SIGNAL[CHF$L_SIG_NAME],
;  516          1048  5                        ARGS[3]);
;  517          1049  5                    CALLG( ARGS[0], LIB$STOP );
;  518          1050  5
;  519          1051  5                    RETURN SS$_CONTINUE         ! Ignore the original error
;  520          1052  5
;  521          1053  4                    END;
;  522          1054  3                END;
;  523          1055  2            END;
;  524          1056  2
;  525          1057  2        TES;
;  526          1058  2
;  527          1059  2        !+
;  528          1060  2        ! Resignal the error if the signalled condition was not one
;  529          1061  2        ! of the expected conditions to be handled.  Also resignal the
;  530          1062  2        ! error if a USE procedure wasn't found or if the error had been
;  531          1063  2        ! a SS$_ROPRAND since the signal name has been changed.
;  532          1064  2        !-
;  533          1065  2
;  534          1066  2        RETURN SS$_RESIGNAL
;  535          1067  1        END;


                                                .TITLE  COB$$HANDLER
                                                .IDENT  \1-022\

                                                .EXTRN  LIB$STOP, LIB$SIGNAL
                                                .EXTRN  COB$$INVOKE_USE
                                                .EXTRN  COB$HANDLER, COB$_ERRDURSOR
                                                .EXTRN  COB$_NO_USEPRO, COB$_LSTHNDUSE
                                                .EXTRN  COB$_LSTHNDLDB, COB$_USE_EXIT
                                                .EXTRN  COB$_INVDECDAT, SYS$UNWIND

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

                                  00FC 00000    .ENTRY  COB$$HANDLER, Save R2,R3,R4,R5,R6,R7    ; 0709
                 57 00000000G   00  9E 00002    MOVAB   COB$HANDLER, R7                        ;
                 5E             30  C2 00009    SUBL2   #48, SP                                ;
                 54       04    AC  D0 0000C    MOVL    SIGNAL, R4                             ; 0773
                 52       04    A4  D0 00010    MOVL    4(R4), CONDITION                       ;
00000000G        8F            52  D1 00014    CMPL    CONDITION, #COB$_USE_EXIT              ; 0783
                               0B  12 0001B    BNEQ    1$                                     ;
                               7E  7C 0001D    CLRQ    -(SP)                                  ; 0785
00000000G        00            02  FB 0001F    CALLS   #2, SYS$UNWIND                         ;
                               7C  11 00026    BRB     9$                                     ;
00000454         8F            52  D1 00028 1$: CMPL   CONDITION, #1108                       ; 0787
                               1C  12 0002F    BNEQ    3$                                     ;
                 50            64  D0 00031    MOVL    (R4), R0                               ; 0801
                 50       FC A440  D0 00034    MOVL    -4(R4)[R0], USER_PC                    ;
                 26            60  91 00039    CMPB    (USER_PC), #38                         ; 0811
                               05  13 0003C    BEQL    2$                                     ;
                 09            60  91 0003E    CMPB    (USER_PC), #9                          ; 0812
                               61  12 00041    BNEQ    9$                                     ;
```

```
                04   A4 00000000G   8F   D0 00043 2$:     MOVL    #COB$_INVDECDAT, 4(R4)                  ; 0821
                                    57   11 0004B         BRB     9$                                      ; 0780
          00000000G  8F             52   D1 0004D 3$:     CMPL    CONDITION, #COB$_LSTHNDUSE              ; 0832
                                    50   12 00054         BNEQ    10$                                     ;
                     50       08   AC   D0 00056         MOVL    MECHANISM, R0                           ; 0842
                     55       04   A0   D0 0005A         MOVL    4(R0), FP                               ;
                     50       0C   A5   D0 0005E         MOVL    12(FP), SFP                             ; 0843
                                    65   13 00062         BEQL    11$                                     ; 0855
                     51             67   9E 00064         MOVAB   COB$HANDLER, R1                         ; 0856
                     51             60   D1 00067         CMPL    (SFP), R1                               ;
                                    62   12 0006A         BNEQ    12$                                     ;
                     52       FC   A0   D0 0006C         MOVL    -4(SFP), USE                            ; 0857
                                    57   13 00070         BEQL    11$                                     ;
                     53       28   A2   9E 00072         MOVAB   40(R2), USEENT                          ; 0871
                     56       25   A2   9A 00076         MOVZBL  37(USE), I                              ; 0874
                                    09   11 0007A         BRB     5$                                      ;
                0C   A4             63   D1 0007C 4$:     CMPL    (USEENT), 12(R4)                        ;
                                    18   13 00080         BEQL    7$                                      ;
                     53             0C   C0 00082         ADDL2   #12, USEENT                             ; 0885
                     F4             56   F4 00085 5$:     SOBGEQ  I, 4$                                   ; 0872
                     53       04   A2   9E 00088         MOVAB   4(R2), USEENT                           ; 0898
                     56             03   D0 0008C         MOVL    #3, I                                   ; 0912
                10   A4             63   D1 0008F 6$:     CMPL    (USEENT), 16(R4)                        ;
                                    09   12 00093         BNEQ    8$                                      ;
                           04       A3   D5 00095         TSTL    4(USEENT)                               ; 0913
                                    04   13 00098         BEQL    8$                                      ;
                                    62   DD 0009A 7$:     PUSHL   (USE)                                   ; 0921
                                    45   11 0009C         BRB     14$                                     ; 0920
                     53             08   C0 0009E 8$:     ADDL2   #8, USEENT                              ; 0924
                     EB             56   F4 000A1         SOBGEQ  I, 6$                                   ; 0899
                                    56   11 000A4 9$:     BRB     17$                                     ; 0780
          00000000G  8F             52   D1 000A6 10$:    CMPL    CONDITION, #COB$_LSTHNDLDB              ; 0939
                                    4F   12 000AD         BNEQ    18$                                     ;
                     50       08   AC   D0 000AF         MOVL    MECHANISM, R0                           ; 0949
                     55       04   A0   D0 000B3         MOVL    4(R0), FP                               ;
                     50       0C   A5   D0 000B7         MOVL    12(FP), SFP                             ; 0950
                                    3F   13 000BB         BEQL    17$                                     ; 0964
                     51             67   9E 000BD         MOVAB   COB$HANDLER, R1                         ; 0965
                     51             60   D1 000C0         CMPL    (SFP), R1                               ;
                                    7A   12 000C3         BNEQ    21$                                     ;
                     52       F8   A0   D0 000C5         MOVL    -8(SFP), USE                            ; 0966
                                    74   13 000C9 11$:     BEQL    21$                                     ;
                     01             62   91 000CB         CMPB    (USE), #1                               ; 0967
                                    6F   12 000CE 12$:     BNEQ    21$                                     ;
                     53       0C   A2   9E 000D0         MOVAB   12(R2), USEENT                          ; 0981
                     56       09   A2   9A 000D4         MOVZBL  9(USE), I                               ; 0984
                                    1F   11 000D8         BRB     16$                                     ;
                0C   A4             63   D1 000DA 13$:     CMPL    (USEENT), 12(R4)                        ;
                                    16   12 000DE         BNEQ    15$                                     ;
                           04       A2   DD 000E0         PUSHL   4(USE)                                  ; 0992
                           04       A3   DD 000E3 14$:     PUSHL   4(USEENT)                               ; 0991
                           08       A5   DD 000E6         PUSHL   8(FP)                                   ; 0990
                                    52   DD 000E9         PUSHL   USE                                     ; 0989
                                    63   DD 000EB         PUSHL   (USEENT)                                ; 0988
          00000000G  00             05   FB 000ED         CALLS   #5, COB$$INVOKE_USE                     ;
                                    45   11 000F4         BRB     20$                                     ; 0993
                     53             0C   C0 000F6 15$:     ADDL2   #12, USEENT                             ; 0995
```

```
                                DE              56  F4 000F9 16$:   SOBGEQ  I, 13$                                  ; 0982
                                                41  11 000FC 17$:   BRB     21$                                     ; 0780
          1C            52      0C              10  ED 000FE 18$:   CMPZV   #16, #12, CONDITION, #28                ; 1013
                                                3A  12 00103        BNEQ    21$                                     ;
          04            52      03              00  ED 00105        CMPZV   #0, #3, CONDITION, #4                   ; 1016
                                                2F  19 0010A        BLSS    20$                                     ;
                                50              64  D0 0010C        MOVL    (R4), R0                                ; 1041
                                0B              50  D1 0010F        CMPL    R0, #11                                 ;
                                                03  1B 00112        BLEQU   19$                                     ;
                                50              0B  D0 00114        MOVL    #11, R0                                 ;
                                6E              50  D0 00117 19$:   MOVL    R0, ARGS                                ;
                          04    AE 00000000G    8F  D0 0011A        MOVL    #COB$_ERRDURSCR, ARGS+4                 ; 1042
                                      08        AE  D4 00122        CLRL    ARGS+8                                  ; 1043
                                50              6E  D0 00125        MOVL    ARGS, R0                                ; 1045
                                50              04  C4 00128        MULL2   #4, R0                                  ; 1046
                                50              08  C2 0012B        SUBL2   #8, R0                                  ;
                    0C    AE        04          A4  50 28 0012E     MOVC3   R0, 4(R4), ARGS+12                      ; 1048
                          00000000G             00  6E FA 00134     CALLG   ARGS, LIB$STOP                          ; 1049
                                50              01  D0 0013B 20$:   MOVL    #1, R0                                  ; 1051
                                                04  0013E         RET                                             ; 1025
                                50      0918    8F  3C 0013F 21$:   MOVZWL  #2328, R0                               ; 1066
                                                04  00144         RET                                             ; 1067

; Routine Size:  325 bytes,    Routine Base:  _COB$CODE + 0000

;  536          1068  1
;  537          1069  1 END
;  538          1070  0 ELUDOM


;                                   PSECT SUMMARY

;      Name                     Bytes                          Attributes

;  _COB$CODE                     325  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         14         0          581        00:00.7
```

```
;                                       COMMAND QUALIFIERS

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

; Size:          325 code + 0 data bytes
; Run Time:        00:08.3
; Elapsed Time:    00:36.4
; Lines/CPU Min:    7772
; Lexemes/CPU-Min: 28895
; Memory Used:  155 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

COBPAUSE
LIS

COBMSG
LIS

COBHANDLE
LIS

COBINTAR
LIS

COBINTER
LIS

COBMULQ
LIS

COBPOSERA
LIS

COBIOEXCE
LIS

COBLINAGE
LIS

COBKEY
LIS

COBINVUSE
LIS