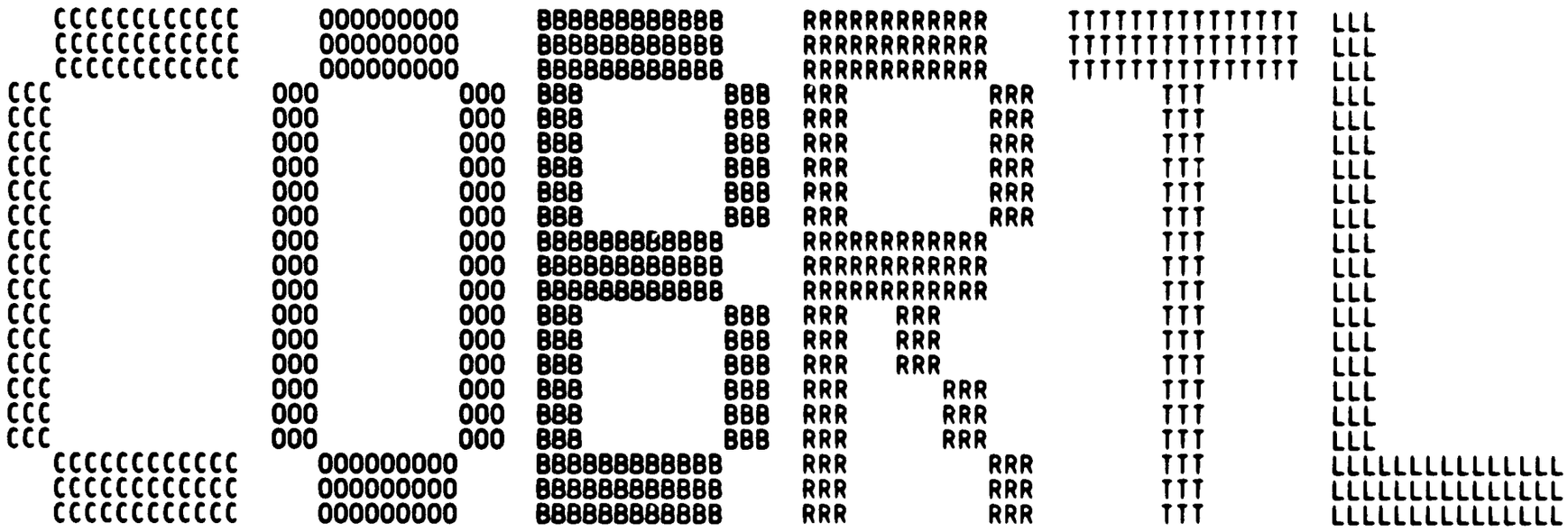
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR 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 BBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBBBBBBBBBBB RRRRRRRRRRR 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
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL

```
  CCCCCCCC    000000    BBBBBBBB      IIIIII      000000    EEEEEEEEEE  XX      XX    CCCCCCCC  EEEEEEEEEE
  CCCCCCCC    000000    BBBBBBBB      IIIIII      000000    EEEEEEEEEE  XX      XX    CCCCCCCC  EEEEEEEEEE
CC           00      00  BB      BB      II      00      00  EE          XX      XX  CC          EE
CC           00      00  BB      BB      II      00      00  EE          XX      XX  CC          EE
CC           00      00  BB      BB      II      00      00  EE            XX  XX    CC          EE
CC           00      00  BB      BB      II      00      00  EE            XX  XX    CC          EE
CC           00      00  BBBBBBBB        II      00      00  EEEEEEEE        XX      CC          EEEEEEEE
CC           00      00  BBBBBBBB        II      00      00  EEEEEEEE        XX      CC          EEEEEEEE
CC           00      00  BB      BB      II      00      00  EE            XX  XX    CC          EE
CC           00      00  BB      BB      II      00      00  EE            XX  XX    CC          EE
CC           00      00  BB      BB      II      00      00  EE          XX      XX  CC          EE
CC           00      00  BB      BB      II      00      00  EE          XX      XX  CC          EE
  CCCCCCCC    000000    BBBBBBBB      IIIIII      000000    EEEEEEEEEE  XX      XX    CCCCCCCC  EEEEEEEEEE
  CCCCCCCC    000000    BBBBBBBB      IIIIII      000000    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
```

```
;     1          0001  0 MODULE COB$IOEXCEPTION(
;     2          0002  0                   IDENT = '1-039'          ! file:COBIOEXCE.B32  EDIT:BH1039
;     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 ! FACILITY:     COBOL SUPPORT
;    31          0031  1 !
;    32          0032  1 ! ABSTRACT:
;    33          0033  1 !
;    34          0034  1 !       This procedure is called to process a wide variety of I/O
;    35          0035  1 ! exceptions.  Depending on the nature of the exception, it sets the
;    36          0036  1 ! file status variable, causes USE procedures to be invoked and
;    37          0037  1 ! signals errors.
;    38          0038  1 !
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:  Vax-11 User Mode
;    41          0041  1 !
;    42          0042  1 ! AUTHOR: MLJ , CREATION DATE: 02-MAY-1979
;    43          0043  1 !
;    44          0044  1 ! MODIFIED BY:
;    45          0045  1 !
;    46          0046  1 ! 1-001 - Original. MLJ  02-MAY-1979
;    47          0047  1 ! 1-002 - Added boilerplate and comments. RKR 22-AUG-1979
;    48          0048  1 ! 1-003 - Redesigned control tables to allow additional checking on
;    49          0049  1 !         status of state variables -- option file errors and repeated
;    50          0050  1 !         EOF error mechanisms.  RKR 10-SEPT-1979
;    51          0051  1 ! 1-004 - Added additional statuses (RMS$_FAB and RMS$_RAB) to control
;    52          0052  1 !         tables and delete explicit checks for "FNO" flag.
;    53          0053  1 !         RKR 12-SEPT-79
;    54          0054  1 ! 1-005 - Complete revamp of action codes in control table RKR 13-SEPT-79
;    55          0055  1 ! 1-006 - Added FAC error codes to READ S S and READ R S control tables
;    56          0056  1 !         RKR 18-SEPT-79
;    57          0057  1 ! 1-007 - Fetch COBDEF from RTLIN rather than LIB$.  RKR 18-SEPT-79
```

```
;   58          0058  1 ! 1-008 - Add COB$K_EXC_CLORS to SET.  RKR 18-SEP-79
;   59          0059  1 ! 1-009 - Fix endcoding and decoding of error messages. RKR 19-SEPT-79
;   60          0060  1 !
;   61          0061  1 ! 1-010 - Add logic to discriminate between COB$_REASMMIN, COB$_WRISMAMIN, and COB$_REWSMAMIN.
;   62          0062  1 !         RKR 25-SEPT-79
;   63          0063  1 ! 1-011 - Added diagnostic for unexpected situation in which a RMS
;   64          0064  1 !         success code is encountered.  RKR 26-SEPT-79
;   65          0065  1 ! 1-012 - Change symbolic name of LIBRARY file. RKR 1-OCT-79
;   66          0066  1 ! 1-013 - Make it use COB$$INVOKE_USE instead of COB$INVOKE_USE
;   67          0067  1 !         RKR 1-OCT-79
;   68          0068  1 ! 1-014 - Stop outputting the COB$_NO_USEPRO error message since it
;   69          0069  1 !         contains no information which is not also contained in
;   70          0070  1 !         the subsequent more specific message.
;   71          0071  1 !         Add entry to CLOSE_T to try to catch a CLOSE REEL operation
;   72          0072  1 !         to an inappropriate device.  If operation is close and error
;   73          0073  1 !         status is RMS$_IOP, SUCCEED with FILESTAT of '00'.
;   74          0074  1 !         RKR 1-OCT-79
;   75          0075  1 ! 1-015 - Add special case for COB$K_EXC_ORG.  RKR 04-OCT-79
;   76          0076  1 ! 1-016 - Use max and mins from REQUIRE file, add new table and new
;   77          0077  1 !         state for CLOSE REEL, cosmetic changes, bugchecks at
;   78          0078  1 !         empty [INRANGE] statements.
;   79          0079  1 !         RKR 20-OCT-79
;   80          0080  1 ! 1-017 - Cosmetic changes.  21-OCT-79
;   81          0081  1 ! 1-018 - Rearrange statuses for tables CLOSE_T and CLOSEREEL_S_S_T.
;   82          0082  1 !         RKR 29-OCT-79
;   83          0083  1 ! 1-019 - Add comments.  RKR 05-NOV-79
;   84          0084  1 ! 1-020 - Change resulting file status and action codes for
;   85          0085  1 !         RMS$_OK_RLK for read actions.  RKR 08-NOV-79
;   86          0086  1 ! 1-021 - Change some table entries dealing with creating of
;   87          0087  1 !         duplicate keys.  RKR 11-NOV-79
;   88          0088  1 ! 1-022 - Added comments and made cosmetic changes.  LB 3-MAR-81
;   89          0089  1 ! 1-023 - Changed code that checked the value of FP; if zero, a call
;   90          0090  1 !         is now made to SIGNAL_STOP indicating a fatal internal error.
;   91          0091  1 !         Also, code was taken out that checked if the current handler
;   92          0092  1 !         was COB$HANDLER; this is not a catch-all check since a user
;   93          0093  1 !         can establish his/her own error handler.  Also, added more
;   94          0094  1 !         comments.  LB  05-MAR-81
;   95          0095  1 ! 1-024 - Replaced arbitrary signalling values for USE procedure code
;   96          0096  1 !         with appropriate symbol names now defined in COBMSGDEF.
;   97          0097  1 !         Added corresponding entries in the EXTERNAL LITERAL
;   98          0098  1 !         declarations for this module.  LB  24-MAR-81
;   99          0099  1 ! 1-025 - Changed external literal for no use procedure on open mode entry
;  100          0100  1 !         and added an external literal for lost handler for file specific
;  101          0101  1 !         entry to correspond with changes made to COBMSG.MDL.  Changed
;  102          0102  1 !         calls to LIB$SIGNAL to use SIGNAL for consistency reasons.
;  103          0103  1 !         Optimized code by encasing calls to SIGNAL_STOP in BEGIN-END
;  104          0104  1 !         blocks.  Changed the calls to SIGNAL to now take an FAO
;  105          0105  1 !         parameter (as is syntactically correct) even though the
;  106          0106  1 !         parameter that is getting passed to the handler is not an
;  107          0107  1 !         FAO parameter (note that the !+ directive in the error message
;  108          0108  1 !         text will ignore it).  LB  16-APR-81
;  109          0109  1 ! 1-026 - Deleted the external literals COB$_LSTHNDLOP and COB$_LSTHNDLFL
;  110          0110  1 !         and added COB$_LSTHNDUSE.  This was done as a result of a change
;  111          0111  1 !         made in COBOL regarding the scoping rules for USE procedures.
;  112          0112  1 !         Changed code in the area of loading up the resultant string
;  113          0113  1 !         descriptor to facilitate the added functionality of RMS special
;  114          0114  1 !         registers within COBOL.  The code now loads the actual string from
```

```
;  115          0115  1 !          the FAB into the resultant string area within the NAM block in the
;  116          0116  1 !          case where the name block string length and extended string length
;  117          0117  1 !          equal zero.  Changed code that searched for USE procedures.  A
;  118          0118  1 !          single condition is now signalled in the case where there are
;  119          0119  1 !          applicable USE procedures (but cannot be seen within this level of
;  120          0120  1 !          code) and an additional parameter is included in the signal (it now
;  121          0121  1 !          takes the entry point of a file specific USE procedure or 0 if none
;  122          0122  1 !          as well as the entry point of an open mode specific USE procedure
;  123          0123  1 !          or 0 if none).  Also, the final action code was removed from each of
;  124          0124  1 !          the separate blocks of search code for USE procedures and moved to a
;  125          0125  1 !          single place outside of those blocks of code.  LB  21-APR-81.
;  126          0126  1 !  1-027 - Added code in success status checkout code for reads, which will
;  127          0127  1 !          take care of an unreported bug that exists in both the V1.0 and
;  128          0128  1 !          V2.0 versions of the COBOL compiler.  The code now further checks
;  129          0129  1 !          if the RMS_STS code indicates a file that (soft) record locked.
;  130          0130  1 !          If so, it sets the FILESTAT to 90 (instead of 0) and sets the
;  131          0131  1 !          ACTION to CONTINUE (instead of SUCCEED).  Also caused the read
;  132          0132  1 !          entries in THE table for RMS$_OK_RLK to be removed.  LB  30-APR-81.
;  133          0133  1 !  1-028 - Added code following the call to COB$$INVOKE_USE to check the
;  134          0134  1 !          ACTION code.  This code was added as a result of the current code
;  135          0135  1 !          falling through into code that would signal an error.  LB  14-MAY-81
;  136          0136  1 !  1-029 - Added a check before calling SIGNAL for ensuring that OPEN_MD_ADDR
;  137          0137  1 !          and FILE_ADDR are not equal to zero (making sure that USE procedures
;  138          0138  1 !          do indeed exist before signalling).  LB  29-JUN-81.
;  139          0139  1 !  1-030 - Added COB$_KEYNOTMAT.  Also, moved setting of ERR_MSG_NUM slightly to
;  140          0140  1 !          allow BLISS to do more omega motion.  Use PLIT_TABLE, and remove case
;  141          0141  1 !          statement to access correct action table.  Rename PLITs consistently.
;  142          0142  1 !          Use common action tables.  PDG 24-JUL-81
;  143          0143  1 !  1-031 - Added support for manual record locking.
;  144          0144  1 !                  Error messages: COB$K_RECNOTLOC and COB$K_UNLUNOFIL
;  145          0145  1 !                  UNL_T table AND modification to PLIT_TABLE bh 28-jul-81
;  146          0146  1 !  1-032 - Added code to move the COBOL condition value to RAB[RMS$_STS/STV]
;  147          0147  1 !          if the error is a non-RMS error (alternately, an error that is
;  148          0148  1 !          detected by compiled code).  This allows the RMS-STS/STV values
;  149          0149  1 !          to be meaningful within a USE procedure.  PDG  7-AUG-81.
;  150          0150  1 !  1-033 - Corrected the action code associated with RECNOTLOC to be SUCCEED,
;  151          0151  1 !          and changed the file status to "00".
;  152          0152  1 !          PDG 18-Aug-1981.
;  153          0153  1 !  1-034 - Added support for UNLOCK in non-automatic record locking mode.
;  154          0154  1 !          BH 29-Oct-1981.
;  155          0155  1 !  1-035 - Added LIB$STOP as external routine.  LB 30-NOV-81
;  156          0156  1 !  1-036 - Changed UNL_T table entries for the RMS ISI and RAB codes from
;  157          0157  1 !          GOTO to ABORT.  LB 1-DEC-81
;  158          0158  1 !  1-037 - Changed REA_R_R, DEL_R_R, REW_R_R, STA_R_S table entries to include
;  159          0159  1 !          RMS$_MRN. Error is treated the same as RMS$_RNF. Did NOT Remove
;  160          0160  1 !          equates of Indexed and Relative table for REA_?_R, DEL_?_R, REW_?_R,
;  161          0161  1 !          STA_?_S because MRN is not applicable to indexed files and the condition
;  162          0162  1 !          should not occur; so we save table size.  bh 10-AUG-82
;  163          0163  1 !  1-038 - Add ANSI74 argument and some ANSI74 code (change file status 13, 15,
;  164          0164  1 !          and 16 to file status 10).  Without the ANSI74 argument there is no
;  165          0165  1 !          change to the file status values (13, 15, and 16).  This allows
;  166          0166  1 !          VAX-11 COBOL Version 2 and Version 3 compatibility.  With the ANSI74
;  167          0167  1 !          argument file status values 13, 15, and 16 are all changed to 10.
;  168          0168  1 !          This allows VAX-11 COBOL to pass Validation with no file status
;  169          0169  1 !          errors.    CR  5-OCT-82
;  170          0170  1 !  1-039 - Added error handling for previous IO was NOT successful READ for
;  171          0171  1 !          DELETE and START statements.  Added error COB$REAMP_D_R.  Updated
```

```
;  172          0172  1 !        COB$K_EXC_MAXM to 7 (also comments).  Added COB$K_EXC_PIO code.
;  173          0173  1 !                                                                BH 1-SEP-83
;  174          0174  1 !--
;  175          0175  1
;  176          0176  1 !<BLF/PAGE>
```

```
;  178          0177  1 !+
;  179          0178  1 ! SWITCHES
;  180          0179  1 !-
;  181          0180  1
;  182          0181  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;  183          0182  1
;  184          0183  1 !+
;  185          0184  1 ! LINKAGES
;  186          0185  1 !       NONE
;  187          0186  1 !-
;  188          0187  1
;  189          0188  1 !++
;  190          0189  1 ! TABLE OF CONTENTS:
;  191          0190  1 !-
;  192          0191  1
;  193          0192  1 FORWARD ROUTINE
;  194          0193  1         COB$IOEXCEPTION ;
;  195          0194  1
;  196          0195  1 !+
;  197          0196  1 ! INCLUDE FILES
;  198          0197  1 !-
;  199          0198  1
;  200          0199  1 REQUIRE 'RTLIN:COBDEF';                 ! COBOL specific RTL macros and literals
;  201          0641  1 REQUIRE 'RTLIN:RTLPSECT' ;              ! Macros for defining psects
;  202          0736  1 LIBRARY 'RTLSTARLE';                    ! RTL routines
;  203          0737  1
;  204          0738  1 !+
;  205          0739  1 ! MACROS
;  206          0740  1 !-
;  207          0741  1
;  208          0742  1 MACRO
;  209          0743  1
;  210          0744  1         !+
;  211          0745  1         ! Field definitions for table.
;  212          0746  1         !-
;  213          0747  1
;  214          0748  1         TAB_STATUS=     0,0,32,0 %,     ! RMS status code (or 0)
;  215          0749  1         TAB_FILESTAT=   4,0,16,0 %,     ! COBOL file status
;  216          0750  1         TAB_ACTION=     6,0,8,0 %,      ! Type of recovery action
;  217          0751  1         TAB_ERR_NO=     7,0,8,0 %,      ! COB$ facility error number
;  218          0752  1         TAB_TST_STATE=  8,0,8,0 %,      ! Optional state to test
;  219          0753  1         TAB_SET_STATE=  9,0,8,0 %,      ! Optional state to set
;  220          0754  1
;  221          0755  1         !+
;  222          0756  1         ! Macro to make table entries.
;  223          0757  1         !-
;  224          0758  1
;  225        M 0759  1         Z[A,B,C,D,E,F]=
;  226          0760  1          LONG(A), WORD(B), BYTE(C), BYTE((D-COB$_NO_USEPRO)^-3), BYTE(E), BYTE(F) %;
;  227          0761  1
;  228          0762  1 !+
;  229          0763  1 ! EQUATED SYMBOLS
;  230          0764  1 !-
;  231          0765  1
;  232          0766  1 LITERAL
;  233          0767  1
;  234          0768  1         MIN_STATE = 0,                  ! Minimum state variable
```

```
;  235     0769  1
;  236     0770  1        NULL = 0 ,                      ! No special action
;  237     0771  1        OPTF = 1 ,                      ! Optional file
;  238     0772  1        OFNP = 2 ,                      ! Optional file not present (OPEN couldn't find)
;  239     0773  1        NNVR = 3 ,                      ! No next valid record (already seen EOF)
;  240     0774  1
;  241     0775  1        MAX_STATE = 3,                  ! Maximum state variable
;  242     0776  1
;  243     0777  1
;  244     0778  1        !+
;  245     0779  1        ! Size of table entry.
;  246     0780  1        !-
;  247     0781  1
;  248     0782  1        TAB_S_ENTRY=    10,
;  249     0783  1
;  250     0784  1        !+
;  251     0785  1        ! Values of ACTION.             Resulting behavior
;  252     0786  1        !-
;  253     0787  1
;  254     0788  1        ABORT=          0,
;  255     0789  1
;  256     0790  1                        !+
;  257     0791  1                        ! If an applicable USE procedure is present,
;  258     0792  1                        ! then invoke the USE procedure followed by
;  259     0793  1                        ! continuing at the "next statement" (i.e.
;  260     0794  1                        ! RETURN 0;
;  261     0795  1                        ! Otherwise, LIBSTOP with appropriate error
;  262     0796  1                        ! signal.
;  263     0797  1                        !
;  264     0798  1        GOTO=           1,
;  265     0799  1
;  266     0800  1                        ! If an exception label is present, goto
;  267     0801  1                        ! exception label;
;  268     0802  1                        ! Otherwise handle like class ABORT.
;  269     0803  1                        !
;  270     0804  1        CONTINUE=       2,
;  271     0805  1
;  272     0806  1                        ! If an applicable USE procedure is present,
;  273     0807  1                        ! then invoke USE procedure followed by
;  274     0808  1                        ! continuing "immediately" (i.e. RETURN 1);
;  275     0809  1                        ! Otherwise, continue "immediately"
;  276     0810  1                        ! (i.e. RETURN 1)
;  277     0811  1                        !
;  278     0812  1        SUCCEED=        3;
;  279     0813  1
;  280     0814  1                        ! The COBOL I/O statement is successful in
;  281     0815  1                        ! every respect.  Therefore, continue
;  282     0816  1                        ! "immediately" (i.e. RETURN 1) after updating
;  283     0817  1                        ! file status variable (if present).
;  284     0818  1                        !-
```

```
;  286          0819  1 !+
;  287          0820  1 ! PSECT DECLARATIONS:
;  288          0821  1 !-
;  289          0822  1
;  290          0823  1 DECLARE_PSECTS (COB) ;                          ! Declare psects for COB$ facility
;  291          0824  1
;  292          0825  1 !+
;  293          0826  1 ! EXTERNAL REFERENCES
;  294          0827  1 !-
;  295          0828  1
;  296          0829  1 EXTERNAL ROUTINE
;  297          0830  1         LIB$STOP:NOVALUE,                      ! Signal fatal error
;  298          0831  1         COB$HANDLER:    ,                      ! COBOL exception handler
;  299          0832  1         COB$$INVOKE_USE: NOVALUE;              ! Invoke USE procedure
;  300          0833  1
;  301          0834  1 EXTERNAL LITERAL
;  302          0835  1     COB$_NO_USEPRO,    ! No USE procedure available for error on file !AS
;  303          0836  1     COB$_ERRON_FIL,    ! Error on file !AS
;  304          0837  1     COB$_OPTMISOPE,    ! Optional file !AS missing on OPEN
;  305          0838  1     COB$_FILALRLOC,    ! File !AS is already locked
;  306          0839  1     COB$_FILALROPE,    ! File !AS is already open
;  307          0840  1     COB$_FILCLOLOC,    ! File !AS is closed with LOCK
;  308          0841  1     COB$_NO_SPACE,     ! No filespace on device for file !AS
;  309          0842  1     COB$_FILNOTFOU,    ! File !AS not found on OPEN
;  310          0843  1     COB$_OPTMISCLO,    ! Optional file !AS missing on CLOSE
;  311          0844  1     COB$_FILALRCLO,    ! File !AS already closed
;  312          0845  1     COB$_NO_NEXLOG,    ! No next logical record on file !AS
;  313          0846  1     COB$_OPTMISREA,    ! Optional file !AS missing on READ
;  314          0847  1     COB$_NO_NEXVAL,    ! No next valid record on file !AS
;  315          0848  1     COB$_RECLOCREA,    ! Record on file !AS is already locked (READ attempt)
;  316          0849  1     COB$_RECLOC_OK,    ! Record on file !AS already locked, but ok
;  317          0850  1     COB$_REAUNOFIL,    ! Attempting READ on unopened file !AS
;  318          0851  1     COB$_REAINCOPE,    ! READ on file !AS incompatible with OPEN mode
;  319          0852  1     COB$_WRIBEYBOU,    ! Attempting WRITE beyond boundaries of file !AS
;  320          0853  1     COB$_WRIUNOFIL,    ! Attempting WRITE on unopened file !AS
;  321          0854  1     COB$_WRIINCOPE,    ! WRITE on file !AS incompatible with OPEN mode
;  322          0855  1     COB$_REWNO_R_S,    ! Attempting REWRITE on file !AS with no previous READ or START
;  323          0856  1     COB$_REWUNOFIL,    ! Attempting REWRITE on unopened file !AS
;  324          0857  1     COB$_REWINCOPE,    ! REWRITE on file !AS incompatible with OPEN mode
;  325          0858  1     COB$_RECNOTEXI,    ! Record does not exist on file !AS
;  326          0859  1     COB$_OPTMISSTA,    ! Optional file !AS missing on START
;  327          0860  1     COB$_RECLOCSTA,    ! Record on file !AS is already locked (START attempt)
;  328          0861  1     COB$_STAUNOFIL,    ! Attempting START on unopened file !AS
;  329          0862  1     COB$_STAINCOPE,    ! START on file !AS incompatible with OPEN mode
;  330          0863  1     COB$_RECLOCWRI,    ! Record on file !AS is already locked (WRITE attempt)
;  331          0864  1     COB$_RECLOCDEL,    ! Record on file !AS is already locked (DELETE attempt)
;  332          0865  1     COB$_DELNO_R_S,    ! Attempting DELETE on file !AS without previous READ or START
;  333          0866  1     COB$_DELUNOFIL,    ! Attempting DELETE on unopened file !AS
;  334          0867  1     COB$_DELINCOPE,    ! DELETE on file !AS incompatible with OPEN mode
;  335          0868  1     COB$_RECLOCREW,    ! Record on file !AS is already locked (REWRITE attempt)
;  336          0869  1     COB$_WRIDUPKEY,    ! Attempting WRITE of duplicate key in file !AS
;  337          0870  1     COB$_WRICREDUP,    ! WRITE created an allowed duplicate alternate key on file !AS
;  338          0871  1     COB$_WRINOTASC,    ! Attempting to WRITE non-ascending ISAM key on file !AS
;  339          0872  1     COB$_WRIDUPALT,    ! Attempting to WRITE duplicate alternate key on file !AS
;  340          0873  1     COB$_REWCREDUP,    ! REWRITE created an allowed duplicate alternate key on file !AS
;  341          0874  1     COB$_PRIKEYCHA,    ! Primary record key on file !AS changed between READ and REWRITE
;  342          0875  1     COB$_REWDISDUP,    ! Attempting to REWRITE disallowed duplicate key on file !AS
```

```
;  343          0876  1      COB$_WRIDISDUP,    ! Attempting to WRITE disallowed duplicate key on file !AS
;  344          0877  1      COB$_REASMAMIN,    ! Attempting READ of variable length smaller than minimum allowed from file !AS
;  345          0878  1      COB$_WRISMAMIN,    ! Attempting WRITE of variable length smaller than minimum allowed to file !AS
;  346          0879  1      COB$_REWSMAMIN,    ! Attempting REWRITE of variable length smaller than minimum allowed to file !AS
;  347          0880  1      COB$_ORGNOTMAT,    ! Attempting to open file whose organization does
;  348          0881  1                         ! not match access mode
;  349          0882  1      COB$_INVARG,       ! Invalid arguments
;  350          0883  1      COB$_LSTHNDUSE,    ! Lost handler for a USE procedure - environment corrupted !2(+)
;  351          0884  1      COB$_KEYNOTMAT,    ! Attempting to open indexed file with keys whose description
;  352          0885  1                         ! does not match those expected
;  353          0886  1      COB$_RECNOTLOC,    ! Record not locked in file !AS (UNLOCK attempt)
;  354          0887  1      COB$_UNLUNOFIL,    ! Attempting UNLOCK on unopened file !AS
;  355          0888  1      COB$_UNLNO_CUR,    ! Attempting UNLOCK on file !AS with no current record
;  356          0889  1      COB$_REAMP_D_R,    ! READ must precede DELETE or REWRITE in sequential access mode
;  357          0890  1      OTS$_FATINTERR;    ! Condition value
;  358          0891  1
;  359          0892  1
```

```
;  361          0893  1 !+
;  362          0894  1 ! The following table contains the logic which drives a large portion
;  363          0895  1 ! of this program.  During a table look-up, a match is found first on
;  364          0896  1 ! the RMS status code and on the State-to-test field.  When a match is
;  365          0897  1 ! found, the State-to-set is set if non-zero, the file status becomes
;  366          0898  1 ! the file status returned to the users file status variable, Action
;  367          0899  1 ! code controls the flow through the code and the error message is
;  368          0900  1 ! the one signalled if the user has not provided a USE procedure to
;  369          0901  1 ! deal with the problem.
;  370          0902  1 !
;  371          0903  1 ! Each table must end with an entry that has a 0 for the RMS status
;  372          0904  1 ! code.  This forms a default termination for the table lookup.
;  373          0905  1 !
;  374          0906  1 ! If the action code is SUCCEED, the Associated Error Message in the
;  375          0907  1 ! table is academic since it is never signaled.
;  376          0908  1 !
;  377          0909  1 ! The following tables contain information as follows:
;  378          0910  1 !
;  379          0911  1 !       RMS status      File    ACTION   Associated    State   State
;  380          0912  1 !       code            Status  code     Error msg    to test  to set
;  381          0913  1 !       ----------      ------  -------  ----------   -------  ------
;  382          0914  1 !
;  383          0915  1 !-
;  384          0916  1
;  385          0917  1
;  386          0918  1 BIND
;  387          0919  1         BASE = UPLIT(REP 0 OF (0)),
;  388        P 0920  1         OPE_T = UPLIT BYTE(Z(
;  389        P 0921  1             RMS$_FUL,   '95',   ABORT,   COB$_NO_SPACE,  0,    0,
;  390        P 0922  1             RMS$_FNF,   '05',   CONTINUE,COB$_OPTMISOPE, OPTF, OFNP,
;  391        P 0923  1             RMS$_FNF,   '97',   ABORT,   COB$_FILNOTFOU, 0,    0,
;  392        P 0924  1             RMS$_FLK,   '91',   ABORT,   COB$_FILALRLOC, 0,    0,
;  393          0925  1             0,          '30',   ABORT,   COB$_ERRON_FIL, 0,    0)),
;  394          0926  1
;  395        P 0927  1         CLO_T = UPLIT BYTE(Z(
;  396        P 0928  1             RMS$_IFI,   '00',   SUCCEED, COB$_OPTMISCLO,  OFNP, 0,
;  397        P 0929  1             RMS$_IFI,   '94',   ABORT,   COB$_FILALRCLO,  0,    0,
;  398        P 0930  1             RMS$_FAB,   '94',   ABORT,   COB$_FILALRCLO,  0,    0,
;  399          0931  1             0,          '98',   ABORT,   COB$_ERRON_FIL,  0,    0)),
;  400          0932  1
;  401        P 0933  1         CLR_S_S_T = UPLIT BYTE(Z(
;  402        P 0934  1             RMS$_ISI,   '00',   SUCCEED, COB$_OPTMISCLO,  OFNP, 0,
;  403        P 0935  1             RMS$_ISI,   '94',   ABORT,   COB$_FILALRCLO,  0,    0,
;  404        P 0936  1             RMS$_RAB,   '94',   ABORT,   COB$_FILALRCLO,  0,    0,
;  405        P 0937  1             RMS$_IOP,   '00',   SUCCEED, COB$_NO_USEPRO,  0,    0,
;  406        P 0938  1             RMS$_EOF,   '00',   SUCCEED, COB$_NO_USEPRO,  0,    0,
;  407          0939  1             0,          '98',   ABORT,   COB$_ERRON_FIL,  0,    0)),
;  408          0940  1
;  409        P 0941  1         REA_S_S_T = UPLIT BYTE(Z(
;  410        P 0942  1             RMS$_EOF,   '16',   GOTO,    COB$_NO_NEXVAL,  NNVR, 0,
;  411        P 0943  1             RMS$_EOF,   '13',   GOTO,    COB$_NO_NEXLOG,  0,    NNVR,
;  412        P 0944  1             RMS$_RLK,   '92',   ABORT,   COB$_RECLOCREA,  0,    0,
;  413        P 0945  1             RMS$_ISI,   '15',   GOTO,    COB$_OPTMISREA,  OFNP, 0,
;  414        P 0946  1             RMS$_ISI,   '94',   ABORT,   COB$_REAUNOFIL,  0,    0,
;  415        P 0947  1             RMS$_RAB,   '94',   ABORT,   COB$_REAUNOFIL,  0,    0,
;  416        P 0948  1             RMS$_FAC,   '94',   ABORT,   COB$_REAINCOPE,  0,    0,
;  417          0949  1             0,          '30',   ABORT,   COB$_ERRON_FIL,  0,    0)),
```

```
;  418          0950  1
;  419          0951  1        REA_R_S_T = REA_S_S_T,
;  420          0952  1        REA_R_S_T = UPLIT BYTE(Z(
;  421          0953  1            RMS$_EOF,   '16',   GOTO,    COB$_NO_NEXVAL,   NNVR, 0,
;  422          0954  1            RMS$_EOF,   '13',   GOTO,    COB$_NO_NEXLOG,   0,    NNVR,
;  423          0955  1            RMS$_RLK,   '92',   ABORT,   COB$_RECLOCREA,   0,    0,
;  424          0956  1            RMS$_ISI,   '15',   GOTO,    COB$_OPTMISREA,   OFNP, 0,
;  425          0957  1            RMS$_ISI,   '94',   ABORT,   COB$_REAUNOFIL,   0,    0,
;  426          0958  1            RMS$_RAB,   '94',   ABORT,   COB$_REAUNOFIL,   0,    0,
;  427          0959  1            RMS$_FAC,   '94',   ABORT,   COB$_REAINCOPE,   0,    0,
;  428          0960  1            0,          '30',   ABORT,   COB$_ERRON_FIL,   0,    0)),
;  429          0961  1
;  430        P 0962  1        REA_R_R_T = UPLIT BYTE(Z(
;  431        P 0963  1            RMS$_RNF,   '23',   GOTO,    COB$_RECNOTEXI,   0,    0,
;  432        P 0964  1            RMS$_MRN,   '23',   GOTO,    COB$_RECNOTEXI,   0,    0,
;  433        P 0965  1            RMS$_KEY,   '23',   GOTO,    COB$_RECNOTEXI,   0,    0,
;  434        P 0966  1            RMS$_KBF,   '23',   GOTO,    COB$_ERRON_FIL,   0,    0,
;  435        P 0967  1            RMS$_RLK,   '92',   ABORT,   COB$_RECLOCREA,   0,    0,
;  436        P 0968  1            RMS$_ISI,   '25',   GOTO,    COB$_OPTMISREA,   OFNP, 0,
;  437        P 0969  1            RMS$_ISI,   '94',   ABORT,   COB$_REAUNOFIL,   0,    0,
;  438        P 0970  1            RMS$_RAB,   '94',   ABORT,   COB$_REAUNOFIL,   0,    0,
;  439        P 0971  1            RMS$_FAC,   '94',   ABORT,   COB$_REAINCOPE,   0,    0,
;  440          0972  1            0,          '30',   ABORT,   COB$_ERRON_FIL,   0,    0)),
;  441          0973  1
;  442          0974  1        REA_I_S_T = REA_S_S_T,
;  443          0975  1        REA_I_S_T = UPLIT BYTE(Z(
;  444          0976  1            RMS$_EOF,   '16',   GOTO,    COB$_NO_NEXVAL,   NNVR, 0,
;  445          0977  1            RMS$_EOF,   '13',   GOTO,    COB$_NO_NEXLOG,   0,    NNVR,
;  446          0978  1            RMS$_RLK,   '92',   ABORT,   COB$_RECLOCREA,   0,    0,
;  447          0979  1            RMS$_ISI,   '15',   GOTO,    COB$_OPTMISREA,   OFNP, 0,
;  448          0980  1            RMS$_ISI,   '94',   ABORT,   COB$_REAUNOFIL,   0,    0,
;  449          0981  1            RMS$_RAB,   '94',   ABORT,   COB$_REAUNOFIL,   0,    0,
;  450          0982  1            RMS$_FAC,   '94',   ABORT,   COB$_REAINCOPE,   0,    0,
;  451          0983  1            0,          '30',   ABORT,   COB$_ERRON_FIL,   0,    0)),
;  452          0984  1
;  453        P 0985  1        REA_I_R_T = UPLIT BYTE(Z(
;  454        P 0986  1            RMS$_RNF,   '23',   GOTO,    COB$_RECNOTEXI,   0,    0,
;  455        P 0987  1            RMS$_RLK,   '92',   ABORT,   COB$_RECLOCREA,   0,    0,
;  456        P 0988  1            RMS$_ISI,   '25',   GOTO,    COB$_OPTMISREA,   OFNP, 0,
;  457        P 0989  1            RMS$_ISI,   '94',   ABORT,   COB$_REAUNOFIL,   0,    0,
;  458        P 0990  1            RMS$_RAB,   '94',   ABORT,   COB$_REAUNOFIL,   0,    0,
;  459        P 0991  1            RMS$_FAC,   '94',   ABORT,   COB$_REAINCOPE,   0,    0,
;  460          0992  1            0,          '30',   ABORT,   COB$_ERRON_FIL,   0,    0)),
;  461          0993  1
;  462        P 0994  1        WRI_S_S_T = UPLIT BYTE(Z(
;  463        P 0995  1            RMS$_FUL,   '34',   ABORT,   COB$_WRIBEYBOU,   0,    0,
;  464        P 0996  1            RMS$_ISI,   '94',   ABORT,   COB$_WRIUNOFIL,   0,    0,
;  465        P 0997  1            RMS$_RAB,   '94',   ABORT,   COB$_WRIUNOFIL,   0,    0,
;  466        P 0998  1            RMS$_FAC,   '94',   ABORT,   COB$_WRIINCOPE,   0,    0,
;  467          0999  1            0,          '30',   ABORT,   COB$_ERRON_FIL,   0,    0)),
;  468          1000  1
;  469        P 1001  1        WRI_R_S_T = UPLIT BYTE(Z(
;  470        P 1002  1            RMS$_FUL,   '24',   GOTO,    COB$_WRIBEYBOU,   0,    0,
;  471        P 1003  1            RMS$_RLK,   '92',   ABORT,   COB$_RECLOCWRI,   0,    0,
;  472        P 1004  1            RMS$_ISI,   '94',   ABORT,   COB$_WRIUNOFIL,   0,    0,
;  473        P 1005  1            RMS$_RAB,   '94',   ABORT,   COB$_WRIUNOFIL,   0,    0,
;  474        P 1006  1            RMS$_FAC,   '94',   ABORT,   COB$_WRIINCOPE,   0,    0,
```

```
;  475          1007  1          0,          '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  476          1008  1
;  477        P 1009  1      WRI_R_R_T = UPLIT BYTE(Z(
;  478        P 1010  1          RMS$_REX,   '22',   GOTO,    COB$_WRIDUPKEY,   0,   0,
;  479        P 1011  1          RMS$_FUL,   '24',   GOTO,    COB$_WRIBEYBOU,   0,   0,
;  480        P 1012  1          RMS$_KEY,   '24',   GOTO,    COB$_WRIBEYBOU,   0,   0,
;  481        P 1013  1          RMS$_RLK,   '92',   ABORT,   COB$_RECLOCWRI,   0,   0,
;  482        P 1014  1          RMS$_ISI,   '94',   ABORT,   COB$_WRIUNOFIL,   0,   0,
;  483        P 1015  1          RMS$_RAB,   '94',   ABORT,   COB$_WRIUNOFIL,   0,   0,
;  484        P 1016  1          RMS$_FAC,   '94',   ABORT,   COB$_WRIINCOPE,   0,   0,
;  485          1017  1          0,          '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  486          1018  1
;  487        P 1019  1      WRI_I_S_T = UPLIT BYTE(Z(
;  488        P 1020  1          RMS$_OK_DUP,'02',   CONTINUE,COB$_WRICREDUP,   0,   0,
;  489        P 1021  1          RMS$_SEQ,   '21',   GOTO,    COB$_WRINOTASC,   0,   0,
;  490        P 1022  1          RMS$_DUP,   '22',   GOTO,    COB$_WRIDISDUP,   0,   0,
;  491        P 1023  1          RMS$_FUL,   '24',   GOTO,    COB$_WRIBEYBOU,   0,   0,
;  492        P 1024  1          RMS$_RLK,   '92',   ABORT,   COB$_RECLOCWRI,   0,   0,
;  493        P 1025  1          RMS$_ISI,   '94',   ABORT,   COB$_WRIUNOFIL,   0,   0,
;  494        P 1026  1          RMS$_RAB,   '94',   ABORT,   COB$_WRIUNOFIL,   0,   0,
;  495        P 1027  1          RMS$_FAC,   '94',   ABORT,   COB$_WRIINCOPE,   0,   0,
;  496          1028  1          0,          '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  497          1029  1
;  498        P 1030  1      WRI_I_R_T = UPLIT BYTE(Z(
;  499        P 1031  1          RMS$_OK_DUP,'02',   CONTINUE,COB$_WRICREDUP,   0,   0,
;  500        P 1032  1          RMS$_DUP,   '22',   GOTO,    COB$_WRIDISDUP,   0,   0,
;  501        P 1033  1          RMS$_REX,   '22',   GOTO,    COB$_WRIDISDUP,   0,   0,
;  502        P 1034  1          RMS$_FUL,   '24',   GOTO,    COB$_WRIBEYBOU,   0,   0,
;  503        P 1035  1          RMS$_RLK,   '92',   ABORT,   COB$_RECLOCWRI,   0,   0,
;  504        P 1036  1          RMS$_ISI,   '94',   ABORT,   COB$_WRIUNOFIL,   0,   0,
;  505        P 1037  1          RMS$_RAB,   '94',   ABORT,   COB$_WRIUNOFIL,   0,   0,
;  506        P 1038  1          RMS$_FAC,   '94',   ABORT,   COB$_WRIINCOPE,   0,   0,
;  507          1039  1          0,          '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  508          1040  1
;  509        P 1041  1      DEL_R_S_T = UPLIT BYTE(Z(
;  510        P 1042  1          RMS$_RLK,   '92',   ABORT,   COB$_RECLOCDEL,   0,   0,
;  511        P 1043  1          RMS$_CUR,   '93',   ABORT,   COB$_DELNO_R_S,   0,   0,
;  512        P 1044  1          RMS$_ISI,   '94',   ABORT,   COB$_DELUNOFIL,   0,   0,
;  513        P 1045  1          RMS$_RAB,   '94',   ABORT,   COB$_DELUNOFIL,   0,   0,
;  514        P 1046  1          RMS$_FAC,   '94',   ABORT,   COB$_DELINCOPE,   0,   0,
;  515          1047  1          0,          '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  516          1048  1
;  517        P 1049  1      DEL_R_R_T = UPLIT BYTE(Z(
;  518        P 1050  1          RMS$_RNF,   '23',   GOTO,    COB$_RECNOTEXI,   0,   0,
;  519        P 1051  1          RMS$_MRN,   '23',   GOTO,    COB$_RECNOTEXI,   0,   0,
;  520        P 1052  1          RMS$_RLK,   '92',   ABORT,   COB$_RECLOCDEL,   0,   0,
;  521        P 1053  1          RMS$_ISI,   '94',   ABORT,   COB$_DELUNOFIL,   0,   0,
;  522        P 1054  1          RMS$_RAB,   '94',   ABORT,   COB$_DELUNOFIL,   0,   0,
;  523        P 1055  1          RMS$_FAC,   '94',   ABORT,   COB$_DELINCOPE,   0,   0,
;  524          1056  1          0,          '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  525          1057  1
;  526          1058  1      DEL_I_S_T - DEL_R_S_T,
;  527          1059  1 !    DEL_I_S_T = UPLIT BYTE(Z(
;  528          1060  1 !        RMS$_RLK,   '92',   ABORT,   COB$_RECLOCDEL,   0,   0,
;  529          1061  1 !        RMS$_CUR,   '93',   ABORT,   COB$_DELNO_R_S,   0,   0,
;  530          1062  1 !        RMS$_ISI,   '94',   ABORT,   COB$_DELUNOFIL,   0,   0,
;  531          1063  1 !        RMS$_RAB,   '94',   ABORT,   COB$_DELUNOFIL,   0,   0,
```

```
;  532          1064  1 !          RMS$_FAC,    '94',   ABORT,   COB$_DELINCOPE,   0,   0,
;  533          1065  1 !          0,           '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  534          1066  1
;  535          1067  1        DEL_I_R_T = DEL_R_R_T,
;  536          1068  1 !      DEL_I_R_T = UPLIT BYTE(Z(
;  537          1069  1 !          RMS$_RNF,    '23',   GOTO,    COB$_RECNOTEXI,   0,   0,
;  538          1070  1 !          RMS$_MRN,    '23',   GOTO,    COB$_RECNOTEXI,   0,   0,
;  539          1071  1 !          RMS$_RLK,    '92',   ABORT,   COB$_RECLOCDEL,   0,   0,
;  540          1072  1 !          RMS$_ISI,    '94',   ABORT,   COB$_DELUNOFIL,   0,   0,
;  541          1073  1 !          RMS$_RAB,    '94',   ABORT,   COB$_DELUNOFIL,   0,   0,
;  542          1074  1 !          RMS$_FAC,    '94',   ABORT,   COB$_DELINCOPE,   0,   0,
;  543          1075  1 !          0,           '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  544          1076  1
;  545        P 1077  1        REW_S_S_T = UPLIT BYTE(Z(
;  546        P 1078  1            RMS$_CUR,    '93',   ABORT,   COB$_REWNO_R_S,   0,   0,
;  547        P 1079  1            RMS$_ISI,    '94',   ABORT,   COB$_REWUNOFIL,   0,   0,
;  548        P 1080  1            RMS$_RAB,    '94',   ABORT,   COB$_REWUNOFIL,   0,   0,
;  549        P 1081  1            RMS$_FAC,    '94',   ABORT,   COB$_REWINCOPE,   0,   0,
;  550          1082  1            0,           '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  551          1083  1
;  552        P 1084  1        REW_R_S_T = UPLIT BYTE(Z(
;  553        P 1085  1            RMS$_RLK,    '92',   ABORT,   COB$_RECLOCREW,   0,   0,
;  554        P 1086  1            RMS$_CUR,    '93',   ABORT,   COB$_REWNO_R_S,   0,   0,
;  555        P 1087  1            RMS$_ISI,    '94',   ABORT,   COB$_REWUNOFIL,   0,   0,
;  556        P 1088  1            RMS$_RAB,    '94',   ABORT,   COB$_REWUNOFIL,   0,   0,
;  557        P 1089  1            RMS$_FAC,    '94',   ABORT,   COB$_REWINCOPE,   0,   0,
;  558          1090  1            0,           '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  559          1091  1
;  560        P 1092  1        REW_R_R_T = UPLIT BYTE(Z(
;  561        P 1093  1            RMS$_RNF,    '23',   GOTO,    COB$_RECNOTEXI,   0,   0,
;  562        P 1094  1            RMS$_MRN,    '23',   GOTO,    COB$_RECNOTEXI,   0,   0,
;  563        P 1095  1            RMS$_RLK,    '92',   ABORT,   COB$_RECLOCREW,   0,   0,
;  564        P 1096  1            RMS$_ISI,    '94',   ABORT,   COB$_REWUNOFIL,   0,   0,
;  565        P 1097  1            RMS$_RAB,    '94',   ABORT,   COB$_REWUNOFIL,   0,   0,
;  566        P 1098  1            RMS$_FAC,    '94',   ABORT,   COB$_REWINCOPE,   0,   0,
;  567          1099  1            0,           '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  568          1100  1
;  569        P 1101  1        REW_I_S_T = UPLIT BYTE(Z(
;  570        P 1102  1            RMS$_OK_DUP,'02',   CONTINUE,COB$_REWCREDUP,   0,   0,
;  571        P 1103  1            RMS$_CHG,    '21',   GOTO,    COB$_PRIKEYCHA,   0,   0,
;  572        P 1104  1            RMS$_DUP,    '22',   GOTO,    COB$_REWDISDUP,   0,   0,
;  573        P 1105  1            RMS$_RLK,    '92',   ABORT,   COB$_RECLOCREW,   0,   0,
;  574        P 1106  1            RMS$_CUR,    '93',   ABORT,   COB$_REWNO_R_S,   0,   0,
;  575        P 1107  1            RMS$_ISI,    '94',   ABORT,   COB$_REWUNOFIL,   0,   0,
;  576        P 1108  1            RMS$_RAB,    '94',   ABORT,   COB$_REWUNOFIL,   0,   0,
;  577        P 1109  1            RMS$_FAC,    '94',   ABORT,   COB$_REWINCOPE,   0,   0,
;  578          1110  1            0,           '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
;  579          1111  1
;  580        P 1112  1        REW_I_R_T = UPLIT BYTE(Z(
;  581        P 1113  1            RMS$_OK_DUP,'02',   CONTINUE,COB$_REWCREDUP,   0,   0,
;  582        P 1114  1            RMS$_DUP,    '22',   GOTO,    COB$_REWDISDUP,   0,   0,
;  583        P 1115  1            RMS$_RNF,    '23',   GOTO,    COB$_RECNOTEXI,   0,   0,
;  584        P 1116  1            RMS$_RLK,    '92',   ABORT,   COB$_RECLOCREW,   0,   0,
;  585        P 1117  1            RMS$_ISI,    '94',   ABORT,   COB$_REWUNOFIL,   0,   0,
;  586        P 1118  1            RMS$_RAB,    '94',   ABORT,   COB$_REWUNOFIL,   0,   0,
;  587        P 1119  1            RMS$_FAC,    '94',   ABORT,   COB$_REWINCOPE,   0,   0,
;  588          1120  1            0,           '30',   ABORT,   COB$_ERRON_FIL,   0,   0)),
```

```
;  589          1121  1
;  590        P 1122  1        STA_R_S_T = UPLIT BYTE(Z(
;  591        P 1123  1            RMS$_RNF,   '23',  GOTO,    COB$_RECNOTEXI,   0,    NNVR,
;  592        P 1124  1            RMS$_MRN,   '23',  GOTO,    COB$_RECNOTEXI,   0,    NNVR,
;  593        P 1125  1            RMS$_ISI,   '25',  GOTO,    COB$_OPTMISSTA,   OFNP, 0,
;  594        P 1126  1            RMS$_ISI,   '94',  ABORT,   COB$_STAUNOFIL,   0,    0,
;  595        P 1127  1            RMS$_RAB,   '94',  ABORT,   COB$_STAUNOFIL,   0,    0,
;  596        P 1128  1            RMS$_RLK,   '92',  ABORT,   COB$_RECLOCSTA,   0,    0,
;  597        P 1129  1            RMS$_FAC,   '94',  ABORT,   COB$_STAINCOPE,   0,    0,
;  598          1130  1            0,          '30',  ABORT,   COB$_ERRON_FIL,   0,    0)),
;  599          1131  1
;  600          1132  1        STA_I_S_T = STA_R_S_T,
;  601          1133  1    !   STA_I_S_T = UPLIT BYTE(Z(
;  602          1134  1    !       RMS$_RNF,   '23',  GOTO,    COB$_RECNOTEXI,   0,    NNVR,
;  603          1135  1    !       RMS$_MRN,   '23',  GOTO,    COB$_RECNOTEXI,   0,    NNVR,
;  604          1136  1    !       RMS$_ISI,   '25',  GOTO,    COB$_OPTMISSTA,   OFNP, 0,
;  605          1137  1    !       RMS$_ISI,   '94',  ABORT,   COB$_STAUNOFIL,   0,    0,
;  606          1138  1    !       RMS$_RAB,   '94',  ABORT,   COB$_STAUNOFIL,   0,    0,
;  607          1139  1    !       RMS$_RLK,   '92',  ABORT,   COB$_RECLOCSTA,   0,    0,
;  608          1140  1    !       RMS$_FAC,   '94',  ABORT,   COB$_STAINCOPE,   0,    0,
;  609          1141  1    !       0,          '30',  ABORT,   COB$_ERRON_FIL,   0,    0));
;  610          1142  1
;  611        P 1143  1        UNL_T = UPLIT BYTE(Z(
;  612        P 1144  1            RMS$_RNL,   '00',  SUCCEED, COB$_RECNOTLOC,   0,    0,
;  613        P 1145  1            RMS$_ISI,   '94',  ABORT,   COB$_UNLUNOFIL,   0,    0,
;  614        P 1146  1            RMS$_RAB,   '94',  ABORT,   COB$_UNLUNOFIL,   0,    0,
;  615        P 1147  1            RMS$_CUR,   '93',  ABORT,   COB$_UNLNO_CUR,   0,    0,
;  616          1148  1            0,          '30',  ABORT,   COB$_ERRON_FIL,   0,    0));
```

```
;  618          1149  1 !+
;  619          1150  1 ! The following table is indexed by an I/O statement type to yield the address
;  620          1151  1 ! (relative to BASE) of the appropriate action table.  Negative values are used
;  621          1152  1 ! to indicate an error condition (COB$_INVARG).
;  622          1153  1 !-
;  623          1154  1 BIND
;  624          1155  1     PLIT_TABLE = UPLIT WORD(
;  625          1156  1         OPE_T - BASE,           ! COB$K_EXC_OPESS= 1,
;  626          1157  1         -1,                     ! 2
;  627          1158  1         OPE_T - BASE,           ! COB$K_EXC_OPERS= 3,
;  628          1159  1         OPE_T - BASE,           ! COB$K_EXC_OPERR= 4,
;  629          1160  1         OPE_T - BASE,           ! COB$K_EXC_OPEIS= 5,
;  630          1161  1         OPE_T - BASE,           ! COB$K_EXC_OPEIR= 6,
;  631          1162  1         CLO_T - BASE,           ! COB$K_EXC_CLOSS= 7,
;  632          1163  1         -1,                     ! 8
;  633          1164  1         CLO_T - BASE,           ! COB$K_EXC_CLORS= 9,
;  634          1165  1         CLO_T - BASE,           ! COB$K_EXC_CLORR=10,
;  635          1166  1         CLO_T - BASE,           ! COB$K_EXC_CLOIS=11,
;  636          1167  1         CLO_T - BASE,           ! COB$K_EXC_CLOIR=12,
;  637          1168  1         REA_S_S_T - BASE,       ! COB$K_EXC_REASS=13,
;  638          1169  1         -1,                     ! 14
;  639          1170  1         REA_R_S_T - BASE,       ! COB$K_EXC_REARS=15,
;  640          1171  1         REA_R_R_T - BASE,       ! COB$K_EXC_REARR=16,
;  641          1172  1         REA_I_S_T - BASE,       ! COB$K_EXC_REAIS=17,
;  642          1173  1         REA_I_R_T - BASE,       ! COB$K_EXC_REAIR=18,
;  643          1174  1         WRI_S_S_T - BASE,       ! COB$K_EXC_WRISS=19,
;  644          1175  1         -1,                     ! 20
;  645          1176  1         WRI_R_S_T - BASE,       ! COB$K_EXC_WRIRS=21,
;  646          1177  1         WRI_R_R_T - BASE,       ! COB$K_EXC_WRIRR=22,
;  647          1178  1         WRI_I_S_T - BASE,       ! COB$K_EXC_WRIIS=23,
;  648          1179  1         WRI_I_R_T - BASE,       ! COB$K_EXC_WRIIR=24,
;  649          1180  1         REW_S_S_T - BASE,       ! COB$K_EXC_REWSS=25,
;  650          1181  1         -1,                     ! 26
;  651          1182  1         REW_R_S_T - BASE,       ! COB$K_EXC_REWRS=27,
;  652          1183  1         REW_R_R_T - BASE,       ! COB$K_EXC_REWRR=28,
;  653          1184  1         REW_I_S_T - BASE,       ! COB$K_EXC_REWIS=29,
;  654          1185  1         REW_I_R_T - BASE,       ! COB$K_EXC_REWIR=30,
;  655          1186  1         -1,                     ! COB$K_EXC_DELSS=31,   !???  Is this used?
;  656          1187  1         -1,                     ! 32
;  657          1188  1         DEL_R_S_T - BASE,       ! COB$K_EXC_DELRS=33,
;  658          1189  1         DEL_R_R_T - BASE,       ! COB$K_EXC_DELRR=34,
;  659          1190  1         DEL_I_S_T - BASE,       ! COB$K_EXC_DELIS=35,
;  660          1191  1         DEL_I_R_T - BASE,       ! COB$K_EXC_DELIR=36,
;  661          1192  1         -1,                     ! 37
;  662          1193  1         -1,                     ! 38
;  663          1194  1         STA_R_S_T - BASE,       ! COB$K_EXC_STARS=39,
;  664          1195  1         -1,                     ! 40
;  665          1196  1         STA_I_S_T - BASE,       ! COB$K_EXC_STAIS=41,
;  666          1197  1         -1,                     ! 42
;  667          1198  1         CLR_S_S_T- BASE,        ! COB$K_EXC_CLRSS=43,
;  668          1199  1         UNL_T    - BASE         ! COB$K_EXC_UNLCK=44,
;  669          1200  1         ) - 2: VECTOR[,WORD];
```

```
;  671          1201  1 GLOBAL ROUTINE COB$IOEXCEPTION (
;  672          1202  1
;  673          1203  1         FLAGS,          ! Type of I/O statement and type of exception
;  674          1204  1         RAB,            ! Address of associated RAB
;  675          1205  1         EXCLAB,         ! Exception transfer label (optional)
;  676          1206  1         STATUS,         ! Address of file status variable (optional)
;  677          1207  1         ANSI74          ! If true use ANSI 74 file status values (optional)
;  678          1208  1
;  679          1209  1                                 ) =
;  680          1210  1
;  681          1211  1 !++
;  682          1212  1 !
;  683          1213  1 ! FUNCTIONAL DESCRIPTION:
;  684          1214  1 !
;  685          1215  1 !       This procedure is called to process a wide variety of I/O
;  686          1216  1 ! exceptions.  Depending on the nature of the exception, it sets the
;  687          1217  1 ! file status variable, causes USE procedures to be invoked and
;  688          1218  1 ! signals errors.
;  689          1219  1 !
;  690          1220  1 ! FORMAL PARAMETERS:
;  691          1221  1 !
;  692          1222  1 !       FLAGS.rl.v      - Type of I/O statement and type of exception
;  693          1223  1 !       RAB.ra.v        - Address of RAB
;  694          1224  1 !       EXCLAB.jzi.v    - Exception transfer label (optional)
;  695          1225  1 !       STATUS.ra.r     - Address of file status variable (optional)
;  696          1226  1 !       ANSI74.rlu.v     - If true use ANSI 74 file status values (optional)
;  697          1227  1 !
;  698          1228  1 !
;  699          1229  1 ! IMPLICIT INPUTS:
;  700          1230  1 !
;  701          1231  1 !       NONE
;  702          1232  1 !
;  703          1233  1 ! IMPLICIT OUTPUTS:
;  704          1234  1 !
;  705          1235  1 !       NONE
;  706          1236  1 !
;  707          1237  1 ! ROUTINE VALUE:
;  708          1238  1 ! COMPLETION CODES:
;  709          1239  1 !
;  710          1240  1 !       Value TRUE if SUCCEED, FALSE if other.
;  711          1241  1 !
;  712          1242  1 ! SIDE EFFECTS:
;  713          1243  1 !
;  714          1244  1 !       NONE
;  715          1245  1 !
;  716          1246  1 !--
```

```
;  718          1247  1 !+
;  719          1248  1 !
;  720          1249  1 ! The USE list structure appears as follows:
;  721          1250  1 !
;  722          1251  1 !
;  723          1252  1 !       ------------------------------------------------
;  724          1253  1 !     0 !              COB$A_USE_PNC                     !
;  725          1254  1 !       ------------------------------------------------
;  726          1255  1 !     1 !              COB$A_USE_MODES                   ! BASE OF OPEN MODE
;  727          1256  1 !       ------------------------------------------------
;  728          1257  1 !     2 !                    .                           !
;  729          1258  1 !       !                    .                           !
;  730          1259  1 !       !                    .  MODE SPECIFIC            !
;  731          1260  1 !       !                    .  USE ENTRIES              !
;  732          1261  1 !       !                    .                           !
;  733          1262  1 !       !                    .                           !
;  734          1263  1 !       ------------------------------------------------
;  735          1264  1 !    36 !                    !COB$B_GUSE_CNT !COB$B_USE_CNT !
;  736          1265  1 !       ------------------------------------------------
;  737          1266  1 !    37 !              COB$A_USE_FILES                   ! BASE OF 1ST FILE ENTRY
;  738          1267  1 !       ------------------------------------------------
;  739          1268  1 !       !                    .                           !
;  740          1269  1 !       !                    .  FILE SPECIFIC GLOBAL USE !
;  741          1270  1 !       !                    .  PROCEDURES IN LOCAL PRGM !
;  742          1271  1 !       !                    .                           !
;  743          1272  1 !       !                    .  FILE SPECIFIC LOCAL USE  !
;  744          1273  1 !       !                    .  PROCEDURES IN LOCAL PRGM !
;  745          1274  1 !       !                    .                           !
;  746          1275  1 !       !                    .  FILE SPECIFIC GLOBAL USE !
;  747          1276  1 !       !                    .  USE ENTRIES DEFINED IN   !
;  748          1277  1 !       !                    .  CONTAINING PROGRAM       !
;  749          1278  1 !       !                    .                           !
;  750          1279  1 !       ------------------------------------------------
;  751          1280  1 !
;  752          1281  1 !
;  753          1282  1 !   where
;  754          1283  1 !        COB$A_USE_PNC    - address of perform nest counter for the declaring
;  755          1284  1 !                           program.
;  756          1285  1 !        COB$A_USE_MODES  - Base of the OPEN MODE entries.
;  757          1286  1 !        COB$B_USE_COUNT  - Number of file specific USE procedure defined for
;  758          1287  1 !                           this program.  This includes both local and global
;  759          1288  1 !                           file specific procedures defined both in this program
;  760          1289  1 !                           and in containing programs.
;  761          1290  1 !        COB$B_GUSE_COUNT- Number of file specific global USE procedures defined
;  762          1291  1 !                           in the local program.
;  763          1292  1 !        COB$A_USE_FILES  - Base of first file entry.
;  764          1293  1 !-
```

```
;  766          1294  1 !+
;  767          1295  1 ! The structure of the USE entries appears as follows:
;  768          1296  1 !
;  769          1297  1 !
;  770          1298  1 !
;  771          1299  1 !       ------------------------------------------------
;  772          1300  1 !     0 !              COB$A_USE_PROC                  !
;  773          1301  1 !       ------------------------------------------------
;  774          1302  1 !     1 !              COB$A_USE_EOPR                  !
;  775          1303  1 !       ------------------------------------------------
;  776          1304  1 !     2 !              COB$A_USE_PAB                   !
;  777          1305  1 !       ------------------------------------------------
;  778          1306  1 !
;  779          1307  1 !   where
;  780          1308  1 !       COB$A_USE_PROC  -  Address of the USE procedure.
;  781          1309  1 !       COB$A_USE_EOPR  -  Address of end of perform range block.
;  782          1310  1 !                          It is a pointer to the end of perform
;  783          1311  1 !                          range block for the USE procedure if
;  784          1312  1 !                          the entry was defined in this program
;  785          1313  1 !                          or 0 if it was defined in a containing
;  786          1314  1 !                          program.
;  787          1315  1 !       COB$A_USE_RAB   -  Address of RAB (only in file entries).
;  788          1316  1 !-
```

```
;  790          1317  2     BEGIN
;  791          1318  2     MAP
;  792          1319  2         FLAGS:          BLOCK[,BYTE],          ! I/O statement and error type
;  793          1320  2         RAB:            REF BLOCK[,BYTE];      ! Address of RAB
;  794          1321  2
;  795          1322  2
;  796          1323  2     LOCAL
;  797          1324  2         FAB:            REF BLOCK[,BYTE],      ! Address of FAB
;  798          1325  2         NAM:            REF BLOCK[,BYTE],      ! Address of NAM block-used to
;  799          1326  2                                                ! communicate optional filename
;  800          1327  2                                                ! related information.
;  801          1328  2         TABLE:          REF BLOCK[,BYTE],      ! Pointer to status decode table entry
;  802          1329  2         PMS_STS,                               ! STS value from FAB or RAB
;  803          1330  2         RMS_STV,                               ! STV value from FAB or RAB
;  804          1331  2         FILESTAT,                              ! COBOL file status
;  805          1332  2         ACTION,                                ! Recovery action
;  806          1333  2         SFP:            REF BLOCK[,BYTE],      ! Saved FP
;  807          1334  2         USE:            REF BLOCK[,BYTE],      ! Pointer to USE list
;  808          1335  2         USEENT:         REF BLOCK[,BYTE],      ! Pointer to USE list entry
;  809          1336  2         ERR_MSG_NUM,                           ! Longword COB$ error number
;  810          1337  2         FILE_ADDR,                             ! Stored entry point to file specific USE procedure
;  811          1338  2         OPEN_MD_ADDR,                          ! Stored entry point to open mode specific USE procedure
;  812          1339  2         RSADESC:        VECTOR[2];             ! Descriptor for RSA (resultant string area)
;  813          1340  2
;  814          1341  2     BUILTIN
;  815          1342  2         ACTUALCOUNT,
;  816          1343  2         FP;
;  817          1344  2
;  818          1345  2     MAP
;  819          1346  2         FP:     REF BLOCK[,BYTE];
;  820          1347  2
;  821          1348  2
;  822          1349  2
;  823          1350  2
;  824          1351  2     !+
;  825          1352  2     ! Ensure that the FLAGS and RAB arguments are present.
;  826          1353  2     !-
;  827          1354  2
;  828          1355  2     IF ACTUALCOUNT() LSS 2                             !\Error if FLAGS and RAB
;  829          1356  2       THEN                                             !/arguments are missing
;  830          1357  3         BEGIN
;  831          1358  3         SIGNAL_STOP(OTS$_FATINTERR);
;  832          1359  3         RETURN 0;
;  833          1360  2         END;
;  834          1361  2     IF .RAB EQL 0
;  835          1362  2       THEN
;  836          1363  3         BEGIN
;  837          1364  3         SIGNAL_STOP(OTS$_FATINTERR);                   ! Error if RAB addr = 0
;  838          1365  3         RETURN 0;
;  839          1366  2         END;
;  840          1367  2     FAB = .RAB[RAB$L_FAB];                             ! Fetch FAB address
;  841          1368  2     IF .FAB EQL 0
;  842          1369  2       THEN
;  843          1370  3         BEGIN
;  844          1371  3         SIGNAL_STOP(OTS$_FATINTERR);                   ! Error if FAB addr = 0
;  845          1372  3         RETURN 0;
;  846          1373  2         END;
```

```
;  847          1374  2     NAM = .FAB[FAB$L_NAM];                          ! Fetch addr of NAM block
;  848          1375  2     IF .NAM EQL 0
;  849          1376  2       THEN
;  850          1377  3         BEGIN
;  851          1378  3         SIGNAL_STOP(OTS$_FATINTERR);                ! Error if addr of NAM block = 0
;  852          1379  3         RETURN 0;
;  853          1380  2         END;
;  854          1381  2
;  855          1382  2 !+
;  856          1383  2 ! Initialize the temporary storage to be used later
;  857          1384  2 ! when searching for a USE procedure.
;  858          1385  2 !-
;  859          1386  2
;  860          1387  2         FILE_ADDR = 0;
;  861          1388  2         OPEN_MD_ADDR = 0;
;  862          1389  2
;  863          1390  2 !+
;  864          1391  2 ! Fetch the address and length of the resultant string area
;  865          1392  2 ! as defined in the NAM block.  If the length field is zero,
;  866          1393  2 ! then fetch the extended string length field.  If that too
;  867          1394  2 ! is a zero length, then just use the FAB file name string
;  868          1395  2 ! name and string length.
;  869          1396  2 !--
;  870          1397  2
;  871          1398  2     RSADESC[1] = .NAM[NAM$L_RSA];                   ! Fetch addr of resultant string area
;  872          1399  2     RSADESC[0] = .NAM[NAM$B_RSL];                   ! Fetch resultant string size
;  873          1400  2     IF .RSADESC[0] EQL 0
;  874          1401  2       THEN
;  875          1402  2         RSADESC[0] = .NAM[NAM$B_ESL];               ! Fetch extended string length
;  876          1403  2     IF .RSADESC[0] EQL 0
;  877          1404  2       THEN
;  878          1405  3         BEGIN
;  879          1406  3         RSADESC[0] = .FAB[FAB$B_FNS];               ! Fetch file name string size
;  880          1407  3
;  881          1408  3         !+
;  882          1409  3         ! Copy the string found in the FAB into the resultant
;  883          1410  3         ! string area.  By doing this, COBOL will just map a special
;  884          1411  3         ! register to this area for obtaining the resultant name
;  885          1412  3         ! string.
;  886          1413  3         !-
;  887          1414  3
;  888          1415  3         CH$MOVE (.FAB[FAB$B_FNS],.FAB[FAB$L_FNA],.NAM[NAM$L_RSA]);
;  889          1416  2         END;
;  890          1417  2
;  891          1418  2
;  892          1419  2     !+
;  893          1420  2     ! Case on the error type.  Note that COB$K_EXC_MINM
;  894          1421  2     ! and COB$K_EXC_MAXM are the minimum (0) and maximum (7)
;  895          1422  2     ! values to be used for case statements involving error
;  896          1423  2     ! types.
;  897          1424  2     !-
;  898          1425  2
;  899          1426  2     CASE .FLAGS[COB$W_EXC_ERROR] FROM COB$K_EXC_MINM TO COB$K_EXC_MAXM OF
;  900          1427  2       SET
;  901          1428  2
;  902          1429  2         [OUTRANGE]:                                 ! Invalid value
;  903          1430  3             BEGIN
```

```
;  904          1431  3          SIGNAL_STOP(COB$_INVARG);
;  905          1432  3          RETURN 0;
;  906          1433  2          END;
;  907          1434  2
;  908          1435  2      [COB$K_EXC_RAB, COB$K_EXC_FAB]:                 ! RMS RAB/FAB error
;  909          1436  3          BEGIN
;  910          1437  3
;  911          1438  3          !+
;  912          1439  3          ! RMS error.  Pick up the appropriate STS and STV.
;  913          1440  3          !-
;  914          1441  3
;  915          1442  3          IF .FLAGS[COB$W_EXC_ERROR] EQL COB$K_EXC_RAB ! Check specifically for RAB error
;  916          1443  3            THEN
;  917          1444  4              BEGIN
;  918          1445  4              IF .RAB[RAB$L_STS] EQL 0                !\If status of RAB = 0
;  919          1446  4                THEN                                  ! then signal as a
;  920          1447  5                  BEGIN                               !/fatal internal error.
;  921          1448  5                  SIGNAL_STOP(OTS$_FATINTERR);
;  922          1449  5                  RETURN 0;
;  923          1450  4                  END;
;  924          1451  4              RMS_STS = .RAB[RAB$L_STS];              ! Load status code from RAB
;  925          1452  4              RMS_STV = .RAB[RAB$L_STV];              ! Load status value from RAB
;  926          1453  4              END
;  927          1454  4
;  928          1455  4      !+
;  929          1456  4      ! Here we know that we have a FAB error.
;  930          1457  4      !-
;  931          1458  4
;  932          1459  3          ELSE
;  933          1460  4              BEGIN
;  934          1461  4              IF .FAB[FAB$L_STS] EQL 0                !\If status of FAB = 0,
;  935          1462  4                THEN                                  ! then signal a
;  936          1463  5                  BEGIN                               !/fatal internal error.
;  937          1464  5                  SIGNAL_STOP(OTS$_FATINTERR);
;  938          1465  5                  RETURN 0;
;  939          1466  4                  END;
;  940          1467  4              RMS_STS = .FAB[FAB$L_STS];              ! Load status code from FAB
;  941          1468  4              RMS_STV = .FAB[FAB$L_STV];              ! Load status value from FAB
;  942          1469  3              END;
;  943          1470  3
;  944          1471  3
;  945          1472  3              !+
;  946          1473  3              ! Map the STS value into the appropriate COBOL file status.
;  947          1474  3              !
;  948          1475  3              ! First, determine the lookup table to be used from the statement
;  949          1476  3              ! type parameter.  Note that this must not be done by indexing a
;  950          1477  3              ! table of addresses because of the need for this routine to be
;  951          1478  3              ! position-independent.
;  952          1479  3              !-
;  953          1480  3
;  954          1481  3              IF .RMS_STS
;  955          1482  3                THEN
;  956          1483  4                  BEGIN                               ! Success cases -- are not
;  957          1484  4                                                      ! found by table look-up.
;  958          1485  4
;  959          1486  4      !+
;  960          1487  4      ! Perform a CASE on the I/O stmt type.  Note that
```

```
;  961          1488  4          ! COB$K_EXC_MINS and COB$K_EXC_MAXS indicate the
;  962          1489  4          ! minimum (1) and maximum (44) values for CASE
;  963          1490  4          ! values for I/O statement types.
;  964          1491  4          !-
;  965          1492  4
;  966          1493  4                  CASE .FLAGS[COB$W_EXC_STMT]
;  967          1494  4                    FROM COB$K_EXC_MINS TO COB$K_EXC_MAXS OF
;  968          1495  4                      SET
;  969          1496  4                      [COB$K_EXC_REASS,
;  970          1497  4                       COB$K_EXC_REARS,
;  971          1498  4                       COB$K_EXC_REARR,
;  972          1499  4                       COB$K_EXC_REAIS,
;  973          1500  4                       COB$K_EXC_REAIR]:
;  974          1501  5                          BEGIN
;  975          1502  5
;  976          1503  5                          !+
;  977          1504  5                          ! Successful read, reset NNVR (no next valid
;  978          1505  5                          ! record) - meaning EOF has been detected.
;  979          1506  5                          !-
;  980          1507  5
;  981          1508  5                          RAB[COB$V_CTX_NNVR] = 0;
;  982          1509  5
;  983          1510  5                          !+
;  984          1511  5                          ! If the RMS_STS is for a soft record lock,
;  985          1512  5                          ! then the FILESTAT should be set to 90 with
;  986          1513  5                          ! an ACTION code of CONTINUE.
;  987          1514  5                          !-
;  988          1515  5
;  989          1516  5                          IF .RMS_STS EQL RMS$_OK_RLK OR .RMS_STS EQL RMS$_OK_RRL
;  990          1517  5                            THEN
;  991          1518  6                              BEGIN
;  992          1519  6                              FILESTAT = '90';
;  993          1520  6                              ACTION = CONTINUE;
;  994          1521  6                              END
;  995          1522  5                            ELSE
;  996          1523  6                              BEGIN
;  997          1524  6                              FILESTAT = '00' ;
;  998          1525  6                              ACTION = SUCCEED ;
;  999          1526  5                              END;
; 1000          1527  4                      END ;
; 1001          1528  4                  [COB$K_EXC_WRIIS,
; 1002          1529  4                   COB$K_EXC_WRIIR,
; 1003          1530  4                   COB$K_EXC_REWIS,
; 1004          1531  4                   COB$K_EXC_REWIR]:
; 1005          1532  5                      BEGIN
; 1006          1533  5
; 1007          1534  5                      !+
; 1008          1535  5                      ! Successful write or rewrite on irdexed
; 1009          1536  5                      ! sequential file.
; 1010          1537  5                      !-
; 1011          1538  5
; 1012          1539  5                      IF .RMS_STS EQL RMS$_OK_DUP
; 1013          1540  5                      THEN
; 1014          1541  6                          BEGIN
; 1015          1542  6                          FILESTAT = '02' ;
; 1016          1543  6                          ACTION = CONTINUE ;
; 1017          1544  6                          END
```

```
;1018     1545  5                    ELSE
;1019     1546  6                        BEGIN
;1020     1547  6                        FILESTAT = '00' ;
;1021     1548  6                        ACTION = SUCCEED ;
;1022     1549  6                        END
;1023     1550  4                    END ;
;1024     1551  4                [INRANGE, OUTRANGE]:                ! Unexpected success code
;1025     1552  5                    BEGIN
;1026     1553  5                    SIGNAL_STOP (COB$_INVARG) ;     !\Signal an error due to
;1027     1554  5                    RETURN 0;                       !/invalid success code
;1028     1555  4                    END;
;1029     1556  4                TES ;
;1030     1557  4            END
;1031     1558  4
;1032     1559  4        !+
;1033     1560  4        ! This section of code deals with the error cases for RMS
;1034     1561  4        ! RAB/FAB errors, where we know that the LSB of RMS_STS
;1035     1562  4        ! (status code) is zero.
;1036     1563  4        !-
;1037     1564  4
;1038     1565  3            ELSE
;1039     1566  4                BEGIN
;1040     1567  4
;1041     1568  4                !+
;1042     1569  4                ! Perform a CASE on I/O statement type.  Note that
;1043     1570  4                ! COB$K_EXC_MINS and COB$K_EXC_MAXS refers to the
;1044     1571  4                ! minimum (1) and maximum (44) CASE values that
;1045     1572  4                ! pertain to I/O statement types.
;1046     1573  4                !-
;1047     1574  5                IF  BEGIN
;1048     1575  6                    IF (.FLAGS[COB$W_EXC_STMT] LSS COB$K_EXC_MINS OR
;1049     1576  6                        .FLAGS[COB$W_EXC_STMT] GTR COB$K_EXC_MAXS)
;1050     1577  5                    THEN
;1051     1578  5                        1
;1052     1579  5                    ELSE
;1053     1580  5                        (TABLE = .PLIT_TABLE[.FLAGS[COB$W_EXC_STMT]]) LSS 0
;1054     1581  5                    END
;1055     1582  4                THEN
;1056     1583  5                    BEGIN                           ! Invalid error code
;1057     1584  5                    SIGNAL_STOP(COB$_INVARG);
;1058     1585  5                    RETURN 0;
;1059     1586  4                    END;
;1060     1587  4
;1061     1588  4                !+
;1062     1589  4                ! Depending on the type of error that was found,
;1063     1590  4                ! load the appropriate table addresses into "TABLE"
;1064     1591  4                ! which is the pointer to the status decode table
;1065     1592  4                ! entry.
;1066     1593  4                !-
;1067     1594  4                TABLE = BASE + .TABLE;
;1068     1595  4
;1069     1596  4                !+
;1070     1597  4                ! Find the appropriate entry.  Since every table ends with a 0
;1071     1598  4                ! there must be one.  Note that [TAB_STATUS] will contain a
;1072     1599  4                ! RMS status code or a zero.
;1073     1600  4                !-
;1074     1601  4
```

```
;  1075          1602  4          UNTIL
;  1076          1603  5              (.TABLE[TAB_STATUS] EQL 0 OR .TABLE[TAB_STATUS] EQL .RMS_STS)
;  1077          1604  4          AND
;  1078          1605  4
;  1079          1606  4              !+
;  1080          1607  4              ! If the test-state field of this entry is non-zero,
;  1081          1608  4              ! we must perform a further check to determine if the
;  1082          1609  4              ! entry we have selected is the one we want.  If the
;  1083          1610  4              ! variable indicated by the test-state setting is not
;  1084          1611  4              ! set, we advance the table pointer to the next entry.
;  1085          1612  4              !-
;  1086          1613  4
;  1087          1614  5              (CASE .TABLE[TAB_TST_STATE] FROM MIN_STATE TO MAX_STATE OF
;  1088          1615  5                SET
;  1089          1616  5                  [NULL]:                         ! No special action
;  1090          1617  5                    1;
;  1091          1618  5                  [OPTF]:                         ! If set, then an optional file
;  1092          1619  5                    .RAB[COB$V_CTX_OPT];
;  1093          1620  5                  [OFNP]:                         !\If set, then optional
;  1094          1621  5                    .RAB[COB$V_CTX_OFNF];         !/file not present
;  1095          1622  5                  [NNVR]:                         !\If set, then no next
;  1096          1623  5                    .RAB[COB$V_CTX_NNVR];         !/valid record (EOF detected)
;  1097          1624  5                TES)
;  1098          1625  4          DO
;  1099          1626  4
;  1100          1627  4              !+
;  1101          1628  4              ! Incr the address of the table to point to the
;  1102          1629  4              ! next table entry if the TAB_TST_STATE field of
;  1103          1630  4              ! the entry was not set.
;  1104          1631  4              !-
;  1105          1632  4
;  1106          1633  5                  BEGIN
;  1107          1634  5                  TABLE = .TABLE + TAB_S_ENTRY;
;  1108          1635  4                  END;
;  1109          1636  4
;  1110          1637  4              !+
;  1111          1638  4              ! Extract the file status code, action code and
;  1112          1639  4              ! error message number for this error situation.
;  1113          1640  4              !-
;  1114          1641  4
;  1115          1642  4              FILESTAT = .TABLE[TAB_FILESTAT];        ! Load COBOL file status
;  1116          1643  4
;  1117          1644  4              !+
;  1118          1645  4              !  1-038 code.
;  1119          1646  4              !  If fifth parameter present then change the file status
;  1120          1647  4              !  values of 13, 15, and 16 to 10.  This change affects three
;  1121          1648  4              !  table entries, REA_S_S_T, REA_R_S_T, and REA_I_S_T.  At this
;  1122          1649  4              !  time it was not necessary to check that the change was made
;  1123          1650  4              !  on these three entries, but perhaps in the future this
;  1124          1651  4              !  additional check will be needed.
;  1125          1652  4              !-
;  1126          1653  4
;  1127          1654  4              IF ACTUALCOUNT() GEQ 5 THEN
;  1128          1655  4                  IF .ANSI74 THEN
;  1129          1656  5                      IF (.FILESTAT EQL '13' OR
;  1130          1657  5                          .FILESTAT EQL '15' OR
;  1131          1658  5                          .FILESTAT EQL '16')
```

```
;  1132          1659  4                    THEN FILESTAT = '10';
;  1133          1660  4
;  1134          1661  4                ACTION = .TABLE[TAB_ACTION];            ! Load type of recovery action
;  1135          1662  4                ERR_MSG_NUM = (.TABLE[TAB_ERR_NO]^3) + COB$_NO_USEPRO;
;  1136          1663  4
;  1137          1664  4
;  1138          1665  4                !+
;  1139          1666  4                ! Check to see if we need to set some state bit in
;  1140          1667  4                ! conjunction with this error situation -- signaled
;  1141          1668  4                ! by a non-zero entry in .TABLE[TAB_SET_STATE]
;  1142          1669  4                !-
;  1143          1670  4
;  1144          1671  4                CASE .TABLE[TAB_SET_STATE] FROM MIN_STATE TO MAX_STATE OF
;  1145          1672  4                    SET
;  1146          1673  4                    [NULL]:                             ! Null clause
;  1147          1674  4                        0;
;  1148          1675  4                    [OPTF]:                             ! Null clause
;  1149          1676  4                        0;
;  1150          1677  4                    [OFNP]:                             !\Set state in COBOL file
;  1151          1678  4                        RAB[COB$V_CTX_OFNF] = 1 ;       ! context area indicating
;  1152          1679  4                                                        !/optional file not present
;  1153          1680  4                    [NNVR]:                             !\Set state in COBOL file
;  1154          1681  4                        RAB[COB$V_CTX_NNVR] = 1 ;       ! context area indicating
;  1155          1682  4                                                        !/NNVR (EOF detected).
;  1156          1683  4                    TES;
;  1157          1684  3                END;
;  1158          1685  3
;  1159          1686  2            END;
;  1160          1687  2
;  1161          1688  2            !+
;  1162          1689  2            ! Additional error types.
;  1163          1690  2            !-
;  1164          1691  2
;  1165          1692  2            [COB$K_EXC_FLK]:
;  1166          1693  3                BEGIN
;  1167          1694  3
;  1168          1695  3                !+
;  1169          1696  3                ! Compiled code detected an OPEN directed to a file closed
;  1170          1697  3                ! WITH LOCK.
;  1171          1698  3                !-
;  1172          1699  3
;  1173          1700  3                ERR_MSG_NUM = COB$_FILCLOLOC ;
;  1174          1701  3                RMS_STS = 0;                                !\Init RMS status code and
;  1175          1702  3                RMS_STV = 0;                                !/status value from FAB/RAB to zero
;  1176          1703  3                FILESTAT = '94';
;  1177          1704  3                ACTION = ABORT;
;  1178          1705  3                RAB[RAB$L_STS] = .ERR_MSG_NUM;
;  1179          1706  3                RAB[RAB$L_STV] = 0;
;  1180          1707  3                END;
;  1181          1708  2
;  1182          1709  2
;  1183          1710  2            [COB$K_EXC_OPN]:
;  1184          1711  3                BEGIN
;  1185          1712  3
;  1186          1713  3                !+
;  1187          1714  3                ! Compiled code detected an OPEN directed to a file
;  1188          1715  3                ! already open.
```

```
;  1189          1716  ;              !-
;  1190          1717  ;
;  1191          1718  ;              ERR_MSG_NUM = COB$_FILALROPE ;
;  1192          1719  ;              RMS_STS = 0;                            !\Set RMS status code and
;  1193          1720  ;              RMS_STV = 0;                            !/status value from FAB/RAB to zero
;  1194          1721  ;              FILESTAT = '94';
;  1195          1722  ;              ACTION = ABORT;
;  1196          1723  ;              RAB[RAB$L_STS] = .ERR_MSG_NUM;
;  1197          1724  ;              RAB[RAB$L_STV] = 0;
;  1198          1725  ;              END;
;  1199          1726  ;
;  1200          1727  ;          [COB$K_EXC_ORG]:
;  1201          1728  ;              BEGIN
;  1202          1729  ;
;  1203          1730  ;              !+
;  1204          1731  ;              ! Compiled code detected an OPEN to a file whose organization
;  1205          1732  ;              ! does not match the access mode specified in OPEN
;  1206          1733  ;              !-
;  1207          1734  ;
;  1208          1735  ;              ERR_MSG_NUM = COB$_ORGNOTMAT ;
;  1209          1736  ;              RMS_STS = 0;                            !\Set RMS status code and
;  1210          1737  ;              RMS_STV = 0;                            !/status value from FAB/RAB to zero
;  1211          1738  ;              FILESTAT = '94';
;  1212          1739  ;              ACTION = ABORT;
;  1213          1740  ;              RAB[RAB$L_STS] = .ERR_MSG_NUM;
;  1214          1741  ;              RAB[RAB$L_STV] = 0;
;  1215          1742  ;              END;
;  1216          1743  ;
;  1217          1744  ;
;  1218          1745  ;          [COB$K_EXC_MIN]:
;  1219          1746  ;              BEGIN
;  1220          1747  ;
;  1221          1748  ;              !+
;  1222          1749  ;              ! Compiled code detected an operation on a variable length record that
;  1223          1750  ;              ! is smaller than the minimum allowed.
;  1224          1751  ;              !-
;  1225          1752  ;
;  1226          1753  ;              !+
;  1227          1754  ;              ! Determine the I/O statement type that failed and select
;  1228          1755  ;              ! the appropriate message.  Note that COB$K_EXC_MINS and
;  1229          1756  ;              ! COB$K_EXC_MAXS refer to the minimum (1) and maximum (44)
;  1230          1757  ;              ! CASE values for I/O statement types.
;  1231          1758  ;              !-
;  1232          1759  ;
;  1233          1760  ;              CASE .FLAGS[COB$W_EXC_STMT]
;  1234          1761  ;                  FROM COB$K_EXC_MINS TO COB$K_EXC_MAXS OF
;  1235          1762  ;                  SET
;  1236          1763  ;
;  1237          1764  ;                  !+
;  1238          1765  ;                  ! Class of read errors, varying in file organization
;  1239          1766  ;                  ! and access type.  For this type of I/O error, return
;  1240          1767  ;                  ! an error message signifying attempting a read of
;  1241          1768  ;                  ! variable length smaller than the minimum allowed.
;  1242          1769  ;                  !-
;  1243          1770  ;
;  1244          1771  ;                  [COB$K_EXC_REASS,
;  1245          1772  ;                   COB$K_EXC_REARS,
```

```
;  1246          1773  3                COB$K_EXC_REARR,
;  1247          1774  3                COB$K_EXC_REAIS,
;  1248          1775  3                COB$K_EXC_REAIR]:
;  1249          1776  3                   ERR_MSG_NUM = COB$_REASMAMIN ;
;  1250          1777  3
;  1251          1778  3             !+
;  1252          1779  3             ! Class of write errors, varying in file organization
;  1253          1780  3             ! and access type.  For this type of I/O error, return
;  1254          1781  3             ! an error message signifying attempting a write of
;  1255          1782  3             ! variable length smaller than the minimum allowed.
;  1256          1783  3             !-
;  1257          1784  3
;  1258          1785  3             [COB$K_EXC_WRISS,
;  1259          1786  3              COB$K_EXC_WRIRS,
;  1260          1787  3              COB$K_EXC_WRIRR,
;  1261          1788  3              COB$K_EXC_WRIIS,
;  1262          1789  3              COB$K_EXC_WRIIR]:
;  1263          1790  3                 ERR_MSG_NUM = COB$_WRISMAMIN ;
;  1264          1791  3
;  1265          1792  3             !+
;  1266          1793  3             ! Class of re-write errors, varying in file organization
;  1267          1794  3             ! and access type.  For this type of I/O error, return
;  1268          1795  3             ! an error message signifying attempting a re-write of
;  1269          1796  3             ! variable length smaller than the minimum allowed.
;  1270          1797  3             !-
;  1271          1798  3
;  1272          1799  3             [COB$K_EXC_REWSS,
;  1273          1800  3              COB$K_EXC_REWRS,
;  1274          1801  3              COB$K_EXC_REWRR,
;  1275          1802  3              COB$K_EXC_REWIS,
;  1276          1803  3              COB$K_EXC_REWIR]:
;  1277          1804  3                 ERR_MSG_NUM = COB$_REWSMAMIN ;
;  1278          1805  3
;  1279          1806  3             [INRANGE]:                          ! Invalid error statement
;  1280          1807  4                 BEGIN
;  1281          1808  4                 SIGNAL_STOP (COB$_INVARG) ;
;  1282          1809  4                 RETURN 0;
;  1283          1810  3                 END;
;  1284          1811  3             TES;
;  1285          1812  3
;  1286          1813  3          RMS_STS = 0;                                   !\Set RMS status code and
;  1287          1814  3          RMS_STV = 0;                                   !/status value from FAB/RAB to zero
;  1288          1815  3          FILESTAT = '94';
;  1289          1816  3          ACTION = ABORT;
;  1290          1817  3          RAB[RAB$L_STS] = .ERR_MSG_NUM;
;  1291          1818  3          RAB[RAB$L_STV] = 0;
;  1292          1819  2          END;
;  1293          1820  2
;  1294          1821  2
;  1295          1822  2       [COB$K_EXC_KEY]:
;  1296          1823  3          BEGIN
;  1297          1824  3
;  1298          1825  3          !+
;  1299          1826  3          ! Compiled code detected an OPEN of an indexed file with keys
;  1300          1827  3          ! that do not match those described by the program.
;  1301          1828  3          !-
;  1302          1829  3
```

```
;  1303          1830  3            ERR_MSG_NUM = COB$_KEYNOTMAT ;
;  1304          1831  3            RMS_STS = 0;                            !\Set RMS status code and
;  1305          1832  3            RMS_STV = 0;                            !/status value from FAB/RAB to zero
;  1306          1833  3            FILESTAT = '94';
;  1307          1834  3            ACTION = ABORT;
;  1308          1835  3            RAB[RAB$L_STS] = .ERR_MSG_NUM;
;  1309          1836  3            RAB[RAB$L_STV] = 0;
;  1310          1837  3            END;
;  1311          1838  2
;  1312          1839  2
;  1313          1840  2        [COB$K_EXC_PIO]:
;  1314          1841  3            BEGIN
;  1315          1842  3
;  1316          1843  3            !+
;  1317          1844  3            ! Compiled code detected a DELETE or REWRITE which was not preceded
;  1318          1845  3            ! by a successful READ while in sequential access mode.
;  1319          1846  3            !-
;  1320          1847  3
;  1321          1848  3            ERR_MSG_NUM = COB$_REAMP_D_R ;
;  1322          1849  3            RMS_STS = 0;                            !\Set RMS status code and
;  1323          1850  3            RMS_STV = 0;                            !/status value from FAB/RAB to zero
;  1324          1851  3            FILESTAT = '93';
;  1325          1852  3            ACTION = ABORT;
;  1326          1853  3            RAB[RAB$L_STS] = .ERR_MSG_NUM;
;  1327          1854  3            RAB[RAB$L_STV] = 0;
;  1328          1855  3            END;
;  1329          1856  2
;  1330          1857  2        TES;                                        ! End of case on error type
;  1331          1858  2
;  1332          1859  2
;  1333          1860  2 !+
;  1334          1861  2 !       At this point in the code, the following have been
;  1335          1862  2 !       computed:
;  1336          1863  2 !
;  1337          1864  2 !               RMS_STS         -- RMS status extracted from FAB/RAB
;  1338          1865  2 !
;  1339          1866  2 !               RMS_STV         -- Extended RMS status extracted from
;  1340          1867  2 !                                  FAB/RAB
;  1341          1868  2 !
;  1342          1869  2 !               FILESTAT        -- COBOL file status to be returned to
;  1343          1870  2 !                                  caller via optional argument.
;  1344          1871  2 !
;  1345          1872  2 !               ACTION          -- Subsequent action
;  1346          1873  2 !                                  Possible settings are:
;  1347          1874  2 !                                       ABORT
;  1348          1875  2 !                                       GOTO
;  1349          1876  2 !                                       CONTINUE
;  1350          1877  2 !                                       SUCCEED
;  1351          1878  2 !                                  (See definitions in LITERAL
;  1352          1879  2 !                                   declarations)
;  1353          1880  2 !
;  1354          1881  2 !               ERR_MSG_NUM     -- COB$ facility error condition code
;  1355          1882  2 !                                  to be signalled if there is no USE
;  1356          1883  2 !                                  procedure available.
;  1357          1884  2 !-
;  1358          1885  2
;  1359          1886  2
```

```
;  1360          1887  2      !+
;  1361          1888  2      ! Store the COBOL file status if appropriate.
;  1362          1889  2      ! Note that "STATUS" is an optional input parameter
;  1363          1890  2      ! that is used to return the COBOL file status to
;  1364          1891  2      ! the caller.
;  1365          1892  2      !-
;  1366          1893  2
;  1367          1894  2      IF ACTUALCOUNT() GEQ 4 THEN IF .STATUS NEQ 0
;  1368          1895  2        THEN
;  1369          1896  2          (.STATUS)<0,16> = .FILESTAT;
;  1370          1897  2
;  1371          1898  2
;  1372          1899  2      !+
;  1373          1900  2      ! If the error is one that requires transfer to the exception label and
;  1374          1901  2      ! it is present, go there by replacing the return PC in the stack
;  1375          1902  2      ! frame with the exception PC and executing a return.
;  1376          1903  2      !-
;  1377          1904  2
;  1378          1905  2      IF .ACTION EQL GOTO THEN IF ACTUALCOUNT() GEQ 3 THEN IF .EXCLAB NEQ 0
;  1379          1906  2        THEN
;  1380          1907  3          BEGIN
;  1381          1908  3          FP[SF$L_SAVE_PC] = .EXCLAB;
;  1382          1909  3          RETURN 0;
;  1383          1910  2          END;
;  1384          1911  2
;  1385          1912  2      !+
;  1386          1913  2      ! If the action is SUCCEED, return to the I/O statement.
;  1387          1914  2      !-
;  1388          1915  2
;  1389          1916  2      IF .ACTION EQL SUCCEED
;  1390          1917  2        THEN
;  1391          1918  2          RETURN 1;
;  1392          1919  2
;  1393          1920  2      !+
;  1394          1921  2      ! Search for an appropriate USE procedure.
;  1395          1922  2      !-
;  1396          1923  2
;  1397          1924  2      SFP = .FP[SF$L_SAVE_FP];                                  ! Get saved FP
;  1398          1925  2      IF .SFP EQL 0                                             !\If frame pointer is
;  1399          1926  2        THEN                                                    ! zero, then we have
;  1400          1927  3          BEGIN                                                 !/serious problems
;  1401          1928  3          SIGNAL_STOP (OTS$_FATINTERR);
;  1402          1929  3          RETURN 0;
;  1403          1930  3          END
;  1404          1931  2        ELSE
;  1405          1932  3              BEGIN
;  1406          1933  3              USE = .SFP[COB$A_SF_USE];                         ! Get USE list
;  1407          1934  3              IF .USE NEQ 0                                     ! If any DECLARATIVES
;  1408          1935  3                THEN
;  1409          1936  4                  BEGIN
;  1410          1937  4
;  1411          1938  4                  !+
;  1412          1939  4                  ! Search for a USE procedure declared for the specific file
;  1413          1940  4                  ! on which the exception occurred.  It is identified by the
;  1414          1941  4                  ! RAB address.
;  1415          1942  4                  !-
;  1416          1943  4
```

```
;  1417          1944  4          USEENT = USE[COB$A_USE_FILES];          ! Point to first
;  1418          1945  4          DECR I FROM .USE[COB$B_USE_COUNT]-1 TO 0 DO ! Count of file entries
;  1419          1946  5              BEGIN                               ! Loop over files
;  1420          1947  5              IF .USEENT[COB$A_USE_RAB] EQLA .RAB ! Right file? (find matching RAB addrs)
;  1421          1948  5                THEN
;  1422          1949  6                  BEGIN
;  1423          1950  6
;  1424          1951  6                  !+
;  1425          1952  6                  ! The EOPR value will be non-zero if the perform
;  1426          1953  6                  ! range is in the current level of the code.  If
;  1427          1954  6                  ! the EOPR is zero, then the USE procedure can't
;  1428          1955  6                  ! be seen in the contained program environment, and
;  1429          1956  6                  ! therefore, an error must be signalled.  Note that
;  1430          1957  6                  ! the USE procedure may be found, since it is possible
;  1431          1958  6                  ! that the USE procedure was in a different level of
;  1432          1959  6                  ! code (an up-level reference).  If this were the case,
;  1433          1960  6                  ! then this would be resolved in COB$HANDLER (or a
;  1434          1961  6                  ! user-defined handler).
;  1435          1962  6                  !-
;  1436          1963  6
;  1437          1964  6                      IF .USEENT[COB$A_USE_EOPR] NEQ 0 !Check addr of end of
;  1438          1965  6                                                       !perform range block
;  1439          1966  6                        THEN
;  1440          1967  7                          BEGIN
;  1441          1968  7                          COB$$INVOKE_USE(             !\Invoke USE procedure;
;  1442          1969  7                                                       !/USE proc is at current level
;  1443          1970  7                              .USEENT[COB$A_USE_PROC],! Addr of USE procedure
;  1444          1971  7                              .USE,
;  1445          1972  7                              .FP[SF$L_SAVE_AP],
;  1446          1973  7                              .USEENT[COB$A_USE_EOPR],
;  1447          1974  7                              .USE[COB$A_USE_PNC]);
;  1448          1975  7                          IF .ACTION EQL CONTINUE
;  1449          1976  7                            THEN
;  1450          1977  7                              RETURN 1
;  1451          1978  7                            ELSE
;  1452          1979  7                              RETURN 0;
;  1453          1980  7                          END
;  1454          1981  6                        ELSE
;  1455          1982  6
;  1456          1983  6                      !+
;  1457          1984  6                      ! Save the address of the file specific USE
;  1458          1985  6                      ! procedure for later call to SIGNAL.
;  1459          1986  6                      !-
;  1460          1987  6
;  1461          1988  6                          FILE_ADDR = .USEENT[COB$A_USE_PROC];
;  1462          1989  5                      END;
;  1463          1990  5              USEENT = .USEENT + COB$S_USE_FILES; ! Step to next entry
;  1464          1991  4              END;
;  1465          1992  4
;  1466          1993  4
;  1467          1994  4          !+
;  1468          1995  4          ! Fall into this section of code since no entry for the
;  1469          1996  4          ! specific file was found.  See if a USE procedure has been
;  1470          1997  4          ! declared for the open mode.  Note that the open mode
;  1471          1998  4          ! entries are INPUT, OUTPUT, I-O and EXTEND with corresponding
;  1472          1999  4          ! values of 0,1,2,3.  Also note that the field reference of
;  1473          2000  4          ! [COB$B_CTX_MODE] refers to the open mode entry field.
```

```
;  1474          2001  4                !-
;  1475          2002  4
;  1476          2003  4                IF .RAB[COB$B_CTX_MODE] GTRU COB$K_CTX_MAX
;  1477          2004  4                  THEN
;  1478          2005  5                    BEGIN
;  1479          2006  5                    SIGNAL_STOP(COB$_INVARG);
;  1480          2007  5                    RETURN 0;
;  1481          2008  4                    END;
;  1482          2009  4                USEENT = USE[COB$A_USE_MODES] + COB$S_USE_MODES*.RAB[COB$B_CTX_MODE];
;  1483          2010  4                IF .USEENT[COB$A_USE_PROC] NEQA 0       ! USE declared?
;  1484          2011  4                  THEN
;  1485          2012  5                    BEGIN
;  1486          2013  5
;  1487          2014  5                !+
;  1488          2015  5                ! Check address of the End of Perform Range Block -
;  1489          2016  5                ! if non-zero, then invoke the USE procedure.  Note
;  1490          2017  5                ! that EOPR if non-zero, indicates that the USE procedure
;  1491          2018  5                ! can be invoked immediately since it is in the current
;  1492          2019  5                ! level of code.  Otherwise, an EOPR of zero indicates
;  1493          2020  5                ! that the USE procedure can't be found, which causes
;  1494          2021  5                ! an error to be signalled.  Note that the USE procedure
;  1495          2022  5                ! may be found, since it is possible that the USE procedure
;  1496          2023  5                ! was in a different level (an up-level reference).  This
;  1497          2024  5                ! would get resolved in COB$HANDLER (or a user-defined
;  1498          2025  5                ! handler).
;  1499          2026  5                !-
;  1500          2027  5
;  1501          2028  5                        IF .USEENT[COB$A_USE_EOPR] NEQ 0
;  1502          2029  5                          THEN
;  1503          2030  6                            BEGIN
;  1504          2031  6                            COB$$INVOKE_USE(                    ! Invoke USE
;  1505          2032  6                                .USEENT[COB$A_USE_PROC],        ! Addr of USE procedure
;  1506          2033  6                                .USE,
;  1507          2034  6                                .FP[SF$L_SAVE_AP],
;  1508          2035  6                                .USEENT[COB$A_USE_EOPR],
;  1509          2036  6                                .USE[COB$A_USE_PNC]);
;  1510          2037  6                            IF .ACTION EQL CONTINUE
;  1511          2038  6                              THEN
;  1512          2039  6                                RETURN 1
;  1513          2040  6                              ELSE
;  1514          2041  6                                RETURN 0;
;  1515          2042  6                            END
;  1516          2043  5                          ELSE
;  1517          2044  5
;  1518          2045  5                        !+
;  1519          2046  5                        ! Save the address of the open mode USE
;  1520          2047  5                        ! procedure for later call to SIGNAL.
;  1521          2048  5                        !-
;  1522          2049  5
;  1523          2050  5                            OPEN_MD_ADDR = .USEENT[COB$A_USE_PROC];
;  1524          2051  4                    END;
;  1525          2052  4
;  1526          2053  4                !+
;  1527          2054  4                ! Note that the error that is signalled is
;  1528          2055  4                ! "Lost handler for USE procedure - environment
;  1529          2056  4                ! corrupted" so that if a USE procedure isn't found,
;  1530          2057  4                ! then this message would appear at the user
```

```
;  1531          2058  4                   ! level.
;  1532          2059  4                   !-
;  1533          2060  4
;  1534          2061  4                   IF .FILE_ADDR NEQ 0 OR .OPEN_MD_ADDR NEQ 0
;  1535          2062  4                     THEN
;  1536          2063  5                       BEGIN
;  1537          2064  5                       SIGNAL (COB$_LSTHNDUSE,2,.FILE_ADDR,.OPEN_MD_ADDR);
;  1538          2065  5                       IF .ACTION EQL CONTINUE
;  1539          2066  5                         THEN
;  1540          2067  5                           RETURN 1
;  1541          2068  5                         ELSE
;  1542          2069  5                           RETURN 0 ;                          ! All done; assume ABORT case
;  1543          2070  4                       END;
;  1544          2071  3                   END;
;  1545          2072  2             END;
;  1546          2073  2
;  1547          2074  2         !+
;  1548          2075  2         ! Return a 1 for CONTINUE action if a USE procedure
;  1549          2076  2         ! has not been found.
;  1550          2077  2         !-
;  1551          2078  2
;  1552          2079  2         IF .ACTION EQL CONTINUE
;  1553          2080  2           THEN
;  1554          2081  2             RETURN 1;
;  1555          2082  2
;  1556          2083  2         !+
;  1557          2084  2         ! Signal the error, since no USE procedure was applicable.
;  1558          2085  2         ! (GOTO with no exception label and ABORT cases)
;  1559          2086  2         !-
;  1560          2087  2
;  1561          2088  2         SIGNAL_STOP(
;  1562          2089  2                     .ERR_MSG_NUM, 1, RSADESC,
;  1563          2090  2                     .RMS_STS,
;  1564          2091  2                     .RMS_STV);
;  1565          2092  2
;  1566          2093  2         RETURN 0                                              ! Never gets here...
;  1567          2094  1         END;


                                        .TITLE  COB$IOEXCEPTION
                                        .IDENT  \1-039\

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

                              00000 P.AAA:  .BLKB   0
                00018544      00000 P.AAB:  .LONG   99652
                  35  39        00004       .ASCII  \95\
                      00        00006       .BYTE   0
                      00*       00007       .BYTE   <<COB$_NO_SPACE-COB$_NO_USEPRO>@-3>
                      00        00008       .BYTE   0
                      00        00009       .BYTE   0
                00018292      0000A         .LONG   98962
                  35  30        0000E       .ASCII  \05\
                      02        00010       .BYTE   2
                      00*       00011       .BYTE   <<COB$_OPTMISOPE-COB$_NO_USEPRO>@-3>
                      01        00012       .BYTE   1
                      02        00013       .BYTE   2
```

```
00018292  00014          .LONG   98962
  37  39  00018          .ASCII  \97\
      00  0001A          .BYTE   0
      00* 0001B          .BYTE   <<COB$_FILNOTFOU-COB$_NO_USEPRO>@-3>
      00  0001C          .BYTE   0
      00  0001D          .BYTE   0
0001828A  0001E          .LONG   98954
  31  39  00022          .ASCII  \91\
      00  00024          .BYTE   0
      00* 00025          .BYTE   <<COB$_FILALRLOC-COB$_NO_USEPRO>@-3>
      00  00026          .BYTE   0
      00  00027          .BYTE   0
00000000  00028          .LONG   0
  30  33  0002C          .ASCII  \30\
      00  0002E          .BYTE   0
      00* 0002F          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  00030          .BYTE   0
      00  00031          .BYTE   0
00018564  00032 P.AAC:   .LONG   99684
  30  30  00036          .ASCII  \00\
      03  00038          .BYTE   3
      00* 00039          .BYTE   <<COB$_OPTMISCLO-COB$_NO_USEPRO>@-3>
      02  0003A          .BYTE   2
      00  0003B          .BYTE   0
00018564  0003C          .LONG   99684
  34  39  00040          .ASCII  \94\
      00  00042          .BYTE   0
      00* 00043          .BYTE   <<COB$_FILALRCLO-COB$_NO_USEPRO>@-3>
      00  00044          .BYTE   0
      00  00045          .BYTE   0
0001850C  00046          .LONG   99596
  34  39  0004A          .ASCII  \94\
      00  0004C          .BYTE   0
      00* 0004D          .BYTE   <<COB$_FILALRCLO-COB$_NO_USEPRO>@-3>
      00  0004E          .BYTE   0
      00  0004F          .BYTE   0
00000000  00050          .LONG   0
  38  39  00054          .ASCII  \98\
      00  00056          .BYTE   0
      00* 00057          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  00058          .BYTE   0
      00  00059          .BYTE   0
00018584  0005A P.AAD:   .LONG   99716
  30  30  0005E          .ASCII  \00\
      03  00060          .BYTE   3
      00* 00061          .BYTE   <<COB$_OPTMISCLO-COB$_NO_USEPRO>@-3>
      02  00062          .BYTE   2
      00  00063          .BYTE   0
00018584  00064          .LONG   99716
  34  39  00068          .ASCII  \94\
      00  0006A          .BYTE   0
      00* 0006B          .BYTE   <<COB$_FILALRCLO-COB$_NO_USEPRO>@-3>
      00  0006C          .BYTE   0
      00  0006D          .BYTE   0
0001863C  0006E          .LONG   99900
  34  39  00072          .ASCII  \94\
      00  00074          .BYTE   0
```

```
                00*  00075          .BYTE   <<COB$_FILALRCLO-COB$_NO_USEPRO>@-3>
                00   00076          .BYTE   0
                00   00077          .BYTE   0
00018574        00078          .LONG   99700
  30  30        0007C          .ASCII  \00\
                03   0007E          .BYTE   3
                00*  0007F          .BYTE   <<COB$_NO_USEPRO-COB$_NO_USEPRO>@-3>
                00   00080          .BYTE   0
                00   00081          .BYTE   0
0001827A        00082          .LONG   98938
  30  30        00086          .ASCII  \00\
                03   00088          .BYTE   3
                00*  00089          .BYTE   <<COB$_NO_USEPRO-COB$_NO_USEPRO>@-3>
                00   0008A          .BYTE   0
                00   0008B          .BYTE   0
00000000        0008C          .LONG   0
  38  39        00090          .ASCII  \98\
                00   00092          .BYTE   0
                00*  00093          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
                00   00094          .BYTE   0
                00   00095          .BYTE   0
0001827A        00096  P.AAE:   .LONG   98938
  36  31        0009A          .ASCII  \16\
                01   0009C          .BYTE   1
                00*  0009D          .BYTE   <<COB$_NO_NEXVAL-COB$_NO_USEPRO>@-3>
                03   0009E          .BYTE   3
                00   0009F          .BYTE   0
0001827A        000A0          .LONG   98938
  33  31        000A4          .ASCII  \13\
                01   000A6          .BYTE   1
                00*  000A7          .BYTE   <<COB$_NO_NEXLOG-COB$_NO_USEPRO>@-3>
                00   000A8          .BYTE   0
                03   000A9          .BYTE   3
000182AA        000AA          .LONG   98986
  32  39        000AE          .ASCII  \92\
                00   000B0          .BYTE   0
                00*  000B1          .BYTE   <<COB$_RECLOCREA-COB$_NO_USEPRO>@-3>
                00   000B2          .BYTE   0
                00   000B3          .BYTE   0
00018584        000B4          .LONG   99716
  35  31        000B8          .ASCII  \15\
                01   000BA          .BYTE   1
                00*  000BB          .BYTE   <<COB$_OPTMISREA-COB$_NO_USEPRO>@-3>
                02   000BC          .BYTE   2
                00   000BD          .BYTE   0
00018584        000BE          .LONG   99716
  34  39        000C2          .ASCII  \94\
                00   000C4          .BYTE   0
                00*  000C5          .BYTE   <<COB$_REAUNOFIL-COB$_NO_USEPRO>@-3>
                00   000C6          .BYTE   0
                00   000C7          .BYTE   0
0001863C        000C8          .LONG   99900
  34  39        000CC          .ASCII  \94\
                00   000CE          .BYTE   0
                00*  000CF          .BYTE   <<COB$_REAUNOFIL-COB$_NO_USEPRO>@-3>
                00   000D0          .BYTE   0
                00   000D1          .BYTE   0
```

```
00018514  000D2          .LONG   99604
  34  39  000D6          .ASCII  \94\
      00  000D8          .BYTE   0
      00* 000D9          .BYTE   <<COB$_REAINCOPE-COB$_NO_USEPRO>@-3>
      00  000DA          .BYTE   0
      00  000DB          .BYTE   0
00000000  000DC          .LONG   0
  30  33  000E0          .ASCII  \30\
      00  000E2          .BYTE   0
      00* 000E3          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  000E4          .BYTE   0
      00  000E5          .BYTE   0
000182B2  000E6 P.AAF:   .LONG   98994
  33  32  000EA          .ASCII  \23\
      01  000EC          .BYTE   1
      00* 000ED          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00  000EE          .BYTE   0
      00  000EF          .BYTE   0
000185CC  000F0          .LONG   99788
  33  32  000F4          .ASCII  \23\
      01  000F6          .BYTE   1
      00* 000F7          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00  000F8          .BYTE   0
      00  000F9          .BYTE   0
00018594  000FA          .LONG   99732
  33  32  000FE          .ASCII  \23\
      01  00100          .BYTE   1
      00* 00101          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00  00102          .BYTE   0
      00  00103          .BYTE   0
0001858C  00104          .LONG   99724
  33  32  00108          .ASCII  \23\
      01  0010A          .BYTE   1
      00* 0010B          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  0010C          .BYTE   0
      00  0010D          .BYTE   0
000182AA  0010E          .LONG   98986
  32  39  00112          .ASCII  \92\
      00  00114          .BYTE   0
      00* 00115          .BYTE   <<COB$_RECLOCREA-COB$_NO_USEPRO>@-3>
      00  00116          .BYTE   0
      00  00117          .BYTE   0
00018584  0011:8         .LONG   99716
  35  32  0011C          .ASCII  \25\
      01  0011E          .BYTE   1
      00* 0011F          .BYTE   <<COB$_OPTMISREA-COB$_NO_USEPRO>@-3>
      02  00120          .BYTE   2
      00  00121          .BYTE   0
00018584  00122          .LONG   99716
  34  39  00126          .ASCII  \94\
      00  00128          .BYTE   0
      00* 00129          .BYTE   <<COB$_REAUNOFIL-COB$_NO_USEPRO>@-3>
      00  0012A          .BYTE   0
      00  0012B          .BYTE   0
0001863C  0012C          .LONG   99900
  34  39  00130          .ASCII  \94\
      00  00132          .BYTE   0
```

```
      00*  00133          .BYTE   <<COB$_REAUNOFIL-COB$_NO_USEPRO>@-3>
      00   00134          .BYTE   0
      00   00135          .BYTE   0
00018514   00136          .LONG   99604
  34  39   00139          .ASCII  \94\
      00   0013C          .BYTE   0
      00*  0013D          .BYTE   <<COB$_REAINCOPE-COB$_NO_USEPRO>@-3>
      00   0013E          .BYTE   0
      00   0013F          .BYTE   0
00000000   00140          .LONG   0
  30  33   00144          .ASCII  \30\
      00   00146          .BYTE   0
      00*  00147          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00   00148          .BYTE   0
      00   00149          .BYTE   0
000182B2   0014A P.AAG:    .LONG   98994
  33  32   0014E          .ASCII  \23\
      01   00150          .BYTE   1
      00*  00151          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00   00152          .BYTE   0
      00   00153          .BYTE   0
000182AA   00154          .LONG   98986
  32  39   00158          .ASCII  \92\
      00   0015A          .BYTE   0
      00*  0015B          .BYTE   <<COB$_RECLOCREA-COB$_NO_USEPRO>@-3>
      00   0015C          .BYTE   0
      00   0015D          .BYTE   0
00018584   0015E          .LONG   99716
  35  32   00162          .ASCII  \25\
      01   00164          .BYTE   1
      00*  00165          .BYTE   <<COB$_OPTMISREA-COB$_NO_USEPRO>@-3>
      02   00166          .BYTE   2
      00   00167          .BYTE   0
00018584   00168          .LONG   99716
  34  39   0016C          .ASCII  \94\
      00   0016E          .BYTE   0
      00*  0016F          .BYTE   <<COB$_REAUNOFIL-COB$_NO_USEPRO>@-3>
      00   00170          .BYTE   0
      00   00171          .BYTE   0
0001863C   00172          .LONG   99900
  34  39   00176          .ASCII  \94\
      00   00178          .BYTE   0
      00*  00179          .BYTE   <<COB$_REAUNOFIL-COB$_NO_USEPRO>@-3>
      00   0017A          .BYTE   0
      00   0017B          .BYTE   0
00018514   0017C          .LONG   99604
  34  39   00180          .ASCII  \94\
      00   00182          .BYTE   0
      00*  00183          .BYTE   <<COB$_REAINCOPE-COB$_NO_USEPRO>@-3>
      00   00184          .BYTE   0
      00   00185          .BYTE   0
00000000   00186          .LONG   0
  30  33   0018A          .ASCII  \30\
      00   0018C          .BYTE   0
      00*  0018D          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00   0018E          .BYTE   0
      00   0018F          .BYTE   0
```

```
00018544  00190 P.AAH:  .LONG   99652
  34  33  00194         .ASCII  \34\
      00  00196         .BYTE   0
      00* 00197         .BYTE   <<COB$_WRIBEYBOU-COB$_NO_USEPRO>@-3>
      00  00198         .BYTE   0
      00  00199         .BYTE   0
00018584  0019A         .LONG   99716
  34  39  0019E         .ASCII  \94\
      00  001A0         .BYTE   0
      00* 001A1         .BYTE   <<COB$_WRIUNOFIL-COB$_NO_USEPRO>@-3>
      00  001A2         .BYTE   0
      00  001A3         .BYTE   0
0001863C  001A4         .LONG   99900
  34  39  001A8         .ASCII  \94\
      00  001AA         .BYTE   0
      00* 001AB         .BYTE   <<COB$_WRIUNOFIL-COB$_NO_USEPRO>@-3>
      00  001AC         .BYTE   0
      00  001AD         .BYTE   0
00018514  001AE         .LONG   99604
  34  39  001B2         .ASCII  \94\
      00  001B4         .BYTE   0
      00* 001B5         .BYTE   <<COB$_WRIINCOPE-COB$_NO_USEPRO>@-3>
      00  001B6         .BYTE   0
      00  001B7         .BYTE   0
00000000  001B8         .LONG   0
  30  33  001BC         .ASCII  \30\
      00  001BE         .BYTE   0
      00* 001BF         .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  001C0         .BYTE   0
      00  001C1         .BYTE   0
00018544  001C2 P.AAI:  .LONG   99652
  34  32  001C6         .ASCII  \24\
      01  001C8         .BYTE   1
      00* 001C9         .BYTE   <<COB$_WRIBEYBOU-COB$_NO_USEPRO>@-3>
      00  001CA         .BYTE   0
      00  001CB         .BYTE   0
000182AA  001CC         .LONG   98986
  32  39  001D0         .ASCII  \92\
      00  001D2         .BYTE   0
      00* 001D3         .BYTE   <<COB$_RECLOCWRI-COB$_NO_USEPRO>@-3>
      00  001D4         .BYTE   0
      00  001D5         .BYTE   0
00018584  001D6         .LONG   99716
  34  39  001DA         .ASCII  \94\
      00  001DC         .BYTE   0
      00* 001DD         .BYTE   <<COB$_WRIUNOFIL-COB$_NO_USEPRO>@-3>
      00  001DE         .BYTE   0
      00  001DF         .BYTE   0
0001863C  001E0         .LONG   99900
  34  39  001E4         .ASCII  \94\
      00  001E6         .BYTE   0
      00* 001E7         .BYTE   <<COB$_WRIUNOFIL-COB$_NO_USEPRO>@-3>
      00  001E8         .BYTE   0
      00  001E9         .BYTE   0
00018514  001EA         .LONG   99604
  34  39  001EE         .ASCII  \94\
      00  001F0         .BYTE   0
```

```
      00*  001F1          .BYTE   <<COB$_WRIINCOPE-COB$_NO_USEPRO>@-3>
      00   001F2          .BYTE   0
      00   001F3          .BYTE   0
00000000   001F4          .LONG   0
  30  33   001F8          .ASCII  \30\
      00   001FA          .BYTE   0
      00*  001FB          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00   001FC          .BYTE   0
      00   001FD          .BYTE   0
000182A2   001FE P.AAJ:   .LONG   98978
  32  32   00202          .ASCII  \22\
      01   00204          .BYTE   1
      00*  00205          .BYTE   <<COB$_WRIDUPKEY-COB$_NO_USEPRO>@-3>
      00   00206          .BYTE   0
      00   00207          .BYTE   0
00018544   00208          .LONG   99652
  34  32   0020C          .ASCII  \24\
      01   0020E          .BYTE   1
      00*  0020F          .BYTE   <<COB$_WRIBEYBOU-COB$_NO_USEPRO>@-3>
      00   00210          .BYTE   0
      00   00211          .BYTE   0
00018594   00212          .LONG   99732
  34  32   00216          .ASCII  \24\
      01   00218          .BYTE   1
      00*  00219          .BYTE   <<COB$_WRIBEYBOU-COB$_NO_USEPRO>@-3>
      00   0021A          .BYTE   0
      00   0021B          .BYTE   0
000182AA   0021C          .LONG   98986
  32  39   00220          .ASCII  \92\
      00   00222          .BYTE   0
      00*  00223          .BYTE   <<COB$_RECLOCWRI-COB$_NO_USEPRO>@-3>
      00   00224          .BYTE   0
      00   00225          .BYTE   0
00018584   00226          .LONG   99716
  34  39   0022A          .ASCII  \94\
      00   0022C          .BYTE   0
      00*  0022D          .BYTE   <<COB$_WRIUNOFIL-COB$_NO_USEPRO>@-3>
      00   0022E          .BYTE   0
      00   0022F          .BYTE   0
0001863C   00230          .LONG   99900
  34  39   00234          .ASCII  \94\
      00   00236          .BYTE   0
      00*  00237          .BYTE   <<COB$_WRIUNOFIL-COB$_NO_USEPRO>@-3>
      00   00238          .BYTE   0
      00   00239          .BYTE   0
00018514   0023A          .LONG   99604
  34  39   0023E          .ASCII  \94\
      00   00240          .BYTE   0
      00*  00241          .BYTE   <<COB$_WRIINCOPE-COB$_NO_USEPRO>@-3>
      00   00242          .BYTE   0
      00   00243          .BYTE   0
00000000   00244          .LONG   0
  30  33   00248          .ASCII  \30\
      00   0024A          .BYTE   0
      00*  0024B          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00   0024C          .BYTE   0
      00   0024D          .BYTE   0
```

```
00018011  0024E P.AAK:  .LONG   98321
  32  30  00252         .ASCII  \02\
      02  00254         .BYTE   2
      00* 00255         .BYTE   <<COB$_WRICREDUP-COB$_NO_USEPRO>@-3>
      00  00256         .BYTE   0
      00  00257         .BYTE   0
000186AC  00258         .LONG   100012
  31  32  0025C         .ASCII  \21\
      01  0025E         .BYTE   1
      00* 0025F         .BYTE   <<COB$_WRINOTASC-COB$_NO_USEPRO>@-3>
      00  00260         .BYTE   0
      00  00261         .BYTE   0
000184EC  00262         .LONG   99564
  32  32  00266         .ASCII  \22\
      01  00268         .BYTE   1
      00* 00269         .BYTE   <<COB$_WRIDISDUP-COB$_NO_USEPRO>@-3>
      00  0026A         .BYTE   0
      00  0026B         .BYTE   0
00018544  0026C         .LONG   99652
  34  32  00270         .ASCII  \24\
      01  00272         .BYTE   1
      00* 00273         .BYTE   <<COB$_WRIBEYBOU-COB$_NO_USEPRO>@-3>
      00  00274         .BYTE   0
      00  00275         .BYTE   0
000182AA  00276         .LONG   98986
  32  39  0027A         .ASCII  \92\
      00  0027C         .BYTE   0
      00* 0027D         .BYTE   <<COB$_RECLOCWRI-COB$_NO_USEPRO>@-3>
      00  0027E         .BYTE   0
      00  0027F         .BYTE   0
00018584  00280         .LONG   99716
  34  39  00284         .ASCII  \94\
      00  00286         .BYTE   0
      00* 00287         .BYTE   <<COB$_WRIUNOFIL-COB$_NO_USEPRO>@-3>
      00  00288         .BYTE   0
      00  00289         .BYTE   0
0001863C  0028A         .LONG   99900
  34  39  0028E         .ASCII  \94\
      00  00290         .BYTE   0
      00* 00291         .BYTE   <<COB$_WRIUNOFIL-COB$_NO_USEPRO>@-3>
      00  00292         .BYTE   0
      00  00293         .BYTE   0
00018514  00294         .LONG   99604
  34  39  00298         .ASCII  \94\
      00  0029A         .BYTE   0
      00* 0029B         .BYTE   <<COB$_WRIINCOPE-COB$_NO_USEPRO>@-3>
      00  0029C         .BYTE   0
      00  0029D         .BYTE   0
00000000  0029E         .LONG   0
  30  33  002A2         .ASCII  \30\
      00  002A4         .BYTE   0
      00* 002A5         .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  002A6         .BYTE   0
      00  002A7         .BYTE   0
00018011  002A8 P.AAL:  .LONG   98321
  32  30  002AC         .ASCII  \02\
      02  002AE         .BYTE   2
```

```
      00* 002AF          .BYTE   <<COB$_WRICREDUP-COB$_NO_USEPRO>@-3>
      00  002B0          .BYTE   0
      00  002B1          .BYTE   0
000184EC  002B2          .LONG   99564
  32  32  002B6          .ASCII  \22\
      01  002B8          .BYTE   1
      00* 002B9          .BYTE   <<COB$_WRIDISDUP-COB$_NO_USEPRO>@-3>
      00  002BA          .BYTE   0
      00  002BB          .BYTE   0
000182A2  002BC          .LONG   98978
  32  32  002C0          .ASCII  \22\
      01  002C2          .BYTE   1
      00* 002C3          .BYTE   <<COB$_WRIDISDUP-COB$_NO_USEPRO>@-3>
      00  002C4          .BYTE   0
      00  002C5          .BYTE   0
00018544  002C6          .LONG   99652
  34  32  002CA          .ASCII  \24\
      01  002CC          .BYTE   1
      00* 002CD          .BYTE   <<COB$_WRIBEYBOU-COB$_NO_USEPRO>@-3>
      00  002CE          .BYTE   0
      00  002CF          .BYTE   0
000182AA  002D0          .LONG   98986
  32  39  002D4          .ASCII  \92\
      00  002D6          .BYTE   0
      00* 002D7          .BYTE   <<COB$_RECLOCWRI-COB$_NO_USEPRO>@-3>
      00  002D8          .BYTE   0
      00  002D9          .BYTE   0
00018584  002DA          .LONG   99716
  34  39  002DE          .ASCII  \94\
      00  002E0          .BYTE   0
      00* 002E1          .BYTE   <<COB$_WRIUNOFIL-COB$_NO_USEPRO>@-3>
      00  002E2          .BYTE   0
      00  002E3          .BYTE   0
0001863C  002E4          .LONG   99900
  34  39  002E8          .ASCII  \94\
      00  002EA          .BYTE   0
      00* 002EB          .BYTE   <<COB$_WRIUNOFIL-COB$_NO_USEPRO>@-3>
      00  002EC          .BYTE   0
      00  002ED          .BYTE   0
00018514  002EE          .LONG   99604
  34  39  002F2          .ASCII  \94\
      00  002F4          .BYTE   0
      00* 002F5          .BYTE   <<COB$_WRIINCOPE-COB$_NO_USEPRO>@-3>
      00  002F6          .BYTE   0
      00  002F7          .BYTE   0
00000000  002F8          .LONG   0
  30  33  002FC          .ASCII  \30\
      00  002FE          .BYTE   0
      00* 002FF          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  00300          .BYTE   0
      00  00301          .BYTE   0
000182AA  00302 P.AAM:   .LONG   98986
  32  39  00306          .ASCII  \92\
      00  00308          .BYTE   0
      00* 00309          .BYTE   <<COB$_RECLOCDEL-COB$_NO_USEPRO>@-3>
      00  0030A          .BYTE   0
      00  0030B          .BYTE   0
```

```
000184B4  0030C          .LONG   99508
  33  39  00310          .ASCII  \93\
      00  00312          .BYTE   0
      00* 00313          .BYTE   <<COB$_DELNO_R_S-COB$_NO_USEPRO>@-3>
      00  00314          .BYTE   0
      00  00315          .BYTE   0
00018584  00316          .LONG   99716
  34  39  0031A          .ASCII  \94\
      00  0031C          .BYTE   0
      00* 0031D          .BYTE   <<COB$_DELUNOFIL-COB$_NO_USEPRO>@-3>
      00  0031E          .BYTE   0
      00  0031F          .BYTE   0
0001863C  00320          .LONG   99900
  34  39  00324          .ASCII  \94\
      00  00326          .BYTE   0
      00* 00327          .BYTE   <<COB$_DELUNOFIL-COB$_NO_USEPRO>@-3>
      00  00328          .BYTE   0
      00  00329          .BYTE   0
00018514  0032A          .LONG   99604
  34  39  0032E          .ASCII  \94\
      00  00330          .BYTE   0
      00* 00331          .BYTE   <<COB$_DELINCOPE-COB$_NO_USEPRO>@-3>
      00  00332          .BYTE   0
      00  00333          .BYTE   0
00000000  00334          .LONG   0
  30  33  00338          .ASCII  \30\
      00  0033A          .BYTE   0
      00* 0033B          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  0033C          .BYTE   0
      00  0033D          .BYTE   0
000182B2  0033E P.AAN:   .LONG   98994
  33  32  00342          .ASCII  \23\
      01  00344          .BYTE   1
      00* 00345          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00  00346          .BYTE   0
      00  00347          .BYTE   0
000185CC  00348          .LONG   99788
  33  32  0034C          .ASCII  \23\
      01  0034E          .BYTE   1
      00* 0034F          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00  00350          .BYTE   0
      00  00351          .BYTE   0
000182AA  00352          .LONG   98986
  32  39  00356          .ASCII  \92\
      00  00358          .BYTE   0
      00* 00359          .BYTE   <<COB$_RECLOCDEL-COB$_NO_USEPRO>@-3>
      00  0035A          .BYTE   0
      00  0035B          .BYTE   0
00018584  0035C          .LONG   99716
  34  39  00360          .ASCII  \94\
      00  00362          .BYTE   0
      00* 00363          .BYTE   <<COB$_DELUNOFIL-COB$_NO_USEPRO>@-3>
      00  00364          .BYTE   0
      00  00365          .BYTE   0
0001863C  00366          .LONG   99900
  34  39  0036A          .ASCII  \94\
      00  0036C          .BYTE   0
```

```
      00* 0036D          .BYTE   <<COB$_DELUNOFIL-COB$_NO_USEPRO>@-3>
      00  0036E          .BYTE   0
      00  0036F          .BYTE   0
00018514  00370          .LONG   99604
  34  39  00374          .ASCII  \94\
      00  00376          .BYTE   0
      00* 00377          .BYTE   <<COB$_DELINCOPE-COB$_NO_USEPRO>@-3>
      00  00378          .BYTE   0
      00  00379          .BYTE   0
00000000  0037A          .LONG   0
  30  33  0037E          .ASCII  \30\
      00  00380          .BYTE   0
      00* 00381          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  00382          .BYTE   0
      00  00383          .BYTE   0
00018484  00384 P.AAO:    .LONG   99508
  33  39  00388          .ASCII  \93\
      00  0038A          .BYTE   0
      00* 0038B          .BYTE   <<COB$_REWNO_R_S-COB$_NO_USEPRO>@-3>
      00  0038C          .BYTE   0
      00  0038D          .BYTE   0
00018584  0038E          .LONG   99716
  34  39  00392          .ASCII  \94\
      00  00394          .BYTE   0
      00* 00395          .BYTE   <<COB$_REWUNOFIL-COB$_NO_USEPRO>@-3>
      00  00396          .BYTE   0
      00  00397          .BYTE   0
0001863C  00398          .LONG   99900
  34  39  0039C          .ASCII  \94\
      00  0039E          .BYTE   0
      00* 0039F          .BYTE   <<COB$_REWUNOFIL-COB$_NO_USEPRO>@-3>
      00  003A0          .BYTE   0
      00  003A1          .BYTE   0
00018514  003A2          .LONG   99604
  34  39  003A6          .ASCII  \94\
      00  003A8          .BYTE   0
      00* 003A9          .BYTE   <<COB$_REWINCOPE-COB$_NO_USEPRO>@-3>
      00  003AA          .BYTE   0
      00  003AB          .BYTE   0
00000000  003AC          .LONG   0
  30  33  003B0          .ASCII  \30\
      00  003B2          .BYTE   0
      00* 003B3          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  003B4          .BYTE   0
      00  003B5          .BYTE   0
000182AA  003B6 P.AAP:    .LONG   98986
  32  39  003BA          .ASCII  \92\
      00  003BC          .BYTE   0
      00* 003BD          .BYTE   <<COB$_RECLOCREW-COB$_NO_USEPRO>@-3>
      00  003BE          .BYTE   0
      00  003BF          .BYTE   0
000184B4  003C0          .LONG   99508
  33  39  003C4          .ASCII  \93\
      00  003C6          .BYTE   0
      00* 003C7          .BYTE   <<COB$_REWNO_R_S-COB$_NO_USEPRO>@-3>
      00  003C8          .BYTE   0
      00  003C9          .BYTE   0
```

```
00018584  003CA          .LONG   99716
  34  39  003CE          .ASCII  \94\
      00  003D0          .BYTE   0
      00* 003D1          .BYTE   <<COB$_REWUNOFIL-COB$_NO_USEPRO>@-3>
      00  003D2          .BYTE   0
      00  003D3          .BYTE   0
0001863C  003D4          .LONG   99900
  34  39  003D8          .ASCII  \94\
      00  003DA          .BYTE   0
      00* 003DB          .BYTE   <<COB$_REWUNOFIL-COB$_NO_USEPRO>@-3>
      00  003DC          .BYTE   0
      00  003DD          .BYTE   0
00018514  003DE          .LONG   99604
  34  39  003E2          .ASCII  \94\
      00  003E4          .BYTE   0
      00* 003E5          .BYTE   <<COB$_REWINCOPE-COB$_NO_USEPRO>@-3>
      00  003E6          .BYTE   0
      00  003E7          .BYTE   0
00000000  003E8          .LONG   0
  30  33  003EC          .ASCII  \30\
      00  003EE          .BYTE   0
      00* 003EF          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  003F0          .BYTE   0
      00  003F1          .BYTE   0
000182B2  003F2 P.AAQ:   .LONG   98994
  33  32  003F6          .ASCII  \23\
      01  003F8          .BYTE   1
      00* 003F9          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00  003FA          .BYTE   0
      00  003FB          .BYTE   0
000185CC  003FC          .LONG   99788
  33  32  00400          .ASCII  \23\
      01  00402          .BYTE   1
      00* 00403          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00  00404          .BYTE   0
      00  00405          .BYTE   0
000182AA  00406          .LONG   98986
  32  39  0040A          .ASCII  \92\
      00  0040C          .BYTE   0
      00* 0040D          .BYTE   <<COB$_RECLOCREW-COB$_NO_USEPRO>@-3>
      00  0040E          .BYTE   0
      00  0040F          .BYTE   0
00018584  00410          .LONG   99716
  34  39  00414          .ASCII  \94\
      00  00416          .BYTE   0
      00* 00417          .BYTE   <<COB$_REWUNOFIL-COB$_NO_USEPRO>@-3>
      00  00418          .BYTE   0
      00  00419          .BYTE   0
0001863C  0041A          .LONG   99900
  34  39  0041E          .ASCII  \94\
      00  00420          .BYTE   0
      00* 00421          .BYTE   <<COB$_REWUNOFIL-COB$_NO_USEPRO>@-3>
      00  00422          .BYTE   0
      00  00423          .BYTE   0
00018514  00424          .LONG   99604
  34  39  00428          .ASCII  \94\
      00  0042A          .BYTE   0
```

```
      00*  0042B          .BYTE   <<COB$_REWINCOPE-COB$_NO_USEPRO>@-3>
      00   0042C          .BYTE   0
      00   0042D          .BYTE   0
00000000   0042E          .LONG   0
  30  33   00432          .ASCII  \30\
      00   00434          .BYTE   0
      00*  00435          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00   00436          .BYTE   0
      00   00437          .BYTE   0
00018011   00438 P.AAR:   .LONG   98321
  32  30   0043C          .ASCII  \02\
      02   0043E          .BYTE   2
      00*  0043F          .BYTE   <<COB$_REWCREDUP-COB$_NO_USEPRO>@-3>
      00   00440          .BYTE   0
      00   00441          .BYTE   0
0001849C   00442          .LONG   99484
  31  32   00446          .ASCII  \21\
      01   00448          .BYTE   1
      00*  00449          .BYTE   <<COB$_PRIKEYCHA-COB$_NO_USEPRO>@-3>
      00   0044A          .BYTE   0
      00   0044B          .BYTE   0
000184EC   0044C          .LONG   99564
  32  32   00450          .ASCII  \22\
      C1   00452          .BYTE   1
      00*  00453          .BYTE   <<COB$_REWDISDUP-COB$_NO_USEPRO>@-3>
      00   00454          .BYTE   0
      00   00455          .BYTE   0
000182AA   00456          .LONG   98986
  32  39   0045A          .ASCII  \92\
      00   0045C          .BYTE   0
      00*  0045D          .BYTE   <<COB$_RECLOCREW-COB$_NO_USEPRO>@-3>
      00   0045E          .BYTE   0
      00   0045F          .BYTE   0
00018484   00460          .LONG   99508
  33  39   00464          .ASCII  \93\
      00   00466          .BYTE   0
      00*  00467          .BYTE   <<COB$_REWNO_R_S-COB$_NO_USEPRO>@-3>
      00   00468          .BYTE   0
      00   00469          .BYTE   0
00018584   0046A          .LONG   99716
  34  39   0046E          .ASCII  \94\
      00   00470          .BYTE   0
      00*  00471          .BYTE   <<COB$_REWUNOFIL-COB$_NO_USEPRO>@-3>
      00   00472          .BYTE   0
      00   00473          .BYTE   0
0001863C   00474          .LONG   99900
  34  39   00478          .ASCII  \94\
      00   0047A          .BYTE   0
      00*  0047B          .BYTE   <<COB$_REWUNOFIL-COB$_NO_USEPRO>@-3>
      00   0047C          .BYTE   0
      00   0047D          .BYTE   0
00018514   0047E          .LONG   99604
  34  39   00482          .ASCII  \94\
      00   00484          .BYTE   0
      00*  00485          .BYTE   <<COB$_REWINCOPE-COB$_NO_USEPRO>@-3>
      00   00486          .BYTE   0
      00   00487          .BYTE   0
```

```
00000000  00488          .LONG   0
  30  33  0048C          .ASCII  \30\
      00  0048E          .BYTE   0
      00* 0048F          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  00490          .BYTE   0
      00  00491          .BYTE   0
00018011  00492 P.AAS:   .LONG   98321
  32  30  00496          .ASCII  \02\
      02  00498          .BYTE   2
      00* 00499          .BYTE   <<COB$_REWCREDUP-COB$_NO_USEPRO>@-3>
      00  0049A          .BYTE   0
      00  0049B          .BYTE   0
000184EC  0049C          .LONG   99564
  32  32  004A0          .ASCII  \22\
      01  004A2          .BYTE   1
      00* 004A3          .BYTE   <<COB$_REWDISDUP-COB$_NO_USEPRO>@-3>
      00  004A4          .BYTE   0
      00  004A5          .BYTE   0
000182B2  004A6          .LONG   98994
  33  32  004AA          .ASCII  \23\
      01  004AC          .BYTE   1
      00* 004AD          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00  004AE          .BYTE   0
      00  004AF          .BYTE   0
000182AA  004B0          .LONG   98986
  32  39  004B4          .ASCII  \92\
      00  004B6          .BYTE   0
      00* 004B7          .BYTE   <<COB$_RECLOCREW-COB$_NO_USEPRO>@-3>
      00  004B8          .BYTE   0
      00  004B9          .BYTE   0
00018584  004BA          .LONG   99716
  34  39  004BE          .ASCII  \94\
      00  004C0          .BYTE   0
      00* 004C1          .BYTE   <<COB$_REWUNOFIL-COB$_NO_USEPRO>@-3>
      00  004C2          .BYTE   0
      00  004C3          .BYTE   0
0001863C  004C4          .LONG   99900
  34  39  004C8          .ASCII  \94\
      00  004CA          .BYTE   0
      00* 004CB          .BYTE   <<COB$_REWUNOFIL-COB$_NO_USEPRO>@-3>
      00  004CC          .BYTE   0
      00  004CD          .BYTE   0
00018514  004CE          .LONG   99604
  34  39  004D2          .ASCII  \94\
      00  004D4          .BYTE   0
      00* 004D5          .BYTE   <<COB$_REWINCOPE-COB$_NO_USEPRO>@-3>
      00  004D6          .BYTE   0
      00  004D7          .BYTE   0
00000000  004D8          .LONG   0
  30  33  004DC          .ASCII  \30\
      00  004DE          .BYTE   0
      00* 004DF          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  004E0          .BYTE   0
      00  004E1          .BYTE   0
000182B2  004E2 P.AAT:   .LONG   98994
  33  32  004E6          .ASCII  \23\
      01  004E8          .BYTE   1
```

```
      00* 004E9          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00  004EA          .BYTE   0
      03  004EB          .BYTE   3
000185CC  004EC          .LONG   99788
  33  32  004F0          .ASCII  \23\
      01  004F2          .BYTE   1
      00* 004F3          .BYTE   <<COB$_RECNOTEXI-COB$_NO_USEPRO>@-3>
      00  004F4          .BYTE   0
      03  004F5          .BYTE   3
00018584  004F6          .LONG   99716
  35  32  004FA          .ASCII  \25\
      01  004FC          .BYTE   1
      00* 004FD          .BYTE   <<COB$_OPTMISSTA-COB$_NO_USEPRO>@-3>
      02  004FE          .BYTE   2
      00  004FF          .BYTE   0
00018584  00500          .LONG   99716
  34  39  00504          .ASCII  \94\
      00  00506          .BYTE   0
      00* 00507          .BYTE   <<COB$_STAUNOFIL-COB$_NO_USEPRO>@-3>
      00  00508          .BYTE   0
      00  00509          .BYTE   0
0001863C  0050A          .LONG   99900
  34  39  0050E          .ASCII  \94\
      00  00510          .BYTE   0
      00* 00511          .BYTE   <<COB$_STAUNOFIL-COB$_NO_USEPRO>@-3>
      00  00512          .BYTE   0
      00  00513          .BYTE   0
000182AA  00514          .LONG   98986
  32  39  00518          .ASCII  \92\
      00  0051A          .BYTE   0
      00* 0051B          .BYTE   <<COB$_RECLOCSTA-COB$_NO_USEPRO>@-3>
      00  0051C          .BYTE   0
      00  0051D          .BYTE   0
00018514  0051E          .LONG   99604
  34  39  00522          .ASCII  \94\
      00  00524          .BYTE   0
      00* 00525          .BYTE   <<COB$_STAINCOPE-COB$_NO_USEPRO>@-3>
      00  00526          .BYTE   0
      00  00527          .BYTE   0
00000000  00528          .LONG   0
  30  33  0052C          .ASCII  \30\
      00  0052E          .BYTE   0
      00* 0052F          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>
      00  00530          .BYTE   0
      00  00531          .BYTE   0
000181A0  00532 P.AAU:   .LONG   98720
  30  30  00536          .ASCII  \00\
      03  00538          .BYTE   3
      00* 00539          .BYTE   <<COB$_RECNOTLOC-COB$_NO_USEPRO>@-3>
      00  0053A          .BYTE   0
      00  0053B          .BYTE   0
00018584  0053C          .LONG   99716
  34  39  00540          .ASCII  \94\
      00  00542          .BYTE   0
      00* 00543          .BYTE   <<COB$_UNLUNOFIL-COB$_NO_USEPRO>@-3>
      00  00544          .BYTE   0
      00  00545          .BYTE   0
```

```
                                                0001863C  00546          .LONG   99900                                                   ;
                                                  34  39  0054A          .ASCII  \94\                                                    ;
                                                      00  0054C          .BYTE   0                                                       ;
                                                      00* 0054D          .BYTE   <<COB$_UNLUNOFIL-COB$_NO_USEPRO>@-3>                    ;
                                                      00  0054E          .BYTE   0                                                       ;
                                                      00  0054F          .BYTE   0                                                       ;
                                                000184B4  00550          .LONG   99508                                                   ;
                                                  33  39  00554          .ASCII  \93\                                                    ;
                                                      00  00556          .BYTE   0                                                       ;
                                                      00* 00557          .BYTE   <<COB$_UNLNO_CUR-COB$_NO_USEPRO>@-3>                    ;
                                                      00  00558          .BYTE   0                                                       ;
                                                      00  00559          .BYTE   0                                                       ;
                                                00000000  0055A          .LONG   0                                                       ;
                                                  30  33  0055E          .ASCII  \30\                                                    ;
                                                      00  00560          .BYTE   0                                                       ;
                                                      00  00561          .BYTE   <<COB$_ERRON_FIL-COB$_NO_USEPRO>@-3>                    ;
                                                      00  00562          .BYTE   0                                                       ;
                                                      00  00563          .BYTE   0                                                       ;
0032  0032  FFFF  0032  0000  0000  0000  0000  FFFF  0000  00564 P.AAV:  .WORD   0, -1, 0, 0, 0, 0, 50, -1, 50, 50, 50, -                 ;
FFFF  0190  014A  0096  00E6  0096  FFFF  0096  0032  0032  00578                  50, 150, -1, 150, 230, 150, 330, 400, -1, -                ;
0492  0438  03F2  03B6  FFFF  0384  02A8  024E  01FE  01C2  0058C                  450, 510, 590, 680, 900, -1, 950, 1010, -                  ;
FFFF  04E2  FFFF  FFFF  033E  0302  033E  0302  FFFF  FFFF  005A0                  1080, 1170, -1, -1, 770, 830, 770, 830, -                  ;
                                  0532  005A  FFFF  04E2  005B4                  -1, -1, 1250, -1, 1250, -1, 90, 1330                       ;
```

```
BASE=                   P.AAA
OPE_T=                  P.AAB
CLO_T=                  P.AAC
CLR_S_S_T=              P.AAD
REA_S_S_T=              P.AAE
REA_R_S_T=              P.AAE
REA_R_R_T=              P.AAF
REA_I_S_T=              P.AAE
REA_I_R_T=              P.AAG
WRI_S_S_T=              P.AAH
WRI_R_S_T=              P.AAI
WRI_R_R_T=              P.AAJ
WRI_I_S_T=              P.AAK
WRI_I_R_T=              P.AAL
DEL_R_S_T=              P.AAM
DEL_R_R_T=              P.AAN
DEL_I_S_T=              P.AAM
DEL_I_R_T=              P.AAN
REW_S_S_T=              P.AAO
REW_R_S_T=              P.AAP
REW_R_R_T=              P.AAQ
REW_I_S_T=              P.AAR
REW_I_R_T=              P.AAS
STA_R_S_T=              P.AAT
STA_I_S_T=              P.AAT
UNL_T=                  P.AAU
PLIT_TABLE=             P.AAV-2
        .EXTRN  LIB$STOP, COB$HANDLER
        .EXTRN  COB$$INVOKE_USE
        .EXTRN  COB$_NO_USEPRO, COB$_ERRON_FIL
        .EXTRN  COB$_OPTMISOPE, COB$_FILALRLOC
        .EXTRN  COB$_FILALROPE, COB$_FILCLOLOC
        .EXTRN  COB$_NO_SPACE, COB$_FILNOTFOU
```

```
                        .EXTRN  COB$_OPTMISCLO, COB$_FILALRCLO
                        .EXTRN  COB$_NO_NEXLOG, COB$_OPTMISREA
                        .EXTRN  COB$_NO_NEXVAL, COB$_RECLOCREA
                        .EXTRN  COB$_RECLOC_OK, COB$_REAUNOFIL
                        .EXTRN  COB$_REAINCOPE, COB$_WRIBEYBOU
                        .EXTRN  COB$_WRIUNOFIL, COB$_WRIINCOPE
                        .EXTRN  COB$_REWNO_R_S, COB$_REWUNOFIL
                        .EXTRN  COB$_REWINCOPE, COB$_RECNOTEXI
                        .EXTRN  COB$_OPTMISSTA, COB$_RECLOCSTA
                        .EXTRN  COB$_STAUNOFIL, COB$_STAINCOPE
                        .EXTRN  COB$_RECLOCWRI, COB$_RECLOCDEL
                        .EXTRN  COB$_DELNO_R_S, COB$_DELUNOFIL
                        .EXTRN  COB$_DELINCOPE, COB$_RECLOCREW
                        .EXTRN  COB$_WRIDUPKEY, COB$_WRICREDUP
                        .EXTRN  COB$_WRINOTASC, COB$_WRIDUPALT
                        .EXTRN  COB$_REWCREDUP, COB$_PRIKEYCHA
                        .EXTRN  COB$_REWDISDUP, COB$_WRIDISDUP
                        .EXTRN  COB$_REASMAMIN, COB$_WRISMAMIN
                        .EXTRN  COB$_REWSMAMIN, COB$_ORGNOTMAT
                        .EXTRN  COB$_INVARG, COB$_LSTHNDUSE
                        .EXTRN  COB$_KEYNOTMAT, COB$_RECNOTLOC
                        .EXTRN  COB$_UNLUNOFIL, COB$_UNLNO_CUR
                        .EXTRN  COB$_REAMP_D_R, OTS$_FATINTERR

                  0FFC 00000    .ENTRY  COB$IOEXCEPTION, Save R2,R3,R4,R5,R6,R7,R8,-; 1201
                                        R9,R10,R11
5E                08  C2 00002 SUBL2   #8, SP                                      ;
02                6C  91 00005 CMPB    (AP), #2                                   ; 1355
                  64  1F 00008 BLSSU   5$                                         ;
                  57  08 AC D0 0000A   MOVL    RAB, R7                            ; 1361
                      5E 13 0000E BEQL    5$                                      ;
                  56  3C A7 D0 00010   MOVL    60(R7), FAB                        ; 1367
                      58 13 00014 BEQL    5$                                      ; 1368
                  52  28 A6 D0 00016   MOVL    40(FAB), NAM                       ; 1374
                      52 13 0001A BEQL    5$                                      ; 1375
                      5A 7C 0001C CLRQ    OPEN_MD_ADDR                           ; 1388
              04  AE  04 A2 D0 0001E   MOVL    4(NAM), RSADESC+4                  ; 1398
                  6E  03 A2 9A 00023   MOVZBL  3(NAM), RSADESC                    ; 1399
                      14 12 00027 BNEQ    1$                                      ; 1400
                  6E  0B A2 9A 00029   MOVZBL  11(NAM), RSADESC                   ; 1402
                      0E 12 0002D BNEQ    1$                                      ; 1403
                  6E  34 A6 9A 0002F   MOVZBL  52(FAB), RSADESC                   ; 1406
                  50  34 A6 9A 00033   MOVZBL  52(FAB), R0                       ; 1415
        04    52      2C B6 50 28 00037   MOVC3   R0, @44(FAB), @4(NAM)           ;
          07      00 06 AC AF 0003D 1$: CASEW   FLAGS+2, #0, #7                   ; 1426
0187        017E      0013  0013  00042 2$: .WORD   3$-2$,-                       ;
0223        0211      0190  0199  0004A           3$-2$,-                       ;
                                              32$-2$,-                          ;
                                              33$-2$,-                          ;
                                              35$-2$,-                          ;
                                              34$-2$,-                          ;
                                              42$-2$,-                          ;
                                              44$-2$                            ;
                  028B 31 00052       BRW     54$                               ; 1431
                06  AC B5 00055 3$: TSTW    FLAGS+2                             ; 1442
                    0F 12 00058     BNEQ    4$                                  ;
                08  A7 D5 0005A     TSTL    8(R7)                               ; 1445
```

```
                                        0F   13 0005D        BEQL    5$                        ;
                        54      08      A7   D0 0005F        MOVL    8(R7), RMS_STS            ; 1451
                        59      0C      A7   D0 00063        MOVL    12(R7), RMS_STV           ; 1452
                                        16   11 00067        BRB     7$                        ; 1442
                                08      A6   D5 00069 4$:    TSTL    8(FAB)                    ; 1461
                                        09   12 0006C        BNEQ    6$                        ;
                           00000000G    8F   DD 0006E 5$:    PUSHL   #OTS$_FATINTERR           ; 1464
                                      026F   31 00074        BRW     55$                       ;
                        54      08      A6   D0 00077 6$:    MOVL    8(FAB), RMS_STS           ; 1467
                        59      0C      A6   D0 0007B        MOVL    12(FAB), RMS_STV          ; 1468
                        52      04      AC   3C 0007F 7$:    MOVZWL  FLAGS, R2                 ; 1493
                        03              54   E8 00083        BLBS    RMS_STS, 8$               ; 1481
                                      009A   31 00086        BRW     17$                       ;
            2B          01              52   CF 00089 8$:    CASEL   R2, #1, #43               ; 1493
  0253      0253        0253          0253      0008D 9$:    .WORD   54$-9$,-                  ;
  0253      0253        0253          0253      00095                54$-9$,-                  ;
  0253      0253        0253          0253      0009D                54$-9$,-                  ;
  005B      005B        0253          005B      000A5                54$-9$,-                  ;
  0253      0253        005B          005B      000AD                54$-9$,-                  ;
  0078      0078        0253          0253      000B5                54$-9$,-                  ;
  0253      0253        0253          0253      000BD                54$-9$,-                  ;
  0253      0253        0078          0078      000C5                54$-9$,-                  ;
  0253      0253        0253          0253      000CD                54$-9$,-                  ;
  0253      0253        0253          0253      000D5                54$-9$,-                  ;
  0253      0253        0253          0253      000DD                54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     11$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     11$-9$,-                  ;
                                                                     11$-9$,-                  ;
                                                                     11$-9$,-                  ;
                                                                     11$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     13$-9$,-                  ;
                                                                     13$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     13$-9$,-                  ;
                                                                     13$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
                                                                     54$-9$,-                  ;
```

```
                                                                                54$-9$
                                                01F8  31 000E5 10$:    BRW     54$                                                 ;1553
                                          FD    A7      02  8A 000E8 11$:    BICB2   #2, -3(R7)                                     ;1508
                                    00018021    8F      54  D1 000EC          CMPL    RMS_STS, #98337                             ;1516
                                                        09  13 000F3          BEQL    12$                                         ;
                                    00018029    8F      54  D1 000F5          CMPL    RMS_STS, #98345                             ;
                                                        1A  12 000FC          BNEQ    15$                                         ;
                                          53    3039    8F  3C 000FE 12$:     MOVZWL  #12345, FILESTAT                            ;1519
                                                        0E  11 00103          BRB     14$                                         ;1520
                                    00018011    8F      54  D1 00105 13$:     CMPL    RMS_STS, #98321                             ;1539
                                                        0A  12 0010C          BNEQ    15$                                         ;
                                          53    3230    8F  3C 0010E          MOVZWL  #12848, FILESTAT                            ;1542
                                          55            02  D0 00113 14$:     MOVL    #2, ACTION                                  ;1543
                                                        08  11 00116          BRB     16$                                         ;1532
                                          53    3030    8F  3C 00118 15$:     MOVZWL  #12336, FILESTAT                            ;1547
                                          55            03  D0 0011D          MOVL    #3, ACTION                                  ;1548
                                                    015B  31 00120 16$:     BRW     46$                                         ;1532
                                                        C0  15 00123 17$:     BLEQ    10$                                         ;1575
                                          2C            52  B1 00125          CMPW    R2, #44                                     ;1576
                                                        BB  1A 00128          BGTRU   10$
                                          52    FE77 CF42  3C 0012A          MOVZWL  PLIT_TABLE[R2], TABLE                       ;1580
                                                        B3  19 00130          BLSS    10$
                                          52    F90D CF42  9E 00132          MOVAB   BASE[TABLE], TABLE                          ;1594
                                                        62  D5 00138 18$:     TSTL    (TABLE)                                     ;1603
                                                        05  13 0013A          BEQL    19$
                                          54            62  D1 0013C          CMPL    (TABLE), RMS_STS
                                                        21  12 0013F          BNEQ    24$
                                    03          00      08  A2  8F 00141 19$:     CASEB   8(TABLE), #0, #3                            ;1614
                              0017          0011          000A    0021      00146 20$:     .WORD   25$-20$,-                                   ;
                                                                                    21$-20$,-                                   ;
                                                                                    22$-20$,-                                   ;
                                                                                    23$-20$
                                                        17  11 0014E          BRB     25$
                                    0D          FC    A7      01  E1 00150 21$:     BBC     #1, -4(R7), 24$                     ;1619
                                                        10  11 00155          BRB     25$
                                          07        FD  A7  E9 00157 22$:     BLBC    -3(R7), 24$                                 ;1621
                                                        0A  11 0015B          BRB     25$
                                    05          FD    A7      01  E0 0015D 23$:     BBS     #1, -3(R7), 25$                     ;1623
                                          52            0A  C0 00162 24$:     ADDL2   #10, TABLE                                  ;1634
                                                        D1  11 00165          BRB     18$                                         ;1602
                                          53    04      A2  3C 00167 25$:     MOVZWL  4(TABLE), FILESTAT                          ;1642
                                          05            6C  91 0016B          CMPB    (AP), #5                                    ;1654
                                                        24  1F 0016E          BLSSU   27$
                                          20    14      AC  E9 00170          BLBC    ANSI74, 27$                                 ;1655
                                    00003331    8F      53  D1 00174          CMPL    FILESTAT, #13105                            ;1656
                                                        12  13 0017B          BEQL    26$
                                    00003531    8F      53  D1 0017D          CMPL    FILESTAT, #13617                            ;1657
                                                        09  13 00184          BEQL    26$
                                    00003631    8F      53  D1 00186          CMPL    FILESTAT, #13873                            ;1658
                                                        05  12 0018D          BNEQ    27$
                                          53    3031    8F  3C 0018F 26$:     MOVZWL  #12337, FILESTAT                            ;1659
                                          55      06    A2  9A 00194 27$:     MOVZBL  6(TABLE), ACTION                            ;1661
                                          50      07    A2  9A 00198          MOVZBL  7(TABLE), R0                                ;1662
                                          56  00000000G9F  40  7E 0019C          MOVAQ   @#COB$_NO_USEPRO[R0], ERR_MSG_NUM
                                    03          00        09  A2  8F 001A4          CASEB   9(TABLE), #0, #3                            ;1671
                              0010          000A          00D5      00D5      001A9 28$:     .WORD   46$-28$,-                                   ;
                                                                                    46$-28$,-                                   ;
```

```
                                                                      29$-28$,-                        ;
                                                                      30$-28$                          ;
                                    0A  11 001B1            BRB       31$                              ;
                  FD    A7          01  88 001B3 29$:       BISB2     #1, -3(R7)                       ; 1678
                                    04  11 001B7            BRB       31$                              ;
                  FD    A7          02  88 001B9 30$:       BISB2     #2, -3(R7)                       ; 1681
                                  00BE  31 001BD 31$:       BRW       46$                              ; 1426
                        56 00000000G  8F  D0 001C0 32$:     MOVL      #COB$_FILCLOLOC, ERR_MSG_NUM     ; 1700
                                    7F  11 001C7            BRB       39$                              ; 1701
                        56 00000000G  8F  D0 001C9 33$:     MOVL      #COB$_FILALROPE, ERR_MSG_NUM     ; 1718
                                    7F  11 001D0            BRB       41$                              ; 1719
                        56 00000000G  8F  D0 001D2 34$:     MOVL      #COB$_ORGNOTMAT, ERR_MSG_NUM     ; 1735
                                    7F  11 001D9            BRB       43$                              ; 1736
              2B          01      04  AC  AF 001DB 35$:     CASEW     FLAGS, #1, #43                   ; 1760
0100          0100        0100       0100    001E0 36$:     .WORD     54$-36$,-                        ;
0100          0100        0100       0100    001E8                    54$-36$,-                        ;
0100          0100        0100       0100    001F0                    54$-36$,-                        ;
0058          0058        0100       0058    001F8                    54$-36$,-                        ;
0100          0061        0058       0058    00200                    54$-36$,-                        ;
0061          0061        0061       0061    00208                    54$-36$,-                        ;
006A          006A        0100       006A    00210                    54$-36$,-                        ;
0100          0100        006A       006A    00218                    54$-36$,-                        ;
0100          0100        0100       0100    00220                    54$-36$,-                        ;
0100          0100        0100       0100    00228                    54$-36$,-                        ;
0100          0100        0100       0100    00230                    54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      37$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      37$-36$,-                        ;
                                                                      37$-36$,-                        ;
                                                                      37$-36$,-                        ;
                                                                      37$-36$,-                        ;
                                                                      38$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      38$-36$,-                        ;
                                                                      38$-36$,-                        ;
                                                                      38$-36$,-                        ;
                                                                      38$-36$,-                        ;
                                                                      40$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      40$-36$,-                        ;
                                                                      40$-36$,-                        ;
                                                                      40$-36$,-                        ;
                                                                      40$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
                                                                      54$-36$,-                        ;
```

```
                                                        54$-36$
                56 00000000G   8F   D0 00238 37$:    MOVL    #COB$_REASMAMIN, ERR_MSG_NUM                1776
                               19   11 0023F         BRB     43$
                56 00000000G   8F   D0 00241 38$:    MOVL    #COB$_WRISMAMIN, ERR_MSG_NUM                1790
                               10   11 00248 39$:    BRB     43$
                56 00000000G   8F   D0 0024A 40$:    MOVL    #COB$_REWSMAMIN, ERR_MSG_NUM                1804
                               07   11 00251 41$:    BRB     43$
                56 00000000G   8F   D0 00253 42$:    MOVL    #COB$_KEYNOTMAT, ERR_MSG_NUM                1830
                               54   D4 0025A 43$:    CLRL    RMS_STS                                     1831
                               59   D4 0025C         CLRL    RMS_STV                                     1832
                53      3439   8F   3C 0025E         MOVZWL  #13369, FILESTAT                            1833
                               10   11 00263         BRB     45$                                         1834
                56 00000000G   8F   D0 00265 44$:    MOVL    #COB$_REAMP_D_R, ERR_MSG_NUM                1848
                               54   D4 0026C         CLRL    RMS_STS                                     1849
                               59   D4 0026E         CLRL    RMS_STV                                     1850
                53      3339   8F   3C 00270         MOVZWL  #13T13, FILESTAT                            1851
                               55   D4 00275 45$:    CLRL    ACTION                                      1852
          08    A7             56   D0 00277         MOVL    ERR_MSG_NUM, 8(R7)                          1853
                      0C       A7   D4 0027B         CLRL    12(R7)                                      1854
                04             6C   91 0027E 46$:    CMPB    (AP), #4                                    1894
                               09   1F 00281         BLSSU   47$
                      10       AC   D5 00283         TSTL    STATUS
                               04   13 00286         BEQL    47$
          10    BC             53   B0 00288         MOVW    FILESTAT, @STATUS                           1896
                01             55   D1 0028C 47$:    CMPL    ACTION, #1                                  1905
                               11   12 0028F         BNEQ    48$
                03             6C   91 00291         CMPB    (AP), #3
                               0C   1F 00294         BLSSU   48$
                      0C       AC   D5 00296         TSTL    EXCLAB
                               07   13 00299         BEQL    48$
             18   AE       0C   AC   D0 0029B         MOVL    EXCLAB, 24(FP)                              1908
                               4B   11 002A0         BRB     56$                                         1909
                03             55   D1 002A2 48$:    CMPL    ACTION, #3                                  1916
                               03   12 002A5         BNEQ    49$
                            0092   31 002A7         BRW     64$
                52       14    AE   D0 002AA 49$:    MOVL    20(FP), SFP                                 1924
                               03   12 002AE         BNEQ    50$                                         1925
                            FDBB   31 002B0         BRW     5$
                53       FC    A2   D0 002B3 50$:    MOVL    -4(SFP), USE                                1933
                               7E   13 002B7         BEQL    63$                                         1934
                52       28    A3   9E 002B9         MOVAB   40(R3), USEENT                              1944
                58       24    A3   9A 002BD         MOVZBL  36(USE), I                                  1945
                               11   11 002C1         BRB     53$
                57       08    A2   D1 002C3 51$:    CMPL    8(USEENT), R7                               1947
                               08   12 002C7         BNEQ    52$
                      04       A2   D5 002C9         TSTL    4(USEENT)                                   1964
                               2F   12 002CC         BNEQ    58$
                5B             62   D0 002CE         MOVL    (USEENT), FILE_ADDR                         1988
                52             0C   C0 002D1 52$:    ADDL2   #12, USEENT                                 1990
                EC             58   F4 002D4 53$:    SOBGEQ  I, 51$                                      1945
                58       FE    A7   9A 002D7         MOVZBL  -2(R7), R8                                  2003
                03             58   91 002DB         CMPB    R8, #3
                               0F   1B 002DE         BLEQU   57$
                   00000000G   8F   DD 002E0 54$:    PUSHL   #COB$_INVARG                                2006
00000000G       00             01   FB 002E6 55$:    CALLS   #1, LIB$STOP
                               63   11 002ED 56$:    BRB     66$                                         2007
                52       04 A348   7E 002EF 57$:    MOVAQ   4(USE)[R8], USEENT                          2009
```

```
                                62  D5 002F4        TSTL    (USEENT)                          ; 2010
                                1D  13 002F6        BEQL    60$                               ;
                          04    A2  D5 002F8        TSTL    4(USEENT)                         ; 2028
                                15  13 002FB        BEQL    59$                               ;
                                63  DD 002FD 58$:   PUSHL   (USE)                             ; 2036
                          04    A2  DD 002FF        PUSHL   4(USEENT)                         ; 2035
                          18    AE  DD 00302        PUSHL   24(FP)                            ; 2034
                                53  DD 00305        PUSHL   USE                               ; 2033
                                62  DD 00307        PUSHL   (USEENT)                          ; 2032
          00000000G  00         05  FB 00309        CALLS   #5, COB$$INVOKE_USE               ;
                                1E  11 00310        BRB     62$                               ; 2037
                     5A         62  D0 00312 59$:   MOVL    (USEENT), OPEN_MD_ADDR            ; 2050
                                5B  D5 00315 60$:   TSTL    FILE_ADDR                         ; 2061
                                04  12 00317        BNEQ    61$                               ;
                                5A  D5 00319        TSTL    OPEN_MD_ADDR                      ;
                                1A  13 0031B        BEQL    63$                               ;
                                5A  DD 0031D 61$:   PUSHL   OPEN_MD_ADDR                      ; 2064
                                5B  DD 0031F        PUSHL   FILE_ADDR                         ;
                                02  DD 00321        PUSHL   #2                                ;
                     00000000G  8F  DD 00323        PUSHL   #COB$_LSTHNDUSE                   ;
          00000000G  00         04  FB 00329        CALLS   #4, LIB$SIGNAL                    ;
                     02         55  D1 00330 62$:   CMPL    ACTION, #2                        ; 2065
                                1D  12 00333        BNEQ    66$                               ;
                                05  11 00335        BRB     64$                               ; 2069
                     02         55  D1 00337 63$:   CMPL    ACTION, #2                        ; 2079
                                04  12 0033A        BNEQ    65$                               ;
                     50         01  D0 0033C 64$:   MOVL    #1, R0                            ; 2081
                                    04 0033F        RET                                       ;
                        0210    8F  BB 00340 65$:   PUSHR   #^M<R4,R9>                        ; 2090
                          08    AE  9F 00344        PUSHAB  RSADESC                           ; 2088
                                01  DD 00347        PUSHL   #1                                ;
                                56  DD 00349        PUSHL   ERR_MSG_NUM                       ; 2089
          00000000G  00         05  FB 0034B        CALLS   #5, LIB$STOP                      ;
                                50  D4 00352 66$:   CLRL    R0                                ; 2094
                                    04 00354        RET                                       ;

; Routine Size:  853 bytes,    Routine Base:  _COB$CODE + 05BC
```

```
; 1569          2095  0 END ELUDOM


                                                        .EXTRN  LIB$SIGNAL, LIB$STOP

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;  _COB$CODE                      2321  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        38         0        581          00:00.7


;                                       COMMAND QUALIFIERS

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

; Size:          853 code + 1468 data bytes
; Run Time:        00:27.3
; Elapsed Time:    01:32.9
; Lines/CPU Min:    4606
; Lexemes/CPU-Min: 37903
; Memory Used:  406 pages
; Compilation Complete
```

COBPAUSE
LIS

COBMSG
LIS

COBHANDLE
LIS

COBINTAR
LIS

COBINTER
LIS

COBMULQ
LIS

COBPOSERA
LIS

COBIOEXCE
LIS

COBLINAGE
LIS

COBKEY
LIS

COBINVUSE
LIS