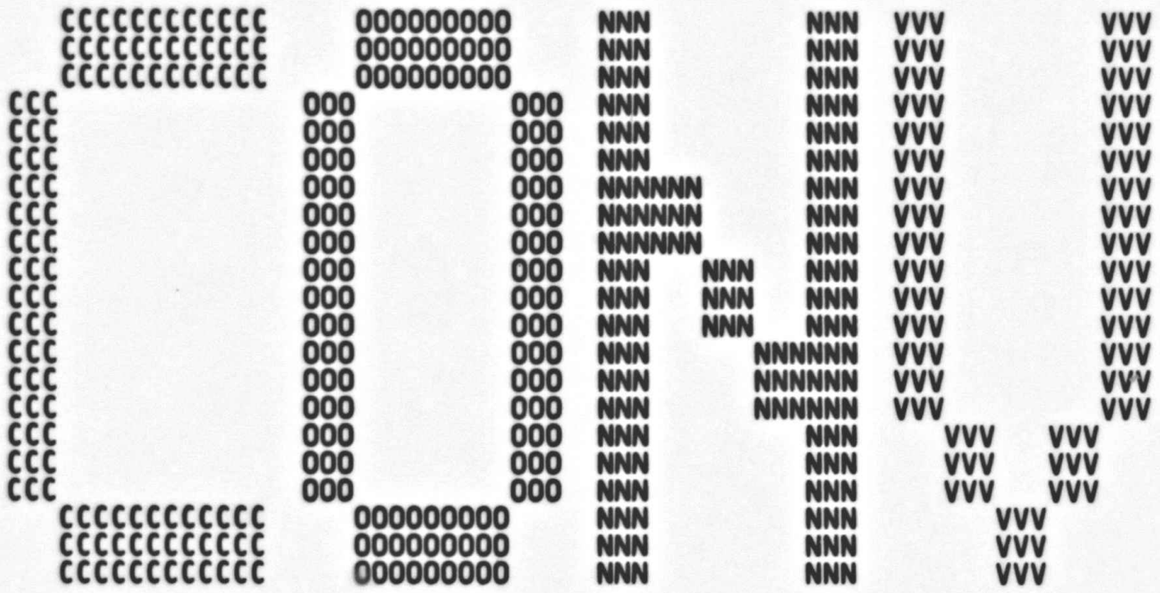
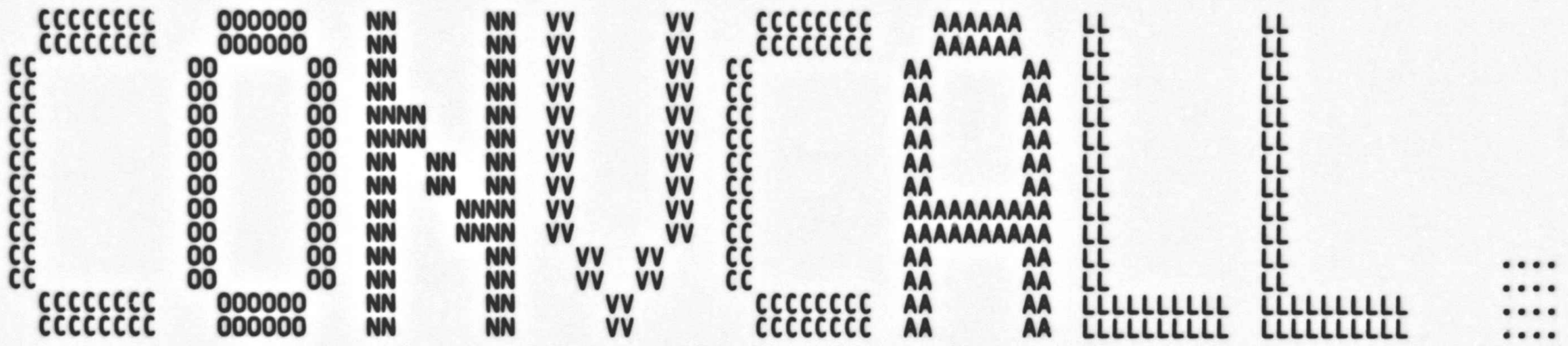

CCCCCCCCCCCC OOOOOOOOO NNN NNN VVV VVV
CCCCCCCCCCCC OOOOOOOOO NNN NNN VVV VVV
CCCCCCCCCCCC OOOOOOOOO NNN NNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCC OOO OOO NNNNNN NNN VVV VVV
CCC OOO OOO NNNNNN NNN VVV VVV
CCC OOO OOO NNNNNN NNN VVV VVV
CCC OOO OOO NNN NNN NNN VVV VVV
CCC OOO OOO NNN NNN NNN VVV VVV
CCC OOO OOO NNN NNN NNN VVV VVV
CCC OOO OOO NNN NNNNNN VVV VVV
CCC OOO OOO NNN NNNNNN VVV VVV
CCC OOO OOO NNN NNNNNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCC OOO OOO NNN NNN VVV VVV
CCCCCCCCCCCC OOOOOOOOO NNN NNN VVV
CCCCCCCCCCCC OOOOOOOOO NNN NNN VVV
CCCCCCCCCCCC OOOOOOOOO NNN NNN VVV

FILEID**CONVCALL



```
;    1          0001  0 %TITLE  'VAX-11 CONVERT'
;    2          0002  0 MODULE  CONV$CALL       ( IDENT='V04-000',
;    3          0003  0                         OPTLEVEL=3
;    4          0004  0                         ) =
;    5          0005  0
;    6          0006  1 BEGIN
;    7          0007  1
;    8          0008  1 !*****************************************************************************
;    9          0009  1 !*                                                                           *
;   10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;   11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
;   12          0012  1 !*  ALL RIGHTS RESERVED.                                                     *
;   13          0013  1 !*                                                                           *
;   14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;   15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;   16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;   17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;   18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;   19          0019  1 !*  TRANSFERRED.                                                             *
;   20          0020  1 !*                                                                           *
;   21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;   22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;   23          0023  1 !*  CORPORATION.                                                             *
;   24          0024  1 !*                                                                           *
;   25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;   26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;   27          0027  1 !*                                                                           *
;   28          0028  1 !*                                                                           *
;   29          0029  1 !*****************************************************************************
```

```
;    31          0030  1 !++
;    32          0031  1 !
;    33          0032  1 ! Facility:     VAX-11 CONVERT
;    34          0033  1 !
;    35          0034  1 ! Abstract:     CONVERT sharable image callable routines
;    36          0035  1 !
;    37          0036  1 ! Contents:
;    38          0037  1 !               PASS_FILES
;    39          0038  1 !               PASS_OPTIONS
;    40          0039  1 !               CONVERT
;    41          0040  1 !               ADD_KEY
;    42          0041  1 !               RECLAIM
;    43          0042  1 !               RUNDOWN
;    44          0043  1 !               CONDITION_HANDLER
;    45          0044  1 !               COPY_DESC
;    46          0045  1 !
;    47          0046  1 ! Environment:
;    48          0047  1 !
;    49          0048  1 !               VAX/VMS Operating System
;    50          0049  1 !
;    51          0050  1 !--
;    52          0051  1
;    53          0052  1 !
;    54          0053  1 ! Author:       Keith B Thompson        Creation date:  August-1981
;    55          0054  1 !
;    56          0055  1 !
;    57          0056  1 ! Modified by:
;    58          0057  1 !
;    59          0058  1 !       V03-013 JWT0185         Jim Teague              2-Jul-1984
;    60          0059  1 !               Add late PUT_RECORD call to empty the STM buffer
;    61          0060  1 !               for FTN --> STM conversions.
;    62          0061  1 !
;    63          0062  1 !       V03-012 RAS0264         Ron Schaefer            8-Mar-1984
;    64          0063  1 !               Improve the condition handler so that it will not
;    65          0064  1 !               rundown everything unless there will be no return.
;    66          0065  1 !
;    67          0066  1 !       V03-011 RAS0260         Ron Schaefer            2-Mar-1984
;    68          0067  1 !               Improve performance of RAS0250 by enabling open by NAM.
;    69          0068  1 !
;    70          0069  1 !       V03-010 RAS0250         Ron Schaefer            23-Feb-1984
;    71          0070  1 !               Add call to LIB$FIND_FILE_END during cleanup.
;    72          0071  1 !               Check for null parameters before using them.
;    73          0072  1 !
;    74          0073  1 !       V03-009 RAS0211         Ron Schaefer            8-Nov-1983
;    75          0074  1 !               Make sure input file is closed in RUNDOWN.
;    76          0075  1 !
;    77          0076  1 !       V03-008 KBT0540         Keith B. Thompson       9-Jun-1983
;    78          0077  1 !               Fix some bugs
;    79          0078  1 !
;    80          0079  1 !       V03-007 KBT0441         Keith B. Thompson       30-Dec-1982
;    81          0080  1 !               Add conv$add_key routine
;    82          0081  1 !
;    83          0082  1 !       V03-006 KBT0436         Keith B. Thompson       16-Dec-1982
;    84          0083  1 !               Use str$analyze_sdesc to check descriptors
;    85          0084  1 !
;    86          0085  1 !       V03-005 KBT0397         Keith B. Thompson       3-Nov-1982
;    87          0086  1 !               Fix the check for statistics block in reclaim
```

```
;    88          0087  1 !
;    89          0088  1 !       V03-004 KBT0390         Keith B. Thompson       28-Oct-1982
;    90          0089  1 !               Add support for prolgue 3 sidrs in reclaim
;    91          0090  1 !
;    92          0091  1 !       V03-003 KBT0371         Keith B. Thompson       19-Oct-1982
;    93          0092  1 !               Copy the user strings to local storage and add flags
;    94          0093  1 !               parameter to user call interface
;    95          0094  1 !
;    96          0095  1 !       V03-002 KBT0345         Keith B. Thompson       4-Oct-1982
;    97          0096  1 !               Use new definitions of the linkages
;    98          0097  1 !
;    99          0098  1 !       V03-001 KBT0021         Keith Thompson          23-Mar-1982
;   100          0099  1 !               Set defered write on output file
;   101          0100  1 !
;   102          0101  1 !****
```

```
;  104          0102  1
;  105          0103  1 PSECT
;  106          0104  1         OWN     = _CONV$OWN     (PIC),
;  107          0105  1         GLOBAL  = _CONV$GLOBAL  (PIC),
;  108          0106  1         PLIT    = _CONV$PLIT    (SHARE,PIC),
;  109          0107  1         CODE    = _CONV$CODE    (SHARE,PIC);
;  110          0108  1
;  111          0109  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  112          0110  1 LIBRARY 'SRC$:CONVERT';
;  113          0111  1
;  114          0112  1 DEFINE_ERROR_CODES;
;  115          0113  1
;  116          0114  1 LINKAGE
;  117          0115  1         L$ANALYZE_SDESC_R1      = JSB ( REGISTER = 0 ) : GLOBAL ( ADDRESS = 1 ),
;  118          0116  1         CL$COPY_DESC            = JSB ( REGISTER = 0 );
;  119          0117  1
;  120          0118  1 EXTERNAL ROUTINE
;  121          0119  1         LIB$ESTABLISH           : ADDRESSING_MODE ( GENERAL ),
;  122          0120  1         LIB$FIND_FILE_END       : ADDRESSING_MODE ( GENERAL ),
;  123          0121  1         LIB$SIG_TO_RET          : ADDRESSING_MODE ( GENERAL ),
;  124          0122  1         STR$ANALYZE_SDESC_R1    : L$ANALYZE_SDESC_R1 ADDRESSING_MODE ( GENERAL ),
;  125          0123  1         CONV$$CREATE_BUFFER,
;  126          0124  1         CONV$$CONVERT,
;  127          0125  1         CONV$$END_OF_FILE,
;  128          0126  1         CONV$$FREE_VM           : CL$FREE_VM    NOVALUE,
;  129          0127  1         CONV$$FREE_TEMP_VM      : CL$FREE_TEMP_VM       NOVALUE,
;  130          0128  1         CONV$$GET_NEXT_REY      : CL$JSB_REG_9,
;  131          0129  1         CONV$$GET_VM            : CL$GET_VM,
;  132          0130  1         CONV$$OPEN_INPUT,
;  133          0131  1         CONV$$OPEN_OUTPUT,
;  134          0132  1         CONV$$PARSE_DEF,
;  135          0133  1         CONV$$RMS_OPEN_ERROR    : NOVALUE,
;  136          0134  1         CONV$$SET_KEY_DESC      : CL$JSB_REG_9,
;  137          0135  1         ADD$$CHECK_KEY          : AL$CHECK_KEY,
;  138          0136  1         ADD$$LOAD_REY           : AL$LOAD_REY,
;  139          0137  1         ADD$$OPEN_OUTPUT,
;  140          0138  1         RECL$$OPEN_FILE,
;  141          0139  1         RECL$$ALLOCATE_BUFFERS  : RL$JSB_REG_9,
;  142          0140  1         RECL$$SCAN_DATA_LEVEL   : RL$JSB_REG_9,
;  143          0141  1         CONV$$PUT_RECORD        : ADDRESSING_MODE (GENERAL);
;  144          0142  1
;  145          0143  1 FORWARD ROUTINE
;  146          0144  1         RUNDOWN                 : NOVALUE,
;  147          0145  1         CONDITION_HANDLER,
;  148          0146  1         COPY_DESC               : CL$COPY_DESC;
;  149          0147  1
;  150          0148  1 EXTERNAL
;  151          0149  1         CONV$AB_FLAGS           : BLOCK [ ,BYTE ],
;  152          0150  1         CONV$GB_CURRENT_FILE    : BYTE,
;  153          0151  1         CONV$GL_FINDFILE_CTX    : LONG,
;  154          0152  1
;  155          0153  1         CONV$AL_IN_FILE_NAM     : VECTOR [ ,LONG ],
;  156          0154  1         CONV$AR_OUT_FILE_NAM    : REF DESC_BLK,
;  157          0155  1         CONV$AR_FDL_FILE_NAM    : REF DESC_BLK,
;  158          0156  1         CONV$AB_EXC_RAB         : $RAB_DECL,
;  159          0157  1         CONV$AB_IN_XABSUM       : $XABSUM_DECL,
;  160          0158  1         CONV$AB_IN_XABFHC       : $XABFHC_DECL,
```

```
;  161     0159  1          CONV$AB_IN_NAM          : $NAM_DECL,
;  162     0160  1          CONV$AB_IN_FAB          : $FAB_DECL,
;  163     0161  1          CONV$AB_IN_RAB          : $RAB_DECL,
;  164     0162  1          CONV$AB_OUT_XABSUM      : $XABSUM_DECL,
;  165     0163  1          CONV$AB_OUT_NAM         : $NAM_DECL,
;  166     0164  1          CONV$AB_OUT_FAB         : $FAB_DECL,
;  167     0165  1          CONV$AB_OUT_RAB         : $RAB_DECL,
;  168     0166  1          CONV$AB_RFA_FAB         : $FAB_DECL,
;  169     0167  1          CONV$AB_RFA_RAB         : $RAB_DECL,
;  170     0168  1
;  171     0169  1          CONV$GL_ADD_DELE_KEY    : LONG,
;  172     0170  1          CONV$GL_STM_BUF,
;  173     0171  1          CONV$GL_STM_REC_LEN,
;  174     0172  1          CONV$GW_OUT_REC_SIZ     : SIGNED WORD;
;  175     0173  1
;  176     0174  1  LITERAL
;  177     0175  1          COUNTERS = 4,           ! Number of counters
;  178     0176  1          OPTIONS  = 19;          ! Number of options
;  179     0177  1
;  180     0178  1  OWN
;  181     0179  1          ! Exception fab for opens and closes
;  182     0180  1          !
;  183     0181  1          EXC_FAB         : $FAB_DECL,
;  184     0182  1
;  185     0183  1          ! Call sequence counter
;  186     0184  1          !
;  187     0185  1          SEQUENCE        : BYTE INITIAL( 0 ),
;  188     0186  1
;  189     0187  1          ! The Option Flags:
;  190     0188  1          !
;  191     0189  1          OPTION_BLOCK    : VECTOR [ OPTIONS + 1,LONG ],  ! All of the options
;  192     0190  1                                                          ! plus one for the size
;  193     0191  1
;  194     0192  1          ! The counters
;  195     0193  1          !
;  196     0194  1          COUNT_BLOCK     : VECTOR [ COUNTERS + 1,LONG ]; ! All of the counters
;  197     0195  1
;  198     0196  1  BIND
;  199     0197  1          OPTION_COUNT    = OPTION_BLOCK [ 0 ] : LONG,    ! Option count
;  200     0198  1          COUNT_COUNT     = COUNT_BLOCK  [ 0 ] : LONG;    ! Counters count
;  201     0199  1
;  202     0200  1  GLOBAL BIND
;  203     0201  1
;  204     0202  1          ! The order of these options define the option block for the call
;  205     0203  1          ! interface. DO NOT change them.
;  206     0204  1          !
;  207     0205  1          CONV$GL_CREATE  = OPTION_BLOCK [  1 ] : LONG,   ! CREATE
;  208     0206  1          CONV$GL_SHARE   = OPTION_BLOCK [  2 ] : LONG,   ! SHARE
;  209     0207  1          CONV$GL_FAST    = OPTION_BLOCK [  3 ] : LONG,   ! FAST
;  210     0208  1          CONV$GL_MERGE   = OPTION_BLOCK [  4 ] : LONG,   ! MERGE
;  211     0209  1          CONV$GL_APPEND  = OPTION_BLOCK [  5 ] : LONG,   ! APPEND
;  212     0210  1          CONV$GL_SORT    = OPTION_BLOCK [  6 ] : LONG,   ! SORT
;  213     0211  1          CONV$GL_WORK_F  = OPTION_BLOCK [  7 ] : LONG,   ! WORK_FILES
;  214     0212  1          CONV$GL_KEY     = OPTION_BLOCK [  8 ] : LONG,   ! KEY
;  215     0213  1          CONV$GL_PAD     = OPTION_BLOCK [  9 ] : LONG,   ! PAD_RECORDS
;  216     0214  1          CONV$GL_PAD_CHAR= OPTION_BLOCK [ 10 ] : LONG,   ! Pad character
;  217     0215  1          CONV$GL_TRUNCATE= OPTION_BLOCK [ 11 ] : LONG,   ! TRUNCATE
```

```
;  218     0216  1          CONV$GL_EXIT     = OPTION_BLOCK [ 12 ] : LONG,    ! EXIT
;  219     0217  1          CONV$GL_FIX      = OPTION_BLOCK [ 13 ] : LONG,    ! FIXED WRITE
;  220     0218  1          CONV$GL_FILL     = OPTION_BLOCK [ 14 ] : LONG,    ! FILL_BUCKETS
;  221     0219  1          CONV$GL_READ_C   = OPTION_BLOCK [ 15 ] : LONG,    ! READ_CHECK
;  222     0220  1          CONV$GL_WRITE_C  = OPTION_BLOCK [ 16 ] : LONG,    ! WRITE_CHECK
;  223     0221  1          CONV$GL_FDL      = OPTION_BLOCK [ 17 ] : LONG,    ! FDL
;  224     0222  1          CONV$GL_EXC      = OPTION_BLOCK [ 18 ] : LONG,    ! EXCEPTION
;  225     0223  1          CONV$GL_PROLOG   = OPTION_BLOCK [ 19 ] : LONG,    ! PROLOGUE
;  226     0224  1
;  227     0225  1          ! These are the counters
;  228     0226  1          !
;  229     0227  1          CONV$GL_FILE_COUNT      = COUNT_BLOCK [ 1 ] : LONG,     ! Number of files processed
;  230     0228  1          CONV$GL_RECORD_COUNT    = COUNT_BLOCK [ 2 ] : LONG,     ! Number of Rec. Processed
;  231     0229  1          CONV$GL_EXCEPT_COUNT    = COUNT_BLOCK [ 3 ] : LONG,     ! Number of Exception Records
;  232     0230  1          CONV$GL_VALID_COUNT     = COUNT_BLOCK [ 4 ] : LONG;     ! Number of Valid Records
;  233     0231  1
;  234     0232  1 !        Data etc. for reclaim
;  235     0233  1 !
;  236     0234  1 OWN
;  237     0235  1          STATISTICS_BLOCK        : VECTOR [ 5,LONG ];
;  238     0236  1
;  239     0237  1 GLOBAL BIND
;  240     0238  1          RECL$GL_BUCKET_COUNT    = STATISTICS_BLOCK [ 1 ] : LONG,
;  241     0239  1          RECL$GL_DATA_COUNT      = STATISTICS_BLOCK [ 2 ] : LONG,
;  242     0240  1          RECL$GL_INDEX_COUNT     = STATISTICS_BLOCK [ 3 ] : LONG;
;  243     0241  1
```

```
;   245         0242  1 %SBTTL  'PASS_FILES'
;   246         0243  1 GLOBAL ROUTINE  CONV$PASS_FILES =
;   247         0244  1 !++
;   248         0245  1 !
;   249         0246  1 ! Functional Description:
;   250         0247  1 !
;   251         0248  1 !       File name passing routine
;   252         0249  1 !
;   253         0250  1 ! Calling Sequence:
;   254         0251  1 !
;   255         0252  1 !   Initial Call:
;   256         0253  1 !
;   257         0254  1 !       CONV$PASS_FILES( in_file_desc,
;   258         0255  1 !                        out_file_desc
;   259         0256  1 !                        [,fdl_file_desc]
;   260         0257  1 !                        [,exc_file_desc]
;   261         0258  1 !                        [,flags] )
;   262         0259  1 !
;   263         0260  1 !   Additional Calls:
;   264         0261  1 !
;   265         0262  1 !       CONV$PASS_FILES( in_file_desc[,flags] )
;   266         0263  1 !
;   267         0264  1 ! Input Parameters:
;   268         0265  1 !
;   269         0266  1 !       in_file_desc    - Address of a file descriptor used as the input file name
;   270         0267  1 !       out_file_desc   - Address of a file descriptor used as the output file name
;   271         0268  1 !       fdl_file_desc   - ( Optional ) Address of a string descriptor to be used
;   272         0269  1 !                                      as the file name of the fdl file
;   273         0270  1 !       exc_file_desc   - ( Optional ) Address of a string descriptor to be used
;   274         0271  1 !                                      as the file name of the exceptions file
;   275         0272  1 !       flags           - ( Optional ) Flags longword
;   276         0273  1 !
;   277         0274  1 ! Implicit Inputs:
;   278         0275  1 !       none
;   279         0276  1 !
;   280         0277  1 ! Output Parameters:
;   281         0278  1 !       none
;   282         0279  1 !
;   283         0280  1 ! Implicit Outputs:
;   284         0281  1 !       none
;   285         0282  1 !
;   286         0283  1 ! Routine Value:
;   287         0284  1 !
;   288         0285  1 !       SS$_NORMAL or
;   289         0286  1 !       CONV$_ORDER
;   290         0287  1 !       CONV$_NARG
;   291         0288  1 !       CONV$_INP_FILES
;   292         0289  1 !
;   293         0290  1 ! Routines Called:
;   294         0291  1 !       none
;   295         0292  1 !
;   296         0293  1 ! Side Effects:
;   297         0294  1 !       none
;   298         0295  1 !
;   299         0296  1 !--
;   300         0297  1
;   301         0298  2     BEGIN
```

```
;  302     0299  2
;  303     0300  2      BUILTIN
;  304     0301  2          ACTUALCOUNT,
;  305     0302  2          ACTUALPARAMETER,
;  306     0303  2          NULLPARAMETER;
;  307     0304  2
;  308     0305  2      ! Set up the condition handler to make sure files are closed and memory
;  309     0306  2      ! released
;  310     0307  2      !
;  311     0308  2      LIB$ESTABLISH( CONDITION_HANDLER );
;  312     0309  2
;  313     0310  2      ! See what kind of call it is
;  314     0311  2      !
;  315     0312  2      IF .SEQUENCE EQLU 0
;  316     0313  2      THEN
;  317     0314  3          BEGIN
;  318     0315  3
;  319     0316  3          LOCAL
;  320     0317  3              BYTES,
;  321     0318  3              VM_POINTER;
;  322     0319  3
;  323     0320  3          ! The first call needs at least two arguments and no more then 5
;  324     0321  3          !
;  325     0322  4          IF ( ACTUALCOUNT() LSSU 2 ) OR ( ACTUALCOUNT() GTRU 5 )
;  326     0323  3          THEN
;  327     0324  3              RETURN CONV$_NARG;
;  328     0325  3
;  329     0326  3          ! Clear the flags
;  330     0327  3          !
;  331     0328  3          CONV$AB_FLAGS = _CLEAR;
;  332     0329  3
;  333     0330  3          ! If the user specified a flags parameter stuff it
;  334     0331  3          !
;  335     0332  3          IF ACTUALCOUNT() EQLU 5
;  336     0333  3          THEN
;  337     0334  3              CONV$AB_FLAGS [ CONV$W_USER ] = .ACTUALPARAMETER(5);
;  338     0335  3
;  339     0336  3          ! Allocate memory for all of the name block buffers
;  340     0337  3          !
;  341     0338  3          BYTES = 2 * ( ESA_BUF_SIZ + RSA_BUF_SIZ );
;  342     0339  3
;  343     0340  3          VM_POINTER = CONV$$GET_VM ( .BYTES );
;  344     0341  3
;  345     0342  3          ! Init the input RMS blocks
;  346     0343  3          !
;  347     0344  3          ! The FAB
;  348     0345  3          !
;  349   P 0346  3          $FAB_INIT ( FAB = CONV$AB_IN_FAB,
;  350   P 0347  3                      FAC = <BRO,GET>,
;  351   P 0348  3                      FOP = <NAM,SQO>,
;  352   P 0349  3                      NAM = CONV$AB_IN_NAM,
;  353     0350  3                      XAB = CONV$AB_IN_XABSUM );
;  354     0351  3
;  355     0352  3          ! The RAB
;  356     0353  3          !
;  357   P 0354  3          $RAB_INIT ( RAB = CONV$AB_IN_RAB,
;  358   P 0355  3                      FAB = CONV$AB_IN_FAB,
```

```
;  359          0356  3                   ROP = RAH );
;  360          0357
;  361          0358                ! The name block
;  362          0359                !
;  363        P 0360                $NAM_INIT ( NAM = CONV$AB_IN_NAM,
;  364        P 0361                            ESA = .VM_POINTER,
;  365        P 0362                            ESS = ESA_BUF_SIZ,
;  366        P 0363                            RSA = .VM_POINTER + ESA_BUF_SIZ,
;  367          0364                            RSS = RSA_BUF_SIZ );
;  368          0365
;  369          0366                ! The xabs
;  370          0367                !
;  371        P 0368                $XABSUM_INIT ( XAB = CONV$AB_IN_XABSUM,
;  372        P 0369                               NXT = CONV$AB_IN_XABFHC );
;  373        P 0370                $XABFHC_INIT ( XAB = CONV$AB_IN_XABFHC,
;  374        P 0371                               NXT = 0 );
;  375          0372
;  376          0373                ! Get the next block of memory
;  377          0374                !
;  378          0375                VM_POINTER = .VM_POINTER + ( ESA_BUF_SIZ + RSA_BUF_SIZ );
;  379          0376
;  380          0377                ! Init the output RMS blocks
;  381          0378                !
;  382          0379                ! The FAB
;  383          0380                !
;  384        P 0381                $FAB_INIT ( FAB = CONV$AB_OUT_FAB,
;  385        P 0382                            FAC = <BRO,GET,PUT>,
;  386        P 0383                            FOP = <DFW,NAM,OFP,SQO>,
;  387        P 0384                            NAM = CONV$AB_OUT_NAM,
;  388          0385                            XAB = CONV$AB_OUT_XABSUM );
;  389          0386
;  390          0387                ! The RAB
;  391          0388                !
;  392        P 0389                $RAB_INIT ( RAB = CONV$AB_OUT_RAB,
;  393        P 0390                            FAB = CONV$AB_OUT_FAB,
;  394          0391                            ROP = WBH );
;  395          0392
;  396          0393                ! The name block
;  397          0394                !
;  398        P 0395                $NAM_INIT ( NAM = CONV$AB_OUT_NAM,
;  399        P 0396                            ESA = .VM_POINTER,
;  400        P 0397                            ESS = ESA_BUF_SIZ,
;  401        P 0398                            RLF = CONV$AB_IN_NAM,
;  402        P 0399                            RSA = .VM_POINTER + ESA_BUF_SIZ,
;  403          0400                            RSS = RSA_BUF_SIZ );
;  404          0401
;  405          0402                ! The xab
;  406          0403                !
;  407        P 0404                $XABSUM_INIT ( XAB = CONV$AB_OUT_XABSUM,
;  408          0405                               NXT = 0 );
;  409          0406
;  410          0407                ! Clear the count of input files
;  411          0408                !
;  412          0409                CONV$GL_FILE_COUNT = 0;
;  413          0410
;  414          0411                ! The second argument is the output file name
;  415          0412  3             !
```

```
;  416     0413  3          IF NULLPARAMETER(2)
;  417     0414  3          THEN
;  418     0415  3              RETURN CONV$_ILL_VALUE
;  419     0416  3          ELSE
;  420     0417  3              CONV$AR_OUT_FILE_NAM = COPY_DESC( ACTUALPARAMETER( 2 ) );
;  421     0418  3
;  422     0419  3          ! If there is a 3rd argument then it's the fdl file descriptor
;  423     0420  3          !
;  424     0421  3          IF NOT NULLPARAMETER(3)
;  425     0422  3          THEN
;  426     0423  3
;  427     0424  3              ! Copy the descriptor
;  428     0425  3              !
;  429     0426  3              CONV$AR_FDL_FILE_NAM = COPY_DESC( ACTUALPARAMETER( 3 ) );
;  430     0427  3
;  431     0428  3
;  432     0429  3          ! If there is a 4th argument then it's the exception file descriptor
;  433     0430  3          !
;  434     0431  3          IF NOT NULLPARAMETER(4)
;  435     0432  3          THEN
;  436     0433  4              BEGIN
;  437     0434  4
;  438     0435  4              LOCAL
;  439     0436  4                  EXC_NAM_PTR,
;  440     0437  4                  EXC_FILE_NAM    : REF DESC_BLK;
;  441     0438  4
;  442     0439  4              EXC_FILE_NAM = COPY_DESC( ACTUALPARAMETER( 4 ) );
;  443     0440  4
;  444     0441  4              ! Allocate memory for name block and buffers
;  445     0442  4              !
;  446     0443  4              EXC_NAM_PTR = CONV$$GET_VM( NAM$K_BLN + ESA_BUF_SIZ + RSA_BUF_SIZ );
;  447     0444  4
;  448     0445  4              ! Init the RMS blocks
;  449     0446  4              !
;  450     0447  4              ! The FAB
;  451     0448  4              !
;  452   P 0449  4              $FAB_INIT ( FAB = EXC_FAB,
;  453   P 0450  4                          DNM = '.EXC',
;  454   P 0451  4                          FNS = .EXC_FILE_NAM [ DSC$W_LENGTH ],
;  455   P 0452  4                          FNA = .EXC_FILE_NAM [ DSC$A_POINTER ],
;  456   P 0453  4                          FOP = <NAM,SQO,TEF>,
;  457   P 0454  4                          NAM = .EXC_NAM_PTR,
;  458   P 0455  4                          RFM = VAR,
;  459     0456  4                          RAT = CR );
;  460     0457  4
;  461     0458  4              ! The RAB
;  462     0459  4              !
;  463   P 0460  4              $RAB_INIT ( RAB = CONV$AB_EXC_RAB,
;  464   P 0461  4                          FAB = EXC_FAB,
;  465     0462  4                          ROP = WBH );
;  466     0463  4
;  467     0464  4              ! The name block
;  468     0465  4              !
;  469   P 0466  4              $NAM_INIT ( NAM = .EXC_NAM_PTR,
;  470   P 0467  4                          ESA = .EXC_NAM_PTR + NAM$K_BLN,
;  471   P 0468  4                          ESS = ESA_BUF_SIZ,
;  472   P 0469  4                          RSA = .EXC_NAM_PTR + NAM$K_BLN + ESA_BUF_SIZ,
```

```
;  473     0470  5                          RSS = RSA_BUF_SIZ )
;  474     0471  5
;  475     0472  5                  END;
;  476     0473  5
;  477     0474  5              ! We are a success so set up for the next call
;  478     0475  5              !
;  479     0476  5              SEQUENCE = 1
;  480     0477  5
;  481     0478  5              END
;  482     0479  5
;  483     0480  5          ! If not the first call it better be the second call to pass files
;  484     0481  5          !
;  485     0482  2          ELSE IF .SEQUENCE EQLU 1
;  486     0483  2          THEN
;  487     0484  2              BEGIN
;  488     0485  2
;  489     0486  2              ! More calls to pass_files means only one or two arguments
;  490     0487  2              !
;  491     0488  2              IF ACTUALCOUNT() NEQU 1
;  492     0489  2              THEN
;  493     0490  2
;  494     0491  2                  IF ACTUALCOUNT() EQLU 2
;  495     0492  2                  THEN
;  496     0493  2                      CONV$AB_FLAGS [ CONV$W_USER ] = .ACTUALPARAMETER(2)
;  497     0494  2                  ELSE
;  498     0495  2                      RETURN CONV$_NARG
;  499     0496  2
;  500     0497  2              END
;  501     0498  2
;  502     0499  3          ! If we are here we were called in the wrong order
;  503     0500  3          !
;  504     0501  2          ELSE
;  505     0502  2              RETURN CONV$_ORDER;
;  506     0503  2
;  507     0504  2          ! If there are to many input files exit
;  508     0505  2          !
;  509     0506  2          IF .CONV$GL_FILE_COUNT GTR 9
;  510     0507  2          THEN
;  511     0508  2              RETURN CONV$_INP_FILES;
;  512     0509  2
;  513     0510  2          ! The first argument is always the input file
;  514     0511  2          !
;  515     0512  2          IF NULLPARAMETER(1)
;  516     0513  2          THEN
;  517     0514  2              RETURN CONV$_ILL_VALUE
;  518     0515  2          ELSE
;  519     0516  2              CONV$AL_IN_FILE_NAM [ .CONV$GL_FILE_COUNT ] =
;  520     0517  2                      COPY_DESC( ACTUALPARAMETER(1) );
;  521     0518  2
;  522     0519  2          CONV$GL_FILE_COUNT = .CONV$GL_FILE_COUNT + 1;
;  523     0520  2
;  524     0521  2          RETURN SS$_NORMAL
;  525     0522  2
;  526     0523  1          END;

                                                        .TITLE  CONV$CALL VAX-11 CONVERT
```

```
                                .IDENT  \V04-000\

                                .PSECT  _CONV$PLIT,NOWRT,NOEXE,  SHR,  PIC,2

43  58  45  2E  00000 P.AAA:    .ASCII  \.EXC\                                  ;

                                .PSECT  _CONV$OWN,NOEXE,  PIC,2

                    00000 EXC_FAB:.BLKB   80
                 00 00050 SEQUENCE:
                                .BYTE   0                                       ;
                    00051       .BLKB   3
                    00054 OPTION_BLOCK:
                                .BLKB   80
                    000A4 COUNT_BLOCK:
                                .BLKB   20
                    000B8 STATISTICS_BLOCK:
                                .BLKB   20

                        OPTION_COUNT=       OPTION_BLOCK
                        COUNT_COUNT=        COUNT_BLOCK
                        CONV$GL_CREATE==    OPTION_BLOCK+4
                        CONV$GL_SHARE==     OPTION_BLOCK+8
                        CONV$GL_FAST==      OPTION_BLOCK+12
                        CONV$GL_MERGE==     OPTION_BLOCK+16
                        CONV$GL_APPEND==    OPTION_BLOCK+20
                        CONV$GL_SORT==      OPTION_BLOCK+24
                        CONV$GL_WORK_F==    OPTION_BLOCK+28
                        CONV$GL_KEY==       OPTION_BLOCK+32
                        CONV$GL_PAD==       OPTION_BLOCK+36
                        CONV$GL_PAD_CHAR==  OPTION_BLOCK+40
                        CONV$GL_TRUNCATE==  OPTION_BLOCK+44
                        CONV$GL_EXIT==      OPTION_BLOCK+48
                        CONV$GL_FIX==       OPTION_BLOCK+52
                        CONV$GL_FILL==      OPTION_BLOCK+56
                        CONV$GL_READ_C==    OPTION_BLOCK+60
                        CONV$GL_WRITE_C==   OPTION_BLOCK+64
                        CONV$GL_FDL==       OPTION_BLOCK+68
                        CONV$GL_EXC==       OPTION_BLOCK+72
                        CONV$GL_PROLOG==    OPTION_BLOCK+76
                        CONV$GL_FILE_COUNT==COUNT_BLOCK+4
                        CONV$GL_RECORD_COUNT==
                                            COUNT_BLOCK+8
                        CONV$GL_EXCEPT_COUNT==
                                            COUNT_BLOCK+12
                        CONV$GL_VALID_COUNT==
                                            COUNT_BLOCK+16
                        RECL$GL_BUCKET_COUNT==
                                            STATISTICS_BLOCK+4
                        RECL$GL_DATA_COUNT==STATISTICS_BLOCK+8
                        RECL$GL_INDEX_COUNT==
                                            STATISTICS_BLOCK+12
                        $RMS_PTR=           EXC_FAB
                                .EXTRN  CONVERT$_FACILITY
                                .EXTRN  CONV$_FAO_MAX, CONV$_BADBLK
                                .EXTRN  CONV$_BADLOGIC, CONV$_BADSORT
                                .EXTRN  CONV$_CONFQUAL, CONV$_CREATEDSTM
```

```
.EXTRN  CONV$_CREA_ERR, CONV$_DELPRI
.EXTRN  CONV$_DUP, CONV$_EXTN_ERR
.EXTRN  CONV$_FATALEXC, CONV$_FILLIM
.EXTRN  CONV$_IDX_LIM, CONV$_ILL_KEY
.EXTRN  CONV$_ILL_VALUE
.EXTRN  CONV$_INP_FILES
.EXTRN  CONV$_INSVIRMEM
.EXTRN  CONV$_INVBKT, CONV$_KEY
.EXTRN  CONV$_KEYREF, CONV$_LOADIDX
.EXTRN  CONV$_NARG, CONV$_NI
.EXTRN  CONV$_NOKEY, CONV$_NOTIDX
.EXTRN  CONV$_NOTSEQ, CONV$_NOWILD
.EXTRN  CONV$_ORDER, CONV$_OPENEXC
.EXTRN  CONV$_OPENIN, CONV$_OPENOUT
.EXTRN  CONV$_PAD, CONV$_PLV
.EXTRN  CONV$_PROERR, CONV$_PROL_WRT
.EXTRN  CONV$_READERR, CONV$_RSK
.EXTRN  CONV$_RSZ, CONV$_RTL
.EXTRN  CONV$_RTS, CONV$_SEQ
.EXTRN  CONV$_UDF_BKS, CONV$_UDF_BLK
.EXTRN  CONV$_VFC, CONV$_WRITEERR
.EXTRN  LIB$ESTABLISH, LIB$FIND_FILE_END
.EXTRN  LIB$SIG_TO_RET, STR$ANALYZE_SDESC_R1
.EXTRN  CONV$$CREATE_BUFFER
.EXTRN  CONV$$CONVERT, CONV$$END_OF_FILE
.EXTRN  CONV$$FREE_VM, CONV$$FREE_TEMP_VM
.EXTRN  CONV$$GET_NEXT_KEY
.EXTRN  CONV$$GET_VM, CONV$$OPEN_INPUT
.EXTRN  CONV$$OPEN_OUTPUT
.EXTRN  CONV$$PARSE_DEF
.EXTRN  CONV$$RMS_OPEN_ERROR
.EXTRN  CONV$$SET_KEY_DESC
.EXTRN  ADD$$CHECK_KEY, ADD$$LOAD_KEY
.EXTRN  ADD$$OPEN_OUTPUT
.EXTRN  RECL$$OPEN_FILE
.EXTRN  RECL$$ALLOCATE_BUFFERS
.EXTRN  RECL$$SCAN_DATA_LEVEL
.EXTRN  CONV$$PUT_RECORD
.EXTRN  CONV$AB_FLAGS, CONV$GB_CURRENT_FILE
.EXTRN  CONV$GL_FINDFILE_CTX
.EXTRN  CONV$AL_IN_FILE_NAM
.EXTRN  CONV$AR_OUT_FILE_NAM
.EXTRN  CONV$AR_FDL_FILE_NAM
.EXTRN  CONV$AB_EXC_RAB
.EXTRN  CONV$AB_IN_XABSUM
.EXTRN  CONV$AB_IN_XABFHC
.EXTRN  CONV$AB_IN_NAM, CONV$AB_IN_FAB
.EXTRN  CONV$AB_IN_RAB, CONV$AB_OUT_XABSUM
.EXTRN  CONV$AB_OUT_NAM
.EXTRN  CONV$AB_OUT_FAB
.EXTRN  CONV$AB_OUT_RAB
.EXTRN  CONV$AB_RFA_FAB
.EXTRN  CONV$AB_RFA_RAB
.EXTRN  CONV$GL_ADD_DELE_KEY
.EXTRN  CONV$GL_STM_BUF
.EXTRN  CONV$GL_STM_REC_LEN
.EXTRN  CONV$GW_OUT_REC_SIZ
```

```
                                                              .PSECT  _CONV$CODE,NOWRT,  SHR,  PIC,2

                                           0FFC 00000         .ENTRY  CONV$PASS_FILES, Save R2,R3,R4,R5,R6,R7,R8,-; 0243
                                                                      R9,R10,R11                                   :
                      5B      0000G   CF   9E 00002           MOVAB   $RMS_PTR, R11                                :
                      5A      0000G   CF   9E 00007           MOVAB   CONV$AB_IN_NAM, R10                          :
                      59      0000G   CF   9E 0000C           MOVAB   $RMS_PTR, R9                                 :
                      58      0000'   CF   9E 00011           MOVAB   CONV$GL_FILE_COUNT, R8                       :
                              0000V   CF   9F 00016           PUSHAB  CONDITION_HANDLER                            ; 0308
               00000000G 00           01   FB 0001A           CALLS   #1, LIB$ESTABLISH                            :
                      50         A8   A8   9A 00021           MOVZBL  SEQUENCE, R0                                 ; 0312
                                      03   13 00025           BEQL    1$                                           :
                                    0218   31 00027           BRW     11$                                          :
                      02              6C   91 0002A 1$:       CMPB    (AP), #2                                     ; 0322
                                      03   1E 0002D           BGEQU   3$                                           :
                                    0227   31 0002F 2$:       BRW     12$                                          :
                      05              6C   91 00032 3$:       CMPB    (AP), #5                                     :
                                      F8   1A 00035           BGTRU   2$                                           :
                              0000G   CF   D4 00037           CLRL    CONV$AB_FLAGS                                ; 0328
                      05              6C   91 0003B           CMPB    (AP), #5                                     ; 0332
                                      06   12 0003E           BNEQ    4$                                           :
                0000G CF         14   BC   B0 00040           MOVW    @20(AP), CONV$AB_FLAGS                       ; 0334
                      50       0140   8F   3C 00046 4$:       MOVZWL  #320, BYTES                                  ; 0338
                                      50   DD 0004B           PUSHL   BYTES                                        ; 0340
                                   0000G   30 0004D           BSBW    CONV$$GET_VM                                 :
                      5E              04   C0 00050           ADDL2   #4, SP                                       :
                      56              50   D0 00053           MOVL    R0, VM_POINTER                               :
  0050   8F        00 6E              00   2C 00056           MOVC5   #0, (SP), #0, #80, $RMS_PTR                  ; 0350
                                      69      0005D                                                                :
                      69       5003   8F   B0 0005E           MOVW    #20483, $RMS_PTR                             :
                   04 A9   01000040   8F   D0 00063           MOVL    #16777280, $RMS_PTR+4                        :
                   16 A9         42   8F   90 0006B           MOVB    #66, $RMS_PTR+22                             :
                   1F A9              02   90 00070           MOVB    #2, $RMS_PTR+31                              :
                   24 A9      0000G   CF   9E 00074           MOVAB   CONV$AB_IN_XABSUM, $RMS_PTR+36               :
                   28 A9              6A   9E 0007A           MOVAB   CONV$AB_IN_NAM, $RMS_PTR+40                  :
  0044   8F        00 6E              00   2C 0007E           MOVC5   #0, (SP), #0, #68, $RMS_PTR                  ; 0356
                              0000G   CF      00085                                                                :
                0000G CF       4401   8F   B0 00088           MOVW    #17409, $RMS_PTR                             :
                0000G CF       0200   8F   3C 0008F           MOVZWL  #512, $RMS_PTR+4                             :
                0000G CF              69   9E 00096           MOVAB   CONV$AB_IN_FAB, $RMS_PTR+60                  :
  0060   8F        00 6E              00   2C 0009B           MOVC5   #0, (SP), #0, #96, $RMS_PTR                  ; 0364
                                      6A      000A2                                                                :
                      6A       6002   8F   B0 000A3           MOVW    #24578, $RMS_PTR                             :
                   02 AA         50   8F   90 000A8           MOVB    #80, $RMS_PTR+2                              :
                   04 AA         50   A6   9E 000AD           MOVAB   80(R6), $RMS_PTR+4                           :
                   0A AA         50   8F   90 000B2           MOVB    #80, $RMS_PTR+10                             :
                   0C AA              56   D0 000B7           MOVL    VM_POINTER, $RMS_PTR+12                      :
         0C        00 6E              00   2C 000BB           MOVC5   #0, (SP), #0, #12, $RMS_PTR                  ; 0369
                              0000G   CF      000C0                                                                :
                0000G CF       0C16   8F   B0 000C3           MOVW    #3094, $RMS_PTR                              :
                0000G CF      0000G   CF   9E 000CA           MOVAB   CONV$AB_IN_XABFHC, $RMS_PTR+4                :
         2C        00 6E              00   2C 000D1           MOVC5   #0, (SP), #0, #44, $RMS_PTR                  ; 0371
                              0000G   CF      000D6                                                                :
                0000G CF       2C1D   8F   B0 000D9           MOVW    #11293, $RMS_PTR                             :
                      56       00A0   C6   9E 000E0           MOVAB   160(R6), VM_POINTER                          ; 0375
  0050   8F        00 6E              00   2C 000E5           MOVC5   #0, (SP), #0, #80, $RMS_PTR                  ; 0385
```

```
                                         6B      000EC                                              ;
                         6B      5003    8F  B0  000ED        MOVW    #20483, $RMS_PTR               ;
                    04   AB  21000060    8F  D0  000F2        MOVL    #553648224, $RMS_PTR+4         ;
                    16   AB        43    8F  90  000FA        MOVB    #67, $RMS_PTR+22               ;
                    1F   AB              02  90  000FF        MOVB    #2, $RMS_PTR+31                ;
                    24   AB      0000G   CF  9E  00103        MOVAB   CONV$AB_OUT_XABSUM, $RMS_PTR+36 ;
                    28   AB      0000G   CF  9E  00109        MOVAB   CONV$AB_OUT_NAM, $RMS_PTR+40   ;
0044   8F        00              6E      00  2C  0010F        MOVC5   #0, (SP), #0, #68, $RMS_PTR    ; 0391
                                 0000G   CF      00116                                              ;
                 0000G   CF      4401    8F  B0  00119        MOVW    #17409, $RMS_PTR               ;
                 0000G   CF      0400    8F  3C  00120        MOVZWL  #1024, $RMS_PTR+4              ;
                 0000G   CF              6B  9E  00127        MOVAB   CONV$AB_OUT_FAB, $RMS_PTR+60   ;
0060   8F        00              6E      00  2C  0012C        MOVC5   #0, (SP), #0, #96, $RMS_PTR    ; 0400
                                 0000G   CF      00133                                              ;
                 0000G   CF      6002    8F  B0  00136        MOVW    #24578, $RMS_PTR               ;
                 0000G   CF        50    8F  90  0013D        MOVB    #80, $RMS_PTR+2                ;
                 0000G   CF        50    A6  9E  00143        MOVAB   80(R6), $RMS_PTR+4             ;
                 0000G   CF        50    8F  90  00149        MOVB    #80, $RMS_PTR+10               ;
                 0000G   CF              56  D0  0014F        MOVL    VM_POINTER, $RMS_PTR+12        ;
                 0000G   CF              6A  9E  00154        MOVAB   CONV$AB_IN_NAM, $RMS_PTR+16    ;
       0C        00              6E      00  2C  00159        MOVC5   #0, (SP), #0, #12, $RMS_PTR    ; 0405
                                 0000G   CF      0015E                                              ;
                 0000G   CF      0C16    8F  B0  00161        MOVW    #3094, $RMS_PTR                ;
                                         68  D4  00168        CLRL    CONV$GL_FILE_COUNT             ; 0409
                         02              6C  91  0016A        CMPB    (AP), #2                       ; 0413
                                         03  1E  0016D        BGEQU   6$                             ;
                                       0110  31  0016F 5$:    BRW     16$                            ;
                                   08    AC  D5  00172 6$:    TSTL    8(AP)                          ;
                                         F8  13  00175        BEQL    5$                             ;
                         50        08    AC  D0  00177        MOVL    8(AP), R0                      ; 0417
                                      0000V  30  0017B        BSBW    COPY_DESC                      ;
                 0000G   CF              50  D0  0017E        MOVL    R0, CONV$AR_OUT_FILE_NAM       ;
                         03              6C  91  00183        CMPB    (AP), #3                       ; 0421
                                         11  1F  00186        BLSSU   7$                             ;
                                   0C    AC  D5  00188        TSTL    12(AP)                         ;
                                         0C  13  0018B        BEQL    7$                             ;
                         50        0C    AC  D0  0018D        MOVL    12(AP), R0                     ; 0426
                                      0000V  30  00191        BSBW    COPY_DESC                      ;
                 0000G   CF              50  D0  00194        MOVL    R0, CONV$AR_FDL_FILE_NAM       ;
                         04              6C  91  00199 7$:    CMPB    (AP), #4                       ; 0431
                                         03  1E  0019C        BGEQU   9$                             ;
                                       009B  31  0019E 8$:    BRW     10$                            ;
                                   10    AC  D5  001A1 9$:    TSTL    16(AP)                         ;
                                         F8  13  001A4        BEQL    8$                             ;
                         50        10    AC  D0  001A6        MOVL    16(AP), R0                     ; 0439
                                      0000V  30  001AA        BSBW    COPY_DESC                      ;
                         57              50  D0  001AD        MOVL    R0, EXC_FILE_NAM               ;
                         7E      0100    8F  3C  001B0        MOVZWL  #256, -(SP)                    ; 0443
                                      0000G  30  001B5        BSBW    CONV$$GET_VM                   ;
                         5E              04  C0  001B8        ADDL2   #4, SP                         ;
                         56              50  D0  001BB        MOVL    R0, EXC_NAM_PTR                ;
0050   8F        00              6E      00  2C  001BE        MOVC5   #0, (SP), #0, #80, $RMS_PTR    ; 0456
                                 FF58    C8      001C5                                              ;
                 FF58    C8      5003    8F  B0  001C8        MOVW    #20483, $RMS_PTR               ;
                 FF5C    C8  11000040    8F  D0  001CF        MOVL    #285212736, $RMS_PTR+4         ;
                 FF6E    C8              02  90  001D8        MOVB    #2, $RMS_PTR+22                ;
                 FF76    C8      0202    8F  B0  001DD        MOVW    #514, $RMS_PTR+30              ;
```

```
                        80  A8               56  D0 001E4            MOVL    EXC_NAM_PTR, $RMS_PTR+40                         ;
                        84  A8         04    A7  D0 001E8            MOVL    4(EXC_FILE_NAM), $RMS_PTR+44                     ;
                        88  A8       0000'   CF  9E 001ED            MOVAB   P.AAA, $RMS_PTR+48                               ;
                        8C  A8               67  90 001F3            MOVB    (EXC_FILE_NAM), $RMS_PTR+52                      ;
                        8D  A8               04  90 001F7            MOVB    #4, $RMS_PTR+53                                  ;
   0044  8F       00        6E               00  2C 001FB            MOVC5   #0, (SP), #0, #68, $RMS_PTR                      ; 0462
                                     0000G   CF     00202                                                                     ;
                     0000G  CF       4401    8F  B0 00205            MOVW    #17409, $RMS_PTR                                 ;
                     0000G  CF       0400    8F  3C 0020C            MOVZWL  #1024, $RMS_PTR+4                                ;
                     0000G  CF       FF58    C8  9E 00213            MOVAB   EXC_FAB, $RMS_PTR+60                             ;
   0060  8F       00        6E               00  2C 0021A            MOVC5   #0, (SP), #0, #96, (EXC_NAM_PTR)                 ; 0470
                                             66     00221                                                                     ;
                            66       6002    8F  B0 00222            MOVW    #24578, (EXC_NAM_PTR)                            ;
                        02  A6         50    8F  90 00227            MOVB    #80, 2(EXC_NAM_PTR)                              ;
                        04  A6       00B0    C6  9E 0022C            MOVAB   176(R6), 4(EXC_NAM_PTR)                          ;
                        0A  A6         50    8F  90 00232            MOVB    #80, 10(EXC_NAM_PTR)                             ;
                        0C  A6         60    A6  9E 00237            MOVAB   96(R6), 12(EXC_NAM_PTR)                          ;
                        A8  A8               01  90 0023C  10$:      MOVB    #1, SEQUENCE                                     ; 0476
                                             27  11 00240            BRB     14$                                              ;
                            01               50  91 00242  11$:      CMPB    R0, #1                                           ; 0482
                                             1A  12 00245            BNEQ    13$                                              ;
                            01               6C  91 00247            CMPB    (AP), #1                                         ; 0488
                                             1D  13 0024A            BEQL    14$                                              ;
                            02               6C  91 0024C            CMPB    (AP), #2                                         ; 0491
                                             08  12 0024F            BNEQ    12$                                              ;
                     0000G  CF         08    BC  B0 00251            MOVW    @8(AP), CONV$AB_FLAGS                            ; 0493
                                             10  11 00257            BRB     14$                                              ;
                            50  00000000G    8F  D0 00259  12$:      MOVL    #CONV$_NARG, R0                                  ; 0495
                                             04     00260            RET                                                      ;
                            50  00000000G    8F  D0 00261  13$:      MOVL    #CONV$_ORDER, R0                                 ; 0502
                                             04     00268            RET                                                      ;
                            52               68  D0 00269  14$:      MOVL    CONV$GL_FILE_COUNT, R2                           ; 0506
                            09               52  D1 0026C            CMPL    R2, #9                                           ;
                                             08  15 0026F            BLEQ    15$                                              ;
                            50  00000000G    8F  D0 00271            MOVL    #CONV$_INP_FILES, R0                             ; 0508
                                             04     00278            RET                                                      ;
                                             6C  95 00279  15$:      TSTB    (AP)                                             ; 0512
                                             05  13 0027B            BEQL    16$                                              ;
                                       04    AC  D5 0027D            TSTL    4(AP)                                            ;
                                             08  12 00280            BNEQ    17$                                              ;
                            50  00000000G    8F  D0 00282  16$:      MOVL    #CONV$_ILL_VALUE, R0                             ; 0514
                                             04     00289            RET                                                      ;
                            50         04    AC  D0 0028A  17$:      MOVL    4(AP), R0                                        ; 0517
                                          0000V  30 0028E            BSBW    COPY_DESC                                        ;
                     0000GCF42               50  D0 00291            MOVL    R0, CONV$AL_IN_FILE_NAM[R2]                      ;
                                             68  D6 00297            INCL    CONV$GL_FILE_COUNT                               ; 0519
                            50               01  D0 00299            MOVL    #1, R0                                           ; 0521
                                             04     0029C            RET                                                      ; 0523

; Routine Size:  669 bytes,    Routine Base:  _CONV$CODE + 0000
```

```
;  1 %SBTTL  'PASS_OPTIONS'
;  1 GLOBAL ROUTINE CONV$PASS_OPTIONS =
;  1 !++
;  1 !
;  1 ! Functional Description:
;  1 !
;  1 !        Initializes the convert control/option block
;  1 !
;  1 ! Calling Sequence:
;  1 !
;  1 !        CONV$PASS_OPTIONS( [option_block][,flags] )
;  1 !
;  1 ! Input Parameters:
;  1 !
;  1 !        option_block    - ( Optional ) Address of convert option block
;  1 !
;  1 !            Structure of option block:
;  1 !
;  1 !                           !--------------------------------!
;  1 !            option_block -->!    number of options           !
;  1 !                           !--------------------------------!
;  1 !                           !    create                      !
;  1 !                           !--------------------------------!
;  1 !                           !    share                       !
;  1 !                           !--------------------------------!
;  1 !                           !    fast                        !
;  1 !                           !--------------------------------!
;  1 !                           !    merge                       !
;  1 !                           !--------------------------------!
;  1 !                           !    append                      !
;  1 !                           !--------------------------------!
;  1 !                           !    sort                        !
;  1 !                           !--------------------------------!
;  1 !                           !    work_files                  !
;  1 !                           !--------------------------------!
;  1 !                           !    key                         !
;  1 !                           !--------------------------------!
;  1 !                           !    pad                         !
;  1 !                           !--------------------------------!
;  1 !                           !    pad_character               !
;  1 !                           !--------------------------------!
;  1 !                           !    truncate                    !
;  1 !                           !--------------------------------!
;  1 !                           !    exit                        !
;  1 !                           !--------------------------------!
;  1 !                           !    fixed_write                 !
;  1 !                           !--------------------------------!
;  1 !                           !    fill_buckets                !
;  1 !                           !--------------------------------!
;  1 !                           !    read_check                  !
;  1 !                           !--------------------------------!
;  1 !                           !    write_check                 !
;  1 !                           !--------------------------------!
;  1 !                           !    fdl                         !
;  1 !                           !--------------------------------!
;  1 !                           !    exception                   !
;  1 !                           !--------------------------------!
```

```
!                                   |    prologue                |
!                                   |----------------------------|
!
!       flags               - ( Optional ) flags longword
!
! Implicit Inputs:
!       none
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!
!       SS$NORMAL or error code
!
! Routines Called:
!
!       $OPEN
!       CONV$$RMS_OPEN_ERROR    - By RMS as an AST
!       $CONNECT
!
! Side Effects:
!
!       Opens the exception file if specified
!
!--

    BEGIN

    BUILTIN
        ACTUALCOUNT,
        ACTUALPARAMETER,
        NULLPARAMETER;

    ! Set up condition handler
    !
    LIB$ESTABLISH ( CONDITION_HANDLER );

    ! Check the order of the calls
    !
    IF .SEQUENCE NEQ 1
    THEN
        RETURN CONV$_ORDER;

    ! Check the number of arguments
    !
    IF ACTUALCOUNT() GTRU 2
    THEN
        RETURN CONV$_NARG;

    ! Were there user flags?
    !
    IF ACTUALCOUNT() EQLU 2
    THEN
```

```
;  642          0638  2          CONV$AB_FLAGS [ CONV$W_USER ] = .ACTUALPARAMETER(2);
;  643          0639  2
;  644          0640  2      ! Initialize the counter block
;  645          0641  2      !
;  646          0642  2      COUNT_COUNT = COUNTERS;
;  647          0643  2
;  648          0644  2      ! Clear the counters (don't clear 2 (file_count) since that has
;  649          0645  2      ! been set by pass_files)
;  650          0646  2      !
;  651          0647  2      INCR I FROM 2 TO COUNTERS BY 1
;  652          0648  2      DO
;  653          0649  2          COUNT_BLOCK [ .I ] = _CLEAR;
;  654          0650  2
;  655          0651  2      ! Initialize the option block
;  656          0652  2      !
;  657          0653  2      OPTION_COUNT = OPTIONS;
;  658          0654  2
;  659          0655  2      ! First clear the entire block since most of the defaults are off
;  660          0656  2      !
;  661          0657  2      INCR I FROM 1 TO OPTIONS BY 1
;  662          0658  2      DO
;  663          0659  2          OPTION_BLOCK [ .I ] = _CLEAR;
;  664          0660  2
;  665          0661  2      ! Now set the defaults
;  666          0662  2      !
;  667          0663  2      CONV$GL_CREATE      = _SET;
;  668          0664  2      CONV$GL_FAST        = _SET;
;  669          0665  2      CONV$GL_SORT        = _SET;
;  670          0666  2      CONV$GL_WORK_F      = 2;
;  671          0667  2
;  672          0668  2      ! If there was an argument then use it
;  673          0669  2      !
;  674          0670  2      IF NOT NULLPARAMETER(1)
;  675          0671  2      THEN
;  676          0672  3          BEGIN
;  677          0673  3
;  678          0674  3          LOCAL   USER_BLOCK      : REF VECTOR [ ,LONG ];
;  679          0675  3
;  680          0676  3          USER_BLOCK = ACTUALPARAMETER( 1 );
;  681          0677  3
;  682          0678  3          ! Check the size of the block
;  683          0679  3          !
;  684          0680  3          IF .USER_BLOCK [ 0 ] GTRU OPTIONS
;  685          0681  3          THEN
;  686          0682  3              RETURN CONV$_BADBLK;
;  687          0683  3
;  688          0684  3          ! If the user block specified a prologue version then set a flag to
;  689          0685  3          ! use it
;  690          0686  3          !
;  691          0687  3          IF .USER_BLOCK [ 0 ] EQLU OPTIONS
;  692          0688  3          THEN
;  693          0689  3              CONV$AB_FLAGS [ CONV$V_PROLOG ] = _SET;
;  694          0690  3
;  695          0691  3          ! Copy the option block specified by the user into ours
;  696          0692  3          !
;  697          0693  3          INCR I FROM 1 TO .USER_BLOCK [ 0 ]
;  698          0694  3          DO
```

```
;  699     0695  3            OPTION_BLOCK [ .I ] = .USER_BLOCK [ .I ]
;  700     0696  3
;  701     0697  2        END;
;  702     0698  2
;  703     0699  2        ! Check for some switch conflicts
;  704     0700  2        !
;  705     0701  2        ! /FDL/NOCREATE
;  706     0702  2        ! /FDL/MERGE
;  707     0703  2        ! /FDL/APPEND
;  708     0704  2        !
;  709     0705  2        !    or combinations of the above is wrong
;  710     0706  2        !
;  711     0707  3        IF .CONV$GL_FDL AND ( ( NOT .CONV$GL_CREATE ) OR
;  712     0708  3                              .CONV$GL_MERGE OR
;  713     0709  3                              .CONV$GL_APPEND )
;  714     0710  2        THEN
;  715     0711  2            RETURN CONV$_CONFQUAL;
;  716     0712  2
;  717     0713  2        ! /MERGE/APPEND
;  718     0714  2        !
;  719     0715  2        IF .CONV$GL_MERGE AND .CONV$GL_APPEND
;  720     0716  2        THEN
;  721     0717  2            RETURN CONV$_CONFQUAL;
;  722     0718  2
;  723     0719  2        ! Lets set the switches strait, NOTE: The order of this sets the presidence
;  724     0720  2        !                                     of the qualifiers, do not change it
;  725     0721  2        !
;  726     0722  2        ! The merge option is really /NOCREATE/NOFAST/NOSORT apppend is simmular
;  727     0723  2        !
;  728     0724  2        IF .CONV$GL_MERGE OR .CONV$GL_APPEND
;  729     0725  2        THEN
;  730     0726  3            BEGIN
;  731     0727  3            CONV$GL_CREATE  = _CLEAR;
;  732     0728  3            CONV$GL_FAST    = _CLEAR;
;  733     0729  3            CONV$GL_SORT    = _CLEAR
;  734     0730  2            END;
;  735     0731  2
;  736     0732  2        ! If we create a file without definition then the files are duplicate
;  737     0733  2        ! therefore index files will be in order (one input file only)
;  738     0734  2        !
;  739     0735  2        IF .CONV$GL_CREATE AND ( NOT .CONV$GL_FDL ) AND
;  740     0736  3                                 ( .CONV$GL_FILE_COUNT EQLU 1 )
;  741     0737  2        THEN
;  742     0738  2            CONV$GL_SORT    = _CLEAR;
;  743     0739  2
;  744     0740  2        ! If we open the input file shared we cannot sort it
;  745     0741  2        !
;  746     0742  2        IF .CONV$GL_SHARE
;  747     0743  2        THEN
;  748     0744  2            CONV$GL_SORT    = _CLEAR;
;  749     0745  2
;  750     0746  2        ! Create exc files if neceassary
;  751     0747  2        !
;  752     0748  2        ! If the EXCEPTION Option was specified THEN Create and Connect
;  753     0749  2        ! the Exception File.
;  754     0750  2        !
;  755     0751  2        IF .CONV$GL_EXC
```

```
0752  2     THEN
0753  3         BEGIN
0754  3
0755  3         ! Signal an OPENEXC error on failure
0756  3         !
0757  3         EXC_FAB [ FAB$L_CTX ] = CONV$_OPENEXC;
0758  3
0759  3         ! Create it
0760  3         !
0761  3         $CREATE ( FAB=EXC_FAB,ERR=CONV$$RMS_OPEN_ERROR );
0762  3
0763  3         ! Say that a exception file is open
0764  3         !
0765  3         CONV$AB_FLAGS [ CONV$V_EXC ] = _SET;
0766  3
0767  3         ! Connect the stream
0768  3         !
0769  4         $CONNECT ( RAB=CONV$AB_EXC_RAB,ERR=CONV$$RMS_OPEN_ERROR )
0770  4
0771  2         END;
0772  2
0773  2     SEQUENCE = .SEQUENCE + 1;
0774  2
0775  2     RETURN CONV$_SUCCESS
0776  2
0777  1     END;



                                                              .EXTRN   SYS$CREATE, SYS$CONNECT

                                      0004 00000              .ENTRY   CONV$PASS_OPTIONS, Save R2                   ; 0525
                52      0000'    CF   9E 00002                MOVAB    CONV$GL_CREATE, R2
                        0000V    CF   9F 00007                PUSHAB   CONDITION_HANDLER                            ; 0620
       00000000G  00              01  FB 0000B                CALLS    #1, LIB$ESTABLISH
                  01      F8     A2   91 00012                CMPB     SEQUENCE, #1                                 ; 0624
                                  08  13 00016                BEQL     1$
                50 00000000G     8F   D0 00018                MOVL     #CONV$_ORDER, R0                             ; 0626
                                      04 0001F                RET
                02               6C   91 00020  1$:           CMPB     (AP), #2                                     ; 0630
                                  08  1B 00023                BLEQU    2$
                50 00000000G     8F   D0 00025                MOVL     #CONV$_NARG, R0                              ; 0632
                                      04 0002C                RET
                                  06  12 0002D  2$:           BNEQ     3$                                           ; 0636
            0000G  CF     08     BC   B0 0002F                MOVW     @8(AP), CONV$AB_FLAGS                        ; 0638
               4C  A2            04   D0 00035  3$:           MOVL     #4, COUNT_COUNT                              ; 0642
                   50            02   D0 00039                MOVL     #2, I                                        ; 0647
                         4C A240  D4 0003C  4$:           CLRL     COUNT_BLOCK[I]                               ; 0649
          F8       50            04   F3 00040                AOBLEQ   #4, I, 4$
               FC  A2            13   D0 00044                MOVL     #19, OPTION_COUNT                            ; 0653
                   50            01   D0 00048                MOVL     #1, I                                        ; 0657
                         FC A240  D4 0004B  5$:           CLRL     OPTION_BLOCK[I]                              ; 0659
          F8       50            13   F3 0004F                AOBLEQ   #19, I, 5$
                   62            01   D0 00053                MOVL     #1, CONV$GL_CREATE                           ; 0663
               08  A2            01   D0 00056                MOVL     #1, CONV$GL_FAST                             ; 0664
               14  A2            01   D0 0005A                MOVL     #1, CONV$GL_SORT                             ; 0665
               18  A2            02   D0 0005E                MOVL     #2, CONV$GL_WORK_F                           ; 0666
```

```
                                    6C  95 00062          TSTB    (AP)                                    ; 0670
                                    2C  13 00064          BEQL    10$                                     ;
                              04    AC  D5 00066          TSTL    4(AP)                                   ;
                                    27  13 00069          BEQL    10$                                     ;
                         51   04    AC  D0 0006B          MOVL    4(AP), USER_BLOCK                       ; 0676
                         13         61  D1 0006F          CMPL    (USER_BLOCK), #19                       ; 0680
                                    08  1B 00072          BLEQU   6$                                      ;
                         50 00000000G 8F  D0 00074          MOVL    #CONV$_BADBLK, R0                       ; 0682
                                    04  00007B           RET                                             ;
                                    06  12 0007C 6$:      BNEQ    7$                                      ; 0687
                   0000G CF         40  8F 88 0007E      BISB2   #64, CONV$AB_FLAGS+2                    ; 0689
                                    50  D4 00084 7$:      CLRL    I                                       ; 0693
                                    06  11 00086          BRB     9$                                      ;
                 FC A240              6140 D0 00088 8$:      MOVL    (USER_BLOCK)[I], OPTION_BLOCK[I]        ; 0695
              F6      50            61  F3 0008E 9$:      AOBLEQ  (USER_BLOCK), I, 8$                     ;
                      0B    40      A2  E9 00092 10$:     BLBC    CONV$GL_FDL, 11$                        ; 0707
                      10            62  E9 00096          BLBC    CONV$GL_CREATE, 12$                     ;
                      0C    0C      A2  E8 00099          BLBS    CONV$GL_MERGE, 12$                      ; 0708
                      08    10      A2  E8 0009D          BLBS    CONV$GL_APPEND, 12$                     ; 0709
                      10    0C      A2  E9 000A1 11$:     BLBC    CONV$GL_MERGE, 14$                      ; 0715
                      08    10      A2  E9 000A5          BLBC    CONV$GL_APPEND, 13$                     ;
                      50 00000000G  8F  D0 000A9 12$:     MOVL    #CONV$_CONFQUAL, R0                     ; 0717
                                    04  000B0           RET                                             ;
                      04    0C      A2  E8 000B1 13$:     BLBS    CONV$GL_MERGE, 15$                      ; 0724
                      08    10      A2  E9 000B5 14$:     BLBC    CONV$GL_APPEND, 16$                     ;
                                    62  D4 000B9 15$:     CLRL    CONV$GL_CREATE                          ; 0727
                              08    A2  D4 000BB          CLRL    CONV$GL_FAST                            ; 0728
                              14    A2  D4 000BE          CLRL    CONV$GL_SORT                            ; 0729
                      0D            62  E9 000C1 16$:     BLBC    CONV$GL_CREATE, 17$                     ; 0735
                      09    40      A2  E8 000C4          BLBS    CONV$GL_FDL, 17$                        ;
                      01    50      A2  D1 000C8          CMPL    CONV$GL_FILE_COUNT, #1                  ; 0736
                                    03  12 000CC          BNEQ    17$                                     ;
                              14    A2  D4 000CE          CLRL    CONV$GL_SORT                            ; 0738
                         03   04    A2  E9 000D1 17$:     BLBC    CONV$GL_SHARE, 18$                      ; 0742
                              14    A2  D4 000D5          CLRL    CONV$GL_SORT                            ; 0744
                         2A   44    A2  E9 000D8 18$:     BLBC    CONV$GL_EXC, 19$                        ; 0751
                   C0    A2 00000000G  8F  D0 000DC          MOVL    #CONV$_OPENEXC, EXC_FAB+24              ; 0757
                                0000G  CF  9F 000E4          PUSHAB  CONV$$RMS_OPEN_ERROR                    ; 0761
                                A8     A2  9F 000E8          PUSHAB  EXC_FAB                                 ;
                00000000G       00         02  FB 000EB          CALLS   #2, SYS$CREATE                          ;
                    0000G       CF         04  88 000F2          BISB2   #4, CONV$AB_FLAGS+2                     ; 0765
                                0000G      CF  9F 000F7          PUSHAB  CONV$$RMS_OPEN_ERROR                    ; 0769
                                0000G      CF  9F 000FB          PUSHAB  CONV$AB_EXC_RAB                         ;
                00000000G       00         02  FB 000FF          CALLS   #2, SYS$CONNECT                         ;
                                F8     A2  96 00106 19$:     INCB    SEQUENCE                                ; 0773
                         50             01  D0 00109          MOVL    #1, R0                                  ; 0775
                                        04  0010C           RET                                             ; 0777

; Routine Size:  269 bytes,    Routine Base:  _CONV$CODE + 029D
```

```
;   783          0778  1 %SBTTL  'CONVERT'
;   784          0779  1 GLOBAL ROUTINE  CONV$CONVERT =
;   785          0780  1 !++
;   786          0781  1 !
;   787          0782  1 ! Functional Description:
;   788          0783  1 !
;   789          0784  1 ! Calling Sequence:
;   790          0785  1 !
;   791          0786  1 !       CONV$CONVERT( [counter_block][,flags] )
;   792          0787  1 !
;   793          0788  1 ! Input Parameters:
;   794          0789  1 !
;   795          0790  1 !       counter_block   - ( Optional ) Address of counter block
;   796          0791  1 !
;   797          0792  1 !           Structure of counter block:
;   798          0793  1 !
;   799          0794  1 !                                   |-------------------------|
;   800          0795  1 !           counter_block -->       |  number of counters     |
;   801          0796  1 !                                   |-------------------------|
;   802          0797  1 !                                   |  number of files proc.  |
;   803          0798  1 !                                   |-------------------------|
;   804          0799  1 !                                   |  number of records      |
;   805          0800  1 !                                   |-------------------------|
;   806          0801  1 !                                   |  number of exception rec.|
;   807          0802  1 !                                   |-------------------------|
;   808          0803  1 !                                   |  number of valid records|
;   809          0804  1 !                                   |-------------------------|
;   810          0805  1 !
;   811          0806  1 !       flags           - ( Optional ) Flags longword
;   812          0807  1 !
;   813          0808  1 ! Implicit Inputs:
;   814          0809  1 !       none
;   815          0810  1 !
;   816          0811  1 ! Output Parameters:
;   817          0812  1 !       none
;   818          0813  1 !
;   819          0814  1 ! Implicit Outputs:
;   820          0815  1 !       none
;   821          0816  1 !
;   822          0817  1 ! Routine Value:
;   823          0818  1 !       none
;   824          0819  1 !
;   825          0820  1 ! Routines called:
;   826          0821  1 !
;   827          0822  1 ! Side Effects:
;   828          0823  1 !       none
;   829          0824  1 !
;   830          0825  1 !--
;   831          0826  1
;   832          0827  2     BEGIN
;   833          0828  2
;   834          0829  2     BUILTIN
;   835          0830  2         ACTUALCOUNT,
;   836          0831  2         ACTUALPARAMETER,
;   837          0832  2         NULLPARAMETER;
;   838          0833  2
;   839          0834  2     LOCAL
```

```
;  840     0835  2          STATUS;
;  841     0836  2
;  842     0837  2      ! Set up condition handler
;  843     0838  2      !
;  844     0839  2      LIB$ESTABLISH ( CONDITION_HANDLER );
;  845     0840  2
;  846     0841  2      ! Check the order of the call
;  847     0842  2      !
;  848     0843  2      IF .SEQUENCE NEQ 2
;  849     0844  2      THEN
;  850     0845  2          RETURN CONV$_ORDER
;  851     0846  2      ELSE
;  852     0847  2          SEQUENCE = .SEQUENCE + 1;
;  853     0848  2
;  854     0849  2      ! Check the number of arguments
;  855     0850  2      !
;  856     0851  2      IF ACTUALCOUNT() GTRU 2
;  857     0852  2      THEN
;  858     0853  2          RETURN CONV$_NARG;
;  859     0854  2
;  860     0855  2      ! Were there user flags?
;  861     0856  2      !
;  862     0857  2      IF ACTUALCOUNT() EQLU 2
;  863     0858  2      THEN
;  864     0859  2          CONV$AB_FLAGS [ CONV$W_USER ] = .ACTUALPARAMETER(2);
;  865     0860  2
;  866     0861  2      ! Clear some variables
;  867     0862  2      !
;  868     0863  2      CONV$GB_CURRENT_FILE = 0;
;  869     0864  2
;  870     0865  2      ! If definition then parse it
;  871     0866  2      !
;  872     0867  2      IF .CONV$GL_FDL
;  873     0868  2      THEN
;  874     0869  2          STATUS = CONV$$PARSE_DEF()
;  875     0870  2      ELSE
;  876     0871  2          STATUS = CONV$_SUCCESS;
;  877     0872  2
;  878     0873  2      ! If all is well continue
;  879     0874  2      !
;  880     0875  2      IF .STATUS
;  881     0876  2      THEN
;  882     0877  2
;  883     0878  2          ! Try to Open an Input File
;  884     0879  2          !
;  885     0880  2          IF STATUS = CONV$$OPEN_INPUT()
;  886     0881  2          THEN
;  887     0882  2              BEGIN
;  888     0883  2
;  889     0884  2              ! Try to Open an Output File
;  890     0885  2              !
;  891     0886  2              STATUS = CONV$$OPEN_OUTPUT();
;  892     0887  2
;  893     0888  2              ! Loop untill error or end
;  894     0889  2              !
;  895     0890  3              WHILE .STATUS
;  896     0891  3              DO
```

```
;  897     0892  4                BEGIN
;  898     0893  4
;  899     0894  4                ! Dynamicly Allocate the Record Buffer
;  900     0895  4                !
;  901     0896  5                IF NOT ( STATUS = CONV$$CREATE_BUFFER() )
;  902     0897  4                THEN
;  903     0898  4                    EXITLOOP;
;  904     0899  4
;  905     0900  4                ! Convert The File
;  906     0901  4                !
;  907     0902  5                IF NOT ( STATUS = CONV$$CONVERT() )
;  908     0903  4                THEN
;  909     0904  4                    EXITLOOP;
;  910     0905  4
;  911     0906  4                CONV$GB_CURRENT_FILE = .CONV$GB_CURRENT_FILE + 1;
;  912     0907  4
;  913     0908  4                IF .CONV$GB_CURRENT_FILE GEQU .CONV$GL_FILE_COUNT
;  914     0909  4                THEN
;  915     0910  4                    EXITLOOP
;  916     0911  4                ELSE
;  917     0912  4                    STATUS = CONV$$OPEN_INPUT()
;  918     0913  4
;  919     0914  4                END
;  920     0915  4
;  921     0916  2            END;
;  922     0917  2
;  923     0918  2        ! Close all Open Files and deallocate memory
;  924     0919  2        !
;  925     0920  2        RUNDOWN();
;  926     0921  2
;  927     0922  2        ! If we got a counter block copy the values into it
;  928     0923  2        !
;  929     0924  2        IF NOT NULLPARAMETER(1)
;  930     0925  2        THEN
;  931     0926  2            BEGIN
;  932     0927  2
;  933     0928  2            LOCAL   USER_BLOCK : REF VECTOR [ ,LONG ];
;  934     0929  2
;  935     0930  2            USER_BLOCK = ACTUALPARAMETER(1);
;  936     0931  2
;  937     0932  2            ! Check the size of the block
;  938     0933  2            !
;  939     0934  2            IF .USER_BLOCK [ 0 ] GTRU COUNTERS
;  940     0935  2            THEN
;  941     0936  2                STATUS = CONV$_BADBLK
;  942     0937  2            ELSE
;  943     0938  2
;  944     0939  2                ! Stuff the counts
;  945     0940  2                !
;  946     0941  2                INCR I FROM 1 TO .USER_BLOCK [ 0 ] BY 1
;  947     0942  2                DO
;  948     0943  2                    USER_BLOCK [ .I ] = .COUNT_BLOCK [ .I ]
;  949     0944  2
;  950     0945  2            END;
;  951     0946  2
;  952     0947  2        SEQUENCE = 0;
;  953     0948  2
```

```
;  954       0949  2      RETURN .STATUS
;  955       0950  2
;  956       0951  1      END;



                                                    001C 00000          .ENTRY  CONV$CONVERT, Save R2,R3,R4                 ; 0779
                               54     0000G   CF    9E 00002           MOVAB   CONV$GB_CURRENT_FILE, R4                     ;
                               53     0000'   CF    9E 00007           MOVAB   SEQUENCE, R3                                 ;
                                      0000V   CF    9F 0000C           PUSHAB  CONDITION_HANDLER                            ; 0839
                  00000000G    00             01    FB 00010           CALLS   #1, LIB$ESTABLISH                            ;
                               02             63    91 00017           CMPB    SEQUENCE, #2                                 ; 0843
                                              08    13 0001A           BEQL    1$                                           ;
                               50 00000000G   8F    D0 0001C           MOVL    #CONV$_ORDER, R0                             ; 0845
                                                    04 00023           RET                                                  ;
                                              63    96 00024  1$:      INCB    SEQUENCE                                     ; 0847
                               02             6C    91 00026           CMPB    (AP), #2                                     ; 0851
                                              08    1B 00029           BLEQU   2$                                           ;
                               50 00000000G   8F    D0 0002B           MOVL    #CONV$_NARG, R0                              ; 0853
                                                    04 00032           RET                                                  ;
                                              06    12 00033  2$:      BNEQ    3$                                           ; 0857
                      0000G    CF       08    BC    B0 00035           MOVW    @8(AP), CONV$AB_FLAGS                        ; 0859
                                              64    94 0003B  3$:      CLRB    CONV$GB_CURRENT_FILE                         ; 0863
                               0A       48    A3    E9 0003D           BLBC    CONV$GL_FDL, 4$                              ; 0867
                      0000G    CF             00    FB 00041           CALLS   #0, CONV$$PARSE_DEF                          ; 0869
                               52             50    D0 00046           MOVL    R0, STATUS                                   ;
                               03             11 00049                 BRB     5$                                           ;
                               52             01    D0 0004B  4$:      MOVL    #1, STATUS                                   ; 0871
                               3D             52    E9 0004E  5$:      BLBC    STATUS, 7$                                   ; 0875
                      0000G    CF             00    FB 00051           CALLS   #0, CONV$$OPEN_INPUT                         ; 0880
                               52             50    D0 00056           MOVL    R0, STATUS                                   ;
                               32             52    E9 00059           BLBC    STATUS, 7$                                   ;
                      0000G    CF             00    FB 0005C           CALLS   #0, CONV$$OPEN_OUTPUT                        ; 0886
                               52             50    D0 00061  6$:      MOVL    R0, STATUS                                   ;
                               27             52    E9 00064           BLBC    STATUS, 7$                                   ; 0890
                      0000G    CF             00    FB 00067           CALLS   #0, CONV$$CREATE_BUFFER                      ; 0896
                               52             50    D0 0006C           MOVL    R0, STATUS                                   ;
                               1C             52    E9 0006F           BLBC    STATUS, 7$                                   ;
                      0000G    CF             00    FB 00072           CALLS   #0, CONV$$CONVERT                            ; 0902
                               52             50    D0 00077           MOVL    R0, STATUS                                   ;
                               11             52    E9 0007A           BLBC    STATUS, 7$                                   ;
                                              64    96 0007D           INCB    CONV$GB_CURRENT_FILE                         ; 0906
     58   A3          64                      08    00    ED 0007F     CMPZV   #0, #8, CONV$GB_CURRENT_FILE, -              ; 0908
                                                                               CONV$GL_FILE_COUNT                           ;
                                              07    1E 00085           BGEQU   7$                                           ;
                      0000G    CF             00    FB 00087           CALLS   #0, CONV$$OPEN_INPUT                         ; 0912
                                              D3    11 0008C           BRB     6$                                           ;
                      0000V    CF             00    FB 0008E  7$:      CALLS   #0, RUNDOWN                                  ; 0920
                                              6C    95 00093           TSTB    (AP)                                         ; 0924
                                              25    13 00095           BEQL    11$                                          ;
                                        04    AC    D5 00097           TSTL    4(AP)                                        ;
                                              20    13 0009A           BEQL    11$                                          ;
                               51       04    AC    D0 0009C           MOVL    4(AP), USER_BLOCK                            ; 0930
                               04             61    D1 000A0           CMPL    (USER_BLOCK), #4                             ; 0934
                                              09    1B 000A3           BLEQU   8$                                           ;
```

```
                                52 00000000G   8F   D0 000A5          MOVL    #CONV$_BADBLK, STATUS                   ; 0936
                                               0E   11 000AC          BRB     11$                                     ;
                                               50   D4 000AE  8$:     CLRL    I                                       ; 0941
                                               06   11 000B0          BRB     10$                                     ;
                              6140        54 A340   D0 000B2  9$:     MOVL    COUNT_BLOCK[I], (USER_BLOCK)[I]         ; 0943
                        F6     50               61   F3 000B8  10$:    AOBLEQ  (USER_BLOCK), I, 9$                     ;
                                               63   94 000BC  11$:    CLRB    SEQUENCE                                ; 0947
                               50              52   D0 000BE          MOVL    STATUS, R0                              ; 0949
                                               04   000C1            RET                                             ; 0951

; Routine Size:  194 bytes,    Routine Base:  _CONV$CODE + 03AA
```

```
;  958     0952  1 %SBTTL  'ADD_KEY'
;  959     0953  1 GLOBAL ROUTINE  CONV$ADD_KEY =
;  960     0954  1 !++
;  961     0955  1 !
;  962     0956  1 ! Functional Description:
;  963     0957  1 !
;  964     0958  1 !       CONVERT/ADD_KEY call interface routine
;  965     0959  1 !
;  966     0960  1 ! Calling Sequence:
;  967     0961  1 !
;  968     0962  1 !       CONV$ADD_KEY( file_name_desc,fdl_file_desc,key[,stat_blk][,flags] )
;  969     0963  1 !
;  970     0964  1 ! Input Parameters:
;  971     0965  1 !
;  972     0966  1 !       file_name_desc  - Address of a string descriptor to be used as
;  973     0967  1 !                         the input file name
;  974     0968  1 !
;  975     0969  1 !       fdl_file_desc   - Address of a string descriptor to be used as
;  976     0970  1 !                         the fdl file name
;  977     0971  1 !
;  978     0972  1 !       key             - Key of reference to add
;  979     0973  1 !
;  980     0974  1 !       stat_blk        - ( Optional ) Address of a block of longwords
;  981     0975  1 !                         which will receive the statistics
;  982     0976  1 !
;  983     0977  1 !       flags           - ( Optional ) Flags longword
;  984     0978  1 !
;  985     0979  1 ! Implicit Inputs:
;  986     0980  1 !       none
;  987     0981  1 !
;  988     0982  1 ! Output Parameters:
;  989     0983  1 !
;  990     0984  1 !       stat_blk
;  991     0985  1 !
;  992     0986  1 ! Implicit Outputs:
;  993     0987  1 !       none
;  994     0988  1 !
;  995     0989  1 ! Routine Value:
;  996     0990  1 !       none
;  997     0991  1 !
;  998     0992  1 ! Routines Called:
;  999     0993  1 !
; 1000     0994  1 !
; 1001     0995  1 ! Side Effects:
; 1002     0996  1 !       none
; 1003     0997  1 !
; 1004     0998  1 !--
; 1005     0999  1
; 1006     1000  2     BEGIN
; 1007     1001  2
; 1008     1002  2     BUILTIN
; 1009     1003  2         ACTUALCOUNT,
; 1010     1004  2         ACTUALPARAMETER,
; 1011     1005  2         NULLPARAMETER;
; 1012     1006  2
; 1013     1007  2     DEFINE_KEY_DESC_GLOBAL;
; 1014     1008  2
```

```
;  1015          1009  2     LOCAL
;  1016          1010  2         BYTES,
;  1017          1011  2         VM_POINTER,
;  1018          1012  2         STATUS;
;  1019          1013  2
;  1020          1014  2     ! Set up the exit handler
;  1021          1015  2     !
;  1022          1016  2     LIB$ESTABLISH( CONDITION_HANDLER );
;  1023          1017  2
;  1024          1018  2     ! Check to make sure convert is not being called
;  1025          1019  2     !
;  1026          1020  2     IF .SEQUENCE NEQU 0
;  1027          1021  2     THEN
;  1028          1022  2         RETURN CONV$_ORDER;
;  1029          1023  2
;  1030          1024  2     ! This call needs at least three arguments and no more then 5
;  1031          1025  2     !
;  1032          1026  2     IF ( ACTUALCOUNT() LSSU 3 ) OR ( ACTUALCOUNT() GTRU 5 )
;  1033          1027  2     THEN
;  1034          1028  2         RETURN CONV$_NARG;
;  1035          1029  2
;  1036          1030  2     ! Clear the flags
;  1037          1031  2     !
;  1038          1032  2     CONV$AB_FLAGS = _CLEAR;
;  1039          1033  2
;  1040          1034  2     ! If the user specified a flags parameter stuff it
;  1041          1035  2     !
;  1042          1036  2     IF ACTUALCOUNT() EQLU 5
;  1043          1037  2     THEN
;  1044          1038  2         CONV$AB_FLAGS [ CONV$W_USER ] = .ACTUALPARAMETER(5);
;  1045          1039  2
;  1046          1040  2     ! Allocate memory for all of the name block buffers
;  1047          1041  2     !
;  1048          1042  2     BYTES = ESA_BUF_SIZ + RSA_BUF_SIZ;
;  1049          1043  2
;  1050          1044  2     VM_POINTER = CONV$$GET_VM ( .BYTES );
;  1051          1045  2
;  1052          1046  2     ! Init the output RMS blocks
;  1053          1047  2     !
;  1054          1048  2     ! The FAB
;  1055          1049  2     !
;  1056        P 1050  2     $FAB_INIT ( FAB = CONV$AB_OUT_FAB,
;  1057        P 1051  2                 FAC = <BRO,GET,PUT>,
;  1058        P 1052  2                 FOP = <DFW,NAM,OFP,SQO>,
;  1059        P 1053  2                 NAM = CONV$AB_OUT_NAM,
;  1060          1054  2                 XAB = CONV$AB_OUT_XABSUM );
;  1061          1055  2
;  1062          1056  2     ! The RAB
;  1063          1057  2     !
;  1064        P 1058  2     $RAB_INIT ( RAB = CONV$AB_OUT_RAB,
;  1065        P 1059  2                 FAB = CONV$AB_OUT_FAB,
;  1066          1060  2                 ROP = <BIO,WBH> );
;  1067          1061  2
;  1068          1062  2     ! The name block
;  1069          1063  2     !
;  1070        P 1064  2     $NAM_INIT ( NAM = CONV$AB_OUT_NAM,
;  1071        P 1065  2                 ESA = .VM_POINTER,
```

```
;  1072        P 1066  2                   ESS = ESA_BUF_SIZ,
;  1073        P 1067  2                   RSA = .VM_POINTER + ESA_BUF_SIZ,
;  1074          1068  2                   RSS = RSA_BUF_SIZ );
;  1075          1069  2
;  1076          1070  2           ! The 1st argument is the output file name
;  1077          1071  2           !
;  1078          1072  2           IF NULLPARAMETER(1)
;  1079          1073  2           THEN
;  1080          1074  2               RETURN CONV$_ILL_VALUE
;  1081          1075  2           ELSE
;  1082          1076  2               CONV$AR_OUT_FILE_NAM = COPY_DESC( ACTUALPARAMETER( 1 ) );
;  1083          1077  2
;  1084          1078  2           ! The 2nd argument is the fdl file descriptor
;  1085          1079  2           !
;  1086          1080  2           IF NULLPARAMETER(2)
;  1087          1081  2           THEN
;  1088          1082  2               RETURN CONV$_ILL_VALUE
;  1089          1083  2           ELSE
;  1090          1084  2               CONV$AR_FDL_FILE_NAM = COPY_DESC( ACTUALPARAMETER( 2 ) );
;  1091          1085  2
;  1092          1086  2           ! The 3rd argument is the key of ref. to add
;  1093          1087  2           !
;  1094          1088  2           IF NULLPARAMETER(3)
;  1095          1089  2           THEN
;  1096          1090  2               RETURN CONV$_ILL_VALUE
;  1097          1091  2           ELSE
;  1098          1092  2               CONV$GL_ADD_DELE_KEY = .ACTUALPARAMETER( 3 );
;  1099          1093  2
;  1100          1094  2           ! Open the output file
;  1101          1095  2           !
;  1102          1096  2           IF STATUS = ADD$$OPEN_OUTPUT()
;  1103          1097  2           THEN
;  1104          1098  2
;  1105          1099  2               ! Check the input key and fdl key and make index desc for it
;  1106          1100  2               !
;  1107          1101  2               IF STATUS = ADD$$CHECK_KEY()
;  1108          1102  2               THEN
;  1109          1103  2
;  1110          1104  2                   ! Load the key into the file
;  1111          1105  2                   !
;  1112          1106  2                   STATUS = ADD$$LOAD_KEY();
;  1113          1107  2
;  1114          1108  2           ! Close the file and deallocate memory
;  1115          1109  2           !
;  1116          1110  2           RUNDOWN();
;  1117          1111  2
;  1118          1112  2           ! If there was a 4th parameter stuff it with the statistics block addr.
;  1119          1113  2           !
;  1120          1114  2           IF NOT NULLPARAMETER(4)
;  1121          1115  2           THEN
;  1122          1116  2               BEGIN
;  1123          1117  2
;  1124          1118  2               LOCAL
;  1125          1119  2                   USER_BLOCK : REF VECTOR [ ,LONG ];
;  1126          1120  2
;  1127          1121  3               ! Get the user block
;  1128          1122  3               !
```

```
;  1129     1123  3          USER_BLOCK = ACTUALPARAMETER(4);
;  1130     1124  3
;  1131     1125  3          ! Check to see if there really is one
;  1132     1126  3          !
;  1133     1127  3          IF .USER_BLOCK NEQ 0
;  1134     1128  3          THEN
;  1135     1129  4              BEGIN
;  1136     1130  4
;  1137     1131  4              INCR I FROM 1 TO .USER_BLOCK [ 0 ] BY 1
;  1138     1132  4              DO
;  1139     1133  4                  USER_BLOCK [ .I ] = .STATISTICS_BLOCK [ .I ]
;  1140     1134  4
;  1141     1135  4              END
;  1142     1136  4
;  1143     1137  2          END;
;  1144     1138  2
;  1145     1139  2      RETURN .STATUS
;  1146     1140  2
;  1147     1141  1      END;


                                        0FFC 00000          .ENTRY  CONV$ADD_KEY, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0953
                                                                    R10,R11
                                 58      0000G  CF  9E 00002        MOVAB   $RMS_PTR, R8                                 ;
                                 57      0000G  CF  9E 00007        MOVAB   CONV$AB_OUT_NAM, R7                          ;
                                 56      0000G  CF  9E 0000C        MOVAB   $RMS_PTR, R6                                 ;
                                         0000V  CF  9F 00011        PUSHAB  CONDITION_HANDLER                            ; 1016
                           00000000G     00     01  FB 00015        CALLS   #1, LIB$ESTABLISH                            ;
                                         0000'  CF  95 0001C        TSTB    SEQUENCE                                     ; 1020
                                                08  13 00020        BEQL    1$                                           ;
                                 50 00000000G   8F  D0 00022        MOVL    #CONV$_ORDER, R0                             ; 1022
                                                04  00029        RET                                                  ;
                                 03                 6C  91 0002A 1$:    CMPB    (AP), #3                             ; 1026
                                                    05  1F 0002D        BLSSU   2$                                   ;
                                 05                 6C  91 0002F        CMPB    (AP), #5                             ;
                                                    08  1B 00032        BLEQU   3$                                   ;
                                 50 00000000G       8F  D0 00034 2$:    MOVL    #CONV$_NARG, R0                      ; 1028
                                                    04  0003B        RET                                          ;
                                         0000G      CF  D4 0003C 3$:    CLRL    CONV$AB_FLAGS                        ; 1032
                                 05                 6C  91 00040        CMPB    (AP), #5                             ; 1036
                                                    06  12 00043        BNEQ    4$                                   ;
                           0000G CF          14     BC  B0 00045        MOVW    @20(AP), CONV$AB_FLAGS               ; 1038
                                 50          A0     8F  9A 0004B 4$:    MOVZBL  #160, BYTES                          ; 1042
                                                    50  DD 0004F        PUSHL   BYTES                                ; 1044
                                                0000G   30  00051        BSBW    CONV$$GET_VM                         ;
                                 5E                     04  C0 00054        ADDL2   #4, SP                               ;
                                 5B                     50  D0 00057        MOVL    R0, VM_POINTER                       ;
0050  8F                 00      6E                     00  2C 0005A        MOVC5   #0, (SP), #0, #80, $RMS_PTR          ; 1054
                                                        66  00061        ;                                            ;
                                 66          5003   8F  B0 00062        MOVW    #20483, $RMS_PTR                     ;
                           04    A6  21000060       8F  D0 00067        MOVL    #553648224, $RMS_PTR+4               ;
                           16    A6          43     8F  90 0006F        MOVB    #67, $RMS_PTR+22                     ;
                           1F    A6                 02  90 00074        MOVB    #2, $RMS_PTR+31                      ;
                           24    A6      0000G      CF  9E 00078        MOVAB   CONV$AB_OUT_XABSUM, $RMS_PTR+36       ;
```

```
                           28      A6              67  9E  0007E        MOVAB   CONV$AB_OUT_NAM, $RMS_PTR+40      ;
   0044   8F         00            6E              00  2C  00082        MOVC5   #0, (SP), #0, #68, $RMS_PTR       ; 1060
                                                   68      00089
                                   68      4401    8F  B0  0008A        MOVW    #17409, $RMS_PTR                  ;
                           04      A8      0C00    8F  3C  0008F        MOVZWL  #3072, $RMS_PTR+4                 ;
                           3C      A8              66  9E  00095        MOVAB   CONV$AB_OUT_FAB, $RMS_PTR+60      ;
   0060   8F         00            6E              00  2C  00099        MOVC5   #0, (SP), #0, #96, $RMS_PTR       ; 1068
                                                   67      000A0
                                   67      6002    8F  B0  000A1        MOVW    #24578, $RMS_PTR                  ;
                           02      A7        50    8F  90  000A6        MOVB    #80, $RMS_PTR+2                   ;
                           04      A7        50    AB  9E  000AB        MOVAB   80(R11), $RMS_PTR+4               ;
                           0A      A7        50    8F  90  000B0        MOVB    #80, $RMS_PTR+10                  ;
                           0C      A7              5B  D0  000B5        MOVL    VM_POINTER, $RMS_PTR+12           ;
                                                   6C  95  000B9        TSTB    (AP)                              ; 1072
                                                   31  13  000BB        BEQL    5$                                ;
                                             04    AC  D5  000BD        TSTL    4(AP)                             ;
                                                   2C  13  000C0        BEQL    5$                                ;
                                   50        04    AC  D0  000C2        MOVL    4(AP), R0                         ; 1076
                                              0000V    30  000C6        BSBW    COPY_DESC                         ;
                         0000G     CF              50  D0  000C9        MOVL    R0, CONV$AR_OUT_FILE_NAM          ;
                                   02              6C  91  000CE        CMPB    (AP), #2                          ; 1080
                                                   1B  1F  000D1        BLSSU   5$                                ;
                                             08    AC  D5  000D3        TSTL    8(AP)                             ;
                                                   16  13  000D6        BEQL    5$                                ;
                                   50        08    AC  D0  000D8        MOVL    8(AP), R0                         ; 1084
                                              0000V    30  000DC        BSBW    COPY_DESC                         ;
                         0000G     CF              50  D0  000DF        MOVL    R0, CONV$AR_FDL_FILE_NAM          ;
                                   03              6C  91  000E4        CMPB    (AP), #3                          ; 1088
                                                   05  1F  000E7        BLSSU   5$                                ;
                                             0C    AC  D5  000E9        TSTL    12(AP)                            ;
                                                   08  12  000EC        BNEQ    6$                                ;
                                   50 00000000G    8F  D0  000EE  5$:   MOVL    #CONV$_ILL_VALUE, R0              ; 1090
                                                       04  000F5        RET                                       ;
                         0000G     CF        0C    BC  D0  000F6  6$:   MOVL    @12(AP), CONV$GL_ADD_DELE_KEY     ; 1092
                         0000G     CF              00  FB  000FC        CALLS   #0, ADD$$OPEN_OUTPUT              ; 1096
                                   52              50  D0  00101        MOVL    R0, STATUS                        ;
                                   0F              52  E9  00104        BLBC    STATUS, 7$                        ;
                                              0000G    30  00107        BSBW    ADD$$CHECK_KEY                    ; 1101
                                   52              50  D0  0010A        MOVL    R0, STATUS                        ;
                                   06              52  E9  0010D        BLBC    STATUS, 7$                        ;
                                              0000G    30  00110        BSBW    ADD$$LOAD_KEY                     ; 1106
                                   52              50  D0  00113        MOVL    R0, STATUS                        ;
                         0000V     CF              00  FB  00116  7$:   CALLS   #0, RUNDOWN                       ; 1110
                                   04              6C  91  0011B        CMPB    (AP), #4                          ; 1114
                                                   1A  1F  0011E        BLSSU   10$                               ;
                                             10    AC  D5  00120        TSTL    16(AP)                            ;
                                                   15  13  00123        BEQL    10$                               ;
                                   51        10    AC  D0  00125        MOVL    16(AP), USER_BLOCK                ; 1123
                                                   0F  13  00129        BEQL    10$                               ; 1127
                                                   50  D4  0012B        CLRL    I                                 ; 1131
                                                   07  11  0012D        BRB     9$                                ;
                              6140      0000'CF40      D0  0012F  8$:   MOVL    STATISTICS_BLOCK[I], (USER_BLOCK)[I]  ; 1133
                     F5            50              61  F3  00136  9$:   AOBLEQ  (USER_BLOCK), I, 8$               ;
                                   50              52  D0  0013A  10$:  MOVL    STATUS, R0                        ; 1139
                                                       04  0013D        RET                                       ; 1141

; Routine Size:  318 bytes,    Routine Base:  _CONV$CODE + 046C
```

```
;  1149          1142  1 %SBTTL  'RECLAIM'
;  1150          1143  1 GLOBAL ROUTINE  CONV$RECLAIM =
;  1151          1144  1 !++
;  1152          1145  1 !
;  1153          1146  1 ! Functional Description:
;  1154          1147  1 !
;  1155          1148  1 !       CONVERT/RECLAIM call interface routine
;  1156          1149  1 !
;  1157          1150  1 ! Calling Sequence:
;  1158          1151  1 !
;  1159          1152  1 !       CONV$RECLAIM( file_name_desc[,stat_blk][,flags] )
;  1160          1153  1 !
;  1161          1154  1 ! Input Parameters:
;  1162          1155  1 !
;  1163          1156  1 !       file_name_desc  - Address of a string descriptor to be used as
;  1164          1157  1 !                         the input file name
;  1165          1158  1 !
;  1166          1159  1 !       stat_blk        - ( Optional ) Address of a block of longwords
;  1167          1160  1 !                         which will receive the statistics
;  1168          1161  1 !
;  1169          1162  1 !       flags           - ( Optional ) Flags longword
;  1170          1163  1 !
;  1171          1164  1 ! Implicit Inputs:
;  1172          1165  1 !       none
;  1173          1166  1 !
;  1174          1167  1 ! Output Parameters:
;  1175          1168  1 !
;  1176          1169  1 !       stat_blk
;  1177          1170  1 !
;  1178          1171  1 ! Implicit Outputs:
;  1179          1172  1 !       none
;  1180          1173  1 !
;  1181          1174  1 ! Routine Value:
;  1182          1175  1 !       none
;  1183          1176  1 !
;  1184          1177  1 ! Routines Called:
;  1185          1178  1 !
;  1186          1179  1 !       RECL$$OPEN_FILE
;  1187          1180  1 !       CONV$$SET_KEY_DESC
;  1188          1181  1 !       CONV$$GET_NEXT_KEY
;  1189          1182  1 !       RECL$$ALLOCATE_BUFFERS
;  1190          1183  1 !       RECL$$SCAN_DATA_LEVEL
;  1191          1184  1 !       RUNDOWN
;  1192          1185  1 !
;  1193          1186  1 ! Side Effects:
;  1194          1187  1 !       none
;  1195          1188  1 !
;  1196          1189  1 !--
;  1197          1190  1
;  1198          1191  2     BEGIN
;  1199          1192  2
;  1200          1193  2     BUILTIN
;  1201          1194  2         ACTUALCOUNT,
;  1202          1195  2         ACTUALPARAMETER,
;  1203          1196  2         NULLPARAMETER;
;  1204          1197  2
;  1205          1198  2     DEFINE_CTX_GLOBAL;
```

```
;  1206     1199  2     DEFINE_BUCKET_GLOBAL;
;  1207     1200  2     DEFINE_KEY_DESC_GLOBAL;
;  1208     1201  2
;  1209     1202  2     LOCAL
;  1210     1203  2         STATUS;
;  1211     1204  2
;  1212     1205  2     ! Set up the exit handler
;  1213     1206  2     !
;  1214     1207  2     LIB$ESTABLISH ( CONDITION_HANDLER );
;  1215     1208  2
;  1216     1209  2     ! Check to make sure convert is not being called
;  1217     1210  2     !
;  1218     1211  2     IF .SEQUENCE NEQU 0
;  1219     1212  2     THEN
;  1220     1213  2         RETURN CONV$_ORDER;
;  1221     1214  2
;  1222     1215  2     ! Check on the number of arguments
;  1223     1216  2     !
;  1224     1217  3     IF ( ACTUALCOUNT() LSSU 1 ) OR ( ACTUALCOUNT() GTRU 3 )
;  1225     1218  2     THEN
;  1226     1219  2         RETURN CONV$_NARG;
;  1227     1220  2
;  1228     1221  2     ! Clear the flags and counters
;  1229     1222  2     !
;  1230     1223  2     CONV$AB_FLAGS        = _CLEAR;
;  1231     1224  2     RECL$GL_BUCKET_COUNT= _CLEAR;
;  1232     1225  2     RECL$GL_DATA_COUNT   = _CLEAR;
;  1233     1226  2     RECL$GL_INDEX_COUNT = _CLEAR;
;  1234     1227  2
;  1235     1228  2     ! Was user flags specified
;  1236     1229  2     !
;  1237     1230  2     IF ACTUALCOUNT() EQLU 3
;  1238     1231  2     THEN
;  1239     1232  2         CONV$AB_FLAGS [ CONV$W_USER ] = .ACTUALPARAMETER(3);
;  1240     1233  2
;  1241     1234  2     ! Open the file
;  1242     1235  2     !
;  1243     1236  2     IF NULLPARAMETER(1)
;  1244     1237  2     THEN
;  1245     1238  2         RETURN CONV$_ILL_VALUE
;  1246     1239  2     ELSE
;  1247     1240  2         RET_ON_ERROR( RECL$$OPEN_FILE( ACTUALPARAMETER(1) ) );
;  1248     1241  2
;  1249     1242  2     ! Get the first key
;  1250     1243  2     !
;  1251     1244  3     IF ( STATUS = CONV$$SET_KEY_DESC( 0 ) )
;  1252     1245  2     THEN
;  1253     1246  2
;  1254     1247  2         ! Process the keys
;  1255     1248  2         !
;  1256     1249  2         DO
;  1257     1250  3             BEGIN
;  1258     1251  3
;  1259     1252  3             ! If the index is not initialized don't try to do anything
;  1260     1253  3             ! with it
;  1261     1254  3             !
;  1262     1255  3             IF NOT .KEY_DESC [ KEY$V_INITIDX ]
```

```
;  1263     1256  3            THEN
;  1264     1257  4                BEGIN
;  1265     1258  4
;  1266     1259  4                ! Allocate bucket buffers and the context block
;  1267     1260  4                !
;  1268     1261  5                IF NOT ( STATUS = RECL$$ALLOCATE_BUFFERS() )
;  1269     1262  4                THEN
;  1270     1263  4                    EXITLOOP;
;  1271     1264  4
;  1272     1265  4                ! Scan the data level buckets and remove the empties
;  1273     1266  4                !
;  1274     1267  5                IF NOT ( STATUS = RECL$$SCAN_DATA_LEVEL() )
;  1275     1268  4                THEN
;  1276     1269  4                    EXITLOOP;
;  1277     1270  4
;  1278     1271  4                ! Deallocate memory used foe the bucket buffers
;  1279     1272  4                !
;  1280     1273  4                CONV$$FREE_TEMP_VM()
;  1281     1274  4
;  1282     1275  4                END
;  1283     1276  3            END
;  1284     1277  2        UNTIL NOT CONV$$GET_NEXT_KEY();
;  1285     1278  2
;  1286     1279  2    ! Close the file and deallocate memory
;  1287     1280  2    !
;  1288     1281  2    RUNDOWN();
;  1289     1282  2
;  1290     1283  2    ! If there was a second parameter stuff it with the statistics block addr.
;  1291     1284  2    !
;  1292     1285  2    IF NOT NULLPARAMETER(2)
;  1293     1286  2    THEN
;  1294     1287  3        BEGIN
;  1295     1288  3
;  1296     1289  3        LOCAL
;  1297     1290  3            USER_BLOCK : REF VECTOR [ ,LONG ];
;  1298     1291  3
;  1299     1292  3        ! Get the user block
;  1300     1293  3        !
;  1301     1294  3        USER_BLOCK = ACTUALPARAMETER(2);
;  1302     1295  3
;  1303     1296  3        ! Check to see if there really is one
;  1304     1297  3        !
;  1305     1298  3        IF .USER_BLOCK NEQ 0
;  1306     1299  3        THEN
;  1307     1300  4            BEGIN
;  1308     1301  4
;  1309     1302  4            ! Stuff the total bucket count
;  1310     1303  4            !
;  1311     1304  4            STATISTICS_BLOCK [ 4 ] = .RECL$GL_DATA_COUNT + .RECL$GL_INDEX_COUNT;
;  1312     1305  4
;  1313     1306  4            INCR I FROM 1 TO .USER_BLOCK [ 0 ] BY 1
;  1314     1307  4            DO
;  1315     1308  4                USER_BLOCK [ .I ] = .STATISTICS_BLOCK [ .I ]
;  1316     1309  4
;  1317     1310  4            END
;  1318     1311  4
;  1319     1312  2        END;
```

```
; 1320      1313  2
; 1321      1314  2        RETURN .STATUS
; 1322      1315  2
; 1323      1316  1        END;



                                         OFFC 00000          .ENTRY   CONV$RECLAIM, Save R2,R3,R4,R5,R6,R7,R8,R9,-;  1143
                                                                      R10,R11                                     ;
                    53       0000'   CF    9E 00002          MOVAB    RECL$GL_INDEX_COUNT, R3                     ;
                             0000V   CF    9F 00007          PUSHAB   CONDITION_HANDLER                           ;  1207
          00000000G 00               01    FB 0000B          CALLS    #1, LIB$ESTABLISH                           ;
                                8C   A3    95 00012          TSTB     SEQUENCE                                    ;  1211
                                     08    13 00015          BEQL     1$                                          ;
                    50 00000000G     8F    D0 00017          MOVL     #CONV$_ORDER, R0                            ;  1213
                                           04 0001E          RET                                                  ;
                                     6C    95 0001F  1$:     TSTB     (AP)                                        ;  1217
                                     05    13 00021          BEQL     2$                                          ;
                    03               6C    91 00023          CMPB     (AP), #3                                    ;
                                     08    1B 00026          BLEQU    3$                                          ;
                    50 00000000G     8F    D0 00028  2$:     MOVL     #CONV$_NARG, R0                             ;  1219
                                           04 0002F          RET                                                  ;
                             0000G   CF    D4 00030  3$:     CLRL     CONV$AB_FLAGS                               ;  1223
                                F8   A3    7C 00034          CLRQ     RECL$GL_BUCKET_COUNT                        ;  1224
                                     63    D4 00037          CLRL     RECL$GL_INDEX_COUNT                         ;  1226
                    03               6C    91 00039          CMPB     (AP), #3                                    ;  1230
                                     06    12 0003C          BNEQ     4$                                          ;
              0000G CF          0C   BC    B0 0003E          MOVW     @12(AP), CONV$AB_FLAGS                      ;  1232
                                     6C    95 00044  4$:     TSTB     (AP)                                        ;  1236
                                     05    13 00046          BEQL     5$                                          ;
                                04   AC    D5 00048          TSTL     4(AP)                                       ;
                                     08    12 0004B          BNEQ     6$                                          ;
                    50 00000000G     8F    D0 0004D  5$:     MOVL     #CONV$_ILL_VALUE, R0                        ;  1238
                                           04 00054          RET                                                  ;
                                04   AC    DD 00055  6$:     PUSHL    4(AP)                                       ;  1240
              0000G CF               01    FB 00058          CALLS    #1, RECL$$OPEN_FILE                         ;
                    5A               50    E9 0005D          BLBC     STATUS, 13$                                 ;
                                     7E    D4 00060          CLRL     -(SP)                                       ;  1244
                                  0000G    30 00062          BSBW     CONV$$SET_KEY_DESC                          ;
                    5E               04    C0 00065          ADDL2    #4, SP                                      ;
                    52               50    D0 00068          MOVL     R0, STATUS                                  ;
                    20               52    E9 0006B          BLBC     STATUS, 9$                                  ;
          15        10  AB           04    E0 0006E  7$:     BBS      #4, 16(KEY_DESC), 8$                        ;  1255
                                  0000G    30 00073          BSBW     RECL$$ALLOCATE_BUFFERS                      ;  1261
                    52               50    D0 00076          MOVL     R0, STATUS                                  ;
                    12               52    E9 00079          BLBC     STATUS, 9$                                  ;
                                  0000G    30 0007C          BSBW     RECL$$SCAN_DATA_LEVEL                       ;  1267
                    52               50    D0 0007F          MOVL     R0, STATUS                                  ;
                    09               52    E9 00082          BLBC     STATUS, 9$                                  ;
                                  0000G    30 00085          BSBW     CONV$$FREE_TEMP_VM                          ;  1273
                                  0000G    30 00088  8$:     BSBW     CONV$$GET_NEXT_KEY                          ;  1277
                    E0               50    E8 0008B          BLBS     R0, 7$                                      ;
              0000V CF               00    FB 0008E  9$:     CALLS    #0, RUNDOWN                                 ;  1281
                    02               6C    91 00093          CMPB     (AP), #2                                    ;  1285
                    1F               1F    1F 00096          BLSSU    12$                                         ;
```

```
                              08   AC  D5 00098          TSTL    8(AP)                                          ;
                                   1A  13 0009B          BEQL    12$                                            ;
                    51        08   AC  D0 0009D          MOVL    8(AP), USER_BLOCK                              ; 1294
                                   14  13 000A1          BEQL    12$                                            ; 1298
          04   A3       FC   A3    63  C1 000A3          ADDL3   RECL$GL_INDEX_COUNT, RECL$GL_DATA_COUNT, -     ; 1304
                                                                 STATISTICS_BLOCK+16
                                   50  D4 000A9          CLRL    I                                              ; 1306
                                   06  11 000AB          BRB     11$                                            ;
                  6140       F4  A340  D0 000AD 10$:     MOVL    STATISTICS_BLOCK[I], (USER_BLOCK)[I]           ; 1308
               F6   50             61  F3 000B3 11$:     AOBLEQ  (USER_BLOCK), I, 10$                           ;
                    50             52  D0 000B7 12$:     MOVL    STATUS, R0                                     ; 1314
                                       04 000BA 13$:     RET                                                    ; 1316

; Routine Size:  187 bytes,    Routine Base:  _CONV$CODE + 05AA
```

```
;  1325     1317  1 %SBTTL  'RUNDOWN'
;  1326     1318  1 ROUTINE RUNDOWN : NOVALUE =
;  1327     1319  1 !++
;  1328     1320  1 !
;  1329     1321  1 ! Functional Description:
;  1330     1322  1 !
;  1331     1323  1 !       Close all open files and deallocate memory
;  1332     1324  1 !
;  1333     1325  1 ! Calling Sequence:
;  1334     1326  1 !
;  1335     1327  1 !       RUNDOWN()
;  1336     1328  1 !
;  1337     1329  1 ! Input Parameters:
;  1338     1330  1 !       none
;  1339     1331  1 !
;  1340     1332  1 ! Implicit Inputs:
;  1341     1333  1 !       none
;  1342     1334  1 !
;  1343     1335  1 ! Output Parameters:
;  1344     1336  1 !       none
;  1345     1337  1 !
;  1346     1338  1 ! Implicit Outputs:
;  1347     1339  1 !       none
;  1348     1340  1 !
;  1349     1341  1 ! Routine Value:
;  1350     1342  1 !       none
;  1351     1343  1 !
;  1352     1344  1 ! Routines Called:
;  1353     1345  1 !
;  1354     1346  1 !       $DISCONNECT
;  1355     1347  1 !       $CLOSE
;  1356     1348  1 !       CONV$$FREE_TEMP_VM
;  1357     1349  1 !       CONV$$FREE_VM
;  1358     1350  1 !
;  1359     1351  1 ! Side Effects:
;  1360     1352  1 !
;  1361     1353  1 !       Closes all files and deallocates memory
;  1362     1354  1 !
;  1363     1355  1 !--
;  1364     1356  1
;  1365     1357  2     BEGIN
;  1366     1358  2
;  1367     1359  2     ! Clear the sequencing so we can start over again
;  1368     1360  2     !
;  1369     1361  2     SEQUENCE = 0;
;  1370     1362  2
;  1371     1363  2     ! Free any miscellaneous memory held by LIB$FIND_FILE
;  1372     1364  2     !
;  1373     1365  2     LIB$FIND_FILE_END(CONV$GL_FINDFILE_CTX);
;  1374     1366  2
;  1375     1367  2     ! Close any open files
;  1376     1368  2     !
;  1377     1369  2     ! Start with input file and RFA file
;  1378     1370  2     !
;  1379     1371  2     CONV$$END_OF_FILE();
;  1380     1372  2
;  1381     1373  2     ! Output File
```

```
; 1382        1374  2      !
; 1383        1375  2      IF .CONV$AB_FLAGS [ CONV$V_OUT ]
; 1384        1376  2      THEN
; 1385        1377  3          BEGIN
; 1386        1378  3          !
; 1387        1379  3          ! If we're doing a FTN --> STM conversion, then we
; 1388        1380  3          !       need to flush our STM buffer now.
; 1389        1381  3          !
; 1390        1382  3          IF .CONV$AB_FLAGS [ CONV$V_MAPFTN ] EQL CONV$C_FTNSTM
; 1391        1383  3          THEN
; 1392        1384  4              BEGIN
; 1393        1385  4              CONV$GW_OUT_REC_SIZ = .CONV$GL_STM_REC_LEN;
; 1394        1386  4              CONV$AB_OUT_RAB [RAB$L_RBF] = .CONV$GL_STM_BUF;
; 1395        1387  4              CONV$$PUT_RECORD ();
; 1396        1388  3              END;
; 1397        1389  3          $DISCONNECT( RAB=CONV$AB_OUT_RAB );
; 1398        1390  3          $CLOSE( FAB=CONV$AB_OUT_FAB );
; 1399        1391  2          END;
; 1400        1392  2
; 1401        1393  2      ! If an Exception file was used close it
; 1402        1394  2      !
; 1403        1395  2      IF .CONV$AB_FLAGS [ CONV$V_EXC ]
; 1404        1396  2      THEN
; 1405        1397  3          BEGIN
; 1406        1398  3          $DISCONNECT( RAB=CONV$AB_EXC_RAB );
; 1407        1399  4          $CLOSE( FAB=EXC_FAB )
; 1408        1400  2          END;
; 1409        1401  2
; 1410        1402  2      ! Deallocate any loose memory floating around
; 1411        1403  2      !
; 1412        1404  2      CONV$$FREE_TEMP_VM();
; 1413        1405  2      CONV$$FREE_VM();
; 1414        1406  2
; 1415        1407  2      RETURN
; 1416        1408  2
; 1417        1409  1      END;


                                                         .EXTRN  SYS$DISCONNECT, SYS$CLOSE

                                   0FFC 00000 RUNDOWN:.WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11               ; 1318
                  54      0000G  CF   9E 00002         MOVAB   CONV$AB_FLAGS+2, R4                                 ;
                  53  00000000G  00   9E 00007         MOVAB   SYS$CLOSE, R3                                       ;
                  52  00000000G  00   9E 0000E         MOVAB   SYS$DISCONNECT, R2                                  ;
                          0000'  CF   94 00015         CLRB    SEQUENCE                                            ; 1361
                          0000G  CF   9F 00019         PUSHAB  CONV$GL_FINDFILE_CTX                                ; 1365
           00000000G  00          01  FB 0001D         CALLS   #1, LIB$FIND_FILE_END                               ;
               0000G  CF          00  FB 00024         CALLS   #0, CONV$$END_OF_FILE                               ; 1371
               2A     64          01  E1 00029         BBC     #1, CONV$AB_FLAGS+2, 2$                             ; 1375
    02         64     02          07  ED 0002D         CMPZV   #7, #2, CONV$AB_FLAGS+2, #2                         ; 1382
                                  15  12 00032         BNEQ    1$                                                  ;
               0000G  CF   0000G  CF  BC 00034         MOVW    CONV$GL_STM_REC_LEN, CONV$GW_OUT_REC_SIZ            ; 1385
               0000G  CF   0000G  CF  D0 0003B         MOVL    CONV$GL_STM_BUF, CONV$AB_OUT_RAB+40                 ; 1386
           00000000G  00          00  FB 00042         CALLS   #0, CONV$$PUT_RECORD                                ; 1387
                          0000G  CF   9F 00049 1$:     PUSHAB  CONV$AB_OUT_RAB                                     ; 1389
                  62          01  FB 0004D             CALLS   #1, SYS$DISCONNECT                                  ;
```

```
                          0000G  CF   9F 00050          PUSHAB  CONV$AB_OUT_FAB                    ; 1390
                      63         01   FB 00054          CALLS   #1, SYS$CLOSE                     ;
                   0E     64         02   E1 00057 2$:      BBC     #2, CONV$AB_FLAGS+2, 3$          ; 1395
                          0000G  CF   9F 0005B          PUSHAB  CONV$AB_EXC_RAB                   ; 1398
                      62         01   FB 0005F          CALLS   #1, SYS$DISCONNECT                ;
                          0000'  CF   9F 00062          PUSHAB  EXC_FAB                           ; 1399
                      63         01   FB 00066          CALLS   #1, SYS$CLOSE                     ;
                               0000G 30  00069 3$:      BSBW    CONV$$FREE_TEMP_VM                ; 1404
                               0000G 30  0006C          BSBW    CONV$$FREE_VM                     ; 1405
                                     04  0006F          RET                                       ; 1409

; Routine Size:  112 bytes,    Routine Base:  _CONV$CODE + 0665
```

```
;  1419     1410  1 %SBTTL  'CONDITION_HANDLER'
;  1420     1411  1 ROUTINE CONDITION_HANDLER ( SIGNAL_VECTOR : REF BLOCK [ ,BYTE ],MECH_VECTOR ) =
;  1421     1412  1 !++
;  1422     1413  1 !
;  1423     1414  1 ! Functional Description:
;  1424     1415  1 !
;  1425     1416  1 !       Exception handler to make sure files are closed for the call interface
;  1426     1417  1 !
;  1427     1418  1 ! Calling Sequence:
;  1428     1419  1 !
;  1429     1420  1 !       Called as exception handler
;  1430     1421  1 !
;  1431     1422  1 ! Input Parameters:
;  1432     1423  1 !
;  1433     1424  1 !       SIGNAL_VECTOR
;  1434     1425  1 !       MECH_VECTOR
;  1435     1426  1 !
;  1436     1427  1 ! Implicit Inputs:
;  1437     1428  1 !       none
;  1438     1429  1 !
;  1439     1430  1 ! Output Parameters:
;  1440     1431  1 !       none
;  1441     1432  1 !
;  1442     1433  1 ! Implicit Outputs:
;  1443     1434  1 !       none
;  1444     1435  1 !
;  1445     1436  1 ! Routine Value:
;  1446     1437  1 !
;  1447     1438  1 !       SS$_RESIGNAL
;  1448     1439  1 !
;  1449     1440  1 ! Routines Called:
;  1450     1441  1 !
;  1451     1442  1 !       RUNDOWN
;  1452     1443  1 !       LIB$SIG_TO_RETURN
;  1453     1444  1 !
;  1454     1445  1 ! Side Effects:
;  1455     1446  1 !
;  1456     1447  1 !       Closes open files and dealloceates memory
;  1457     1448  1 !
;  1458     1449  1 !--
;  1459     1450  1
;  1460     1451  2     BEGIN
;  1461     1452  2
;  1462     1453  2     LOCAL
;  1463     1454  2         CONDITION_CODE : BLOCK [ 4,BYTE ];
;  1464     1455  2
;  1465     1456  2     ! Get the condition code
;  1466     1457  2     !
;  1467     1458  2     CONDITION_CODE = .SIGNAL_VECTOR [ CHF$L_SIG_NAME ];
;  1468     1459  2
;  1469     1460  2     ! If an unwind is in progress simply clean-up and return
;  1470     1461  2     !
;  1471     1462  2     IF .CONDITION_CODE EQLU SS$_UNWIND
;  1472     1463  2     THEN
;  1473     1464  3         BEGIN
;  1474     1465  3         RUNDOWN();
;  1475     1466  3         RETURN SS$_RESIGNAL;
```

```
; 1476          1467  2          END;
; 1477          1468  2
; 1478          1469  2      ! Do we signal errors or just return
; 1479          1470  2      !
; 1480          1471  2      IF .CONV$AB_FLAGS [ CONV$V_SIGNAL ]
; 1481          1472  2      THEN
; 1482          1473  3          BEGIN
; 1483          1474  3          IF .CONDITION_CODE [ STS$V_SEVERITY ] EQL STS$K_SEVERE
; 1484          1475  3          THEN
; 1485          1476  3              RUNDOWN();
; 1486          1477  3          END
; 1487          1478  2      ELSE
; 1488          1479  2          IF .CONDITION_CODE [ STS$V_SEVERITY ] EQL STS$K_SUCCESS
; 1489          1480  2          OR .CONDITION_CODE [ STS$V_SEVERITY ] EQL STS$K_INFO
; 1490          1481  2          THEN
; 1491          1482  2              RETURN SS$_CONTINUE
; 1492          1483  2          ELSE
; 1493          1484  2              LIB$SIG_TO_RET ( .SIGNAL_VECTOR,.MECH_VECTOR );
; 1494          1485  2
; 1495          1486  2      ! If a signal then call resignal
; 1496          1487  2      !
; 1497          1488  2      RETURN SS$_RESIGNAL
; 1498          1489  2
; 1499          1490  1      END;


                                            000C 00000 CONDITION_HANDLER:
                                                                .WORD   Save R2,R3                          ; 1411
                        53        04   AC  D0 00002             MOVL    SIGNAL_VECTOR, R3                   ; 1458
                        52        04   A3  D0 00006             MOVL    4(R3), CONDITION_CODE               ;
              00000920  8F             52  D1 0000A             CMPL    CONDITION_CODE, #2336               ; 1462
                                       0C  13 00011             BEQL    1$                                  ;
                        0E     0000G   CF  E9 00013             BLBC    CONV$AB_FLAGS, 2$                   ; 1471
     04          52     03             00  ED 00018             CMPZV   #0, #3, CONDITION_CODE, #4          ; 1474
                                       25  12 0001D             BNEQ    5$                                  ;
                  FF6C  CF             00  FB 0001F  1$:        CALLS   #0, RUNDOWN                         ; 1476
                                       1E  11 00024             BRB     5$                                  ; 1471
     01          52     03             00  ED 00026  2$:        CMPZV   #0, #3, CONDITION_CODE, #1          ; 1479
                                       07  13 0002B             BEQL    3$                                  ;
     03          52     03             00  ED 0002D             CMPZV   #0, #3, CONDITION_CODE, #3          ; 1480
                                       04  12 00032             BNEQ    4$                                  ;
                        50             01  D0 00034  3$:        MOVL    #1, R0                              ; 1482
                                           04 00037             RET                                         ;
                                  08   AC  DD 00038  4$:        PUSHL   MECH_VECTOR                         ; 1484
                                       53  DD 0003B             PUSHL   R3                                  ;
             00000000G  00             02  FB 0003D             CALLS   #2, LIB$SIG_TO_RET                  ;
                        50      0918   8F  3C 00044  5$:        MOVZWL  #2328, R0                           ; 1488
                                           04 00049             RET                                         ; 1490

; Routine Size:  74 bytes,    Routine Base:  _CONV$CODE + 06D5


; 1500          1491  1
```

```
;  1502          1492  1 %SBTTL  'COPY_DESC'
;  1503          1493  1 ROUTINE COPY_DESC ( DESC ) : CL$COPY_DESC =
;  1504          1494  1
;  1505          1495  2     BEGIN
;  1506          1496  2
;  1507          1497  2     LOCAL
;  1508          1498  2         BYTES           : LONG,
;  1509          1499  2         LENGTH          : LONG,
;  1510          1500  2         BUFFER          : REF VECTOR [ ,BYTE ],
;  1511          1501  2         COPY_DESC       : REF BLOCK [ ,BYTE ];
;  1512          1502  2
;  1513          1503  2     GLOBAL REGISTER
;  1514          1504  2         ADDRESS = 1     : LONG;
;  1515          1505  2
;  1516          1506  2     LENGTH = STR$ANALYZE_SDESC_R1( .DESC );
;  1517          1507  2
;  1518          1508  2     ! Allocate a vm for a descriptor and a copy of the users string
;  1519          1509  2     !
;  1520          1510  2     BYTES = .LENGTH + 8;
;  1521          1511  2
;  1522          1512  2     ! Get the address of the descriptor
;  1523          1513  2     !
;  1524          1514  2     COPY_DESC = CONV$$GET_VM( .BYTES );
;  1525          1515  2
;  1526          1516  2     ! The string is just past that
;  1527          1517  2     !
;  1528          1518  2     BUFFER = .COPY_DESC + 8;
;  1529          1519  2
;  1530          1520  2     ! Stuff length
;  1531          1521  2     !
;  1532          1522  2     COPY_DESC [ DSC$W_LENGTH ] = .LENGTH;
;  1533          1523  2
;  1534          1524  2     ! Stuff address
;  1535          1525  2     !
;  1536          1526  2     COPY_DESC [ DSC$A_POINTER ] = .BUFFER;
;  1537          1527  2
;  1538          1528  2     ! Copy the user string
;  1539          1529  2     !
;  1540          1530  2     CH$MOVE( .LENGTH,.ADDRESS,.BUFFER );
;  1541          1531  2
;  1542          1532  2     ! Return address of descriptor
;  1543          1533  2     !
;  1544          1534  2     RETURN .COPY_DESC
;  1545          1535  2
;  1546          1536  1     END;


                 007C   8F  BB 00000 COPY_DESC:
                                        PUSHR   #^M<R2,R3,R4,R5,R6>                     ; 1493
            00000000G   00  16 00004            JSB     STR$ANALYZE_SDESC_R1            ; 1506
          52            50  D0 0000A            MOVL    R0, LENGTH                      ;
          50       08   A2  9E 0000D            MOVAB   8(R2), BYTES                    ; 1510
                        50  DD 00011            PUSHL   BYTES                           ; 1514
                    0000G   30 00013            BSBW    CONV$$GET_VM                    ;
```

```
                    5E              04 C0 00016        ADDL2   #4, SP                                ;
                    56              50 D0 00019        MOVL    R0, COPY_DESC                         ;
                    50       08     A6 9E 0001C        MOVAB   8(R6), BUFFER                         ; 1518
                    66              52 B0 00020        MOVW    LENGTH, (COPY_DESC)                   ; 1522
              04    A6              50 D0 00023        MOVL    BUFFER, 4(COPY_DESC)                  ; 1526
        60          61              52 28 00027        MOVC3   LENGTH, (ADDRESS), (BUFFER)           ; 1530
                    50              56 D0 0002B        MOVL    COPY_DESC, R0                         ; 1534
                           007C     8F BA 0002E        POPR    #^M<R2,R3,R4,R5,R6>                   ; 1536
                                       05 00032        RSB                                           ;

; Routine Size:  51 bytes,    Routine Base:  _CONV$CODE + 071F


; 1547           1537  1
; 1548           1538  0 END     ELUDOM




;                                 PSECT SUMMARY
;
;     Name                       Bytes                     Attributes
;
;   _CONV$OWN                      204  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;   _CONV$PLIT                       4  NOVEC,NOWRT,  RD ,NOEXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;   _CONV$CODE                    1874  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]LIB.L32;1           18619      114         0         1000         00:01.9
;   _$255$DUA28:[CONV.SRC]CONVERT.L32;1       165       25        15           17         00:00.2




;                                  COMMAND QUALIFIERS

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

; Size:          1874 code + 208 data bytes
; Run Time:         00:45.7
; Elapsed Time:     02:14.3
; Lines/CPU Min:     2020
; Lexemes/CPU-Min: 31811
; Memory Used:  297 pages
; Compilation Complete
```

COBRMSBLO
LIS

COBSPANC
LIS

COBSWITCH
LIS

COBRESTVA
LIS

CONVSHR
MAP

CONVCOMIO
LIS

COBSETSWI
LIS

CONVDEF
MDL

CONVLINK
REQ

COBSPANC2
LIS

CONV

CONVERT
MAP

CONVCALL
LIS

COBVECTOR
LIS

CONVERT
REQ

RECLAIM
MAP

ADDKEY
LIS