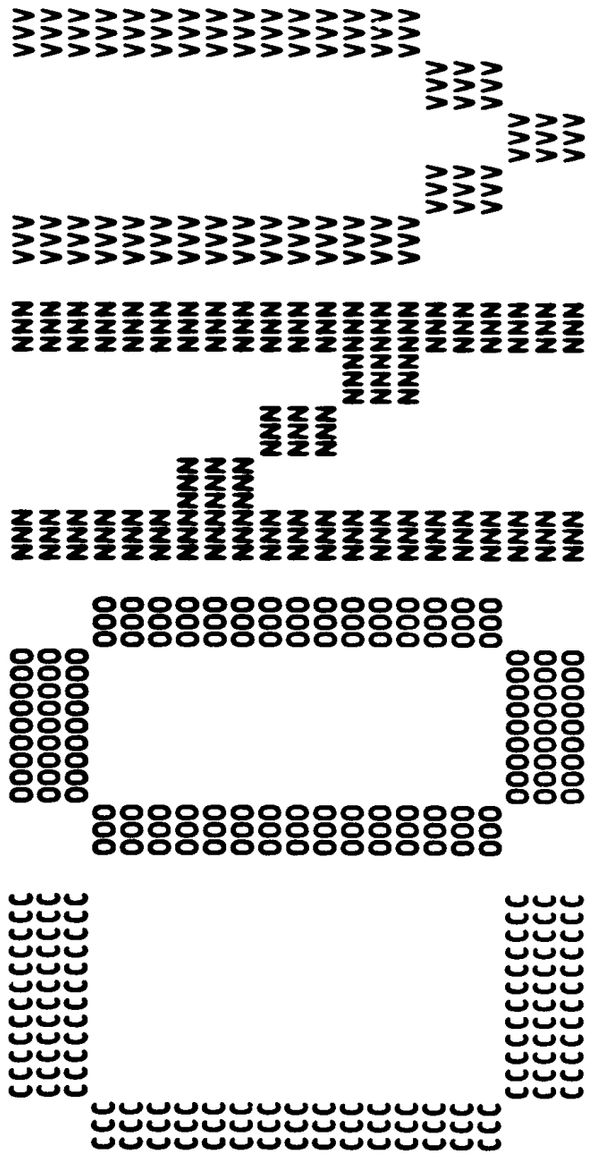

FILEID**CONVMAIN

N 2

```
  CCCCCCCC    000000    NN      NN  VV      VV  MM      MM    AAAAAA      IIIIII    NN      NN
  CCCCCCCC    000000    NN      NN  VV      VV  MM      MM    AAAAAA      IIIIII    NN      NN
CC            00  00    NN      NN  VV      VV  MMMM  MMMM  AA    AA      II      NN      NN
CC            00  00    NN      NN  VV      VV  MMMM  MMMM  AA    AA      II      NN      NN
CC            00  00    NNNN    NN  VV      VV  MM  MM  MM  AA    AA      II      NNNN    NN
CC            00  00    NNNN    NN  VV      VV  MM  MM  MM  AA    AA      II      NNNN    NN
CC            00  00    NN  NN  NN  VV      VV  MM      MM  AA    AA      II      NN  NN  NN
CC            00  00    NN  NN  NN  VV      VV  MM      MM  AA    AA      II      NN  NN  NN
CC            00  00    NN    NNNN  VV      VV  MM      MM  AAAAAAAAAA      II      NN    NNNN
CC            00  00    NN    NNNN  VV      VV  MM      MM  AAAAAAAAAA      II      NN    NNNN
CC            00  00    NN      NN    VV  VV    MM      MM  AA    AA      II      NN      NN      ....
CC            00  00    NN      NN    VV  VV    MM      MM  AA    AA      II      NN      NN      ....
  CCCCCCCC    000000    NN      NN      VV      MM      MM  AA    AA    IIIIII    NN      NN      ....
  CCCCCCCC    000000    NN      NN      VV      MM      MM  AA    AA    IIIIII    NN      NN      ....

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

```
;    1          0001  0 %TITLE  'VAX-11 CONVERT'
;    2          0002  0 MODULE  CONV$MAIN       ( 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 main routines
;    36          0035  1 !
;    37          0036  1 ! Contents:
;    38          0037  1 !                CONVERT
;    39          0038  1 !                GET_RECORD
;    40          0039  1 !                GET_VM
;    41          0040  1 !                FREE_VM
;    42          0041  1 !                GET_TEMP_VM
;    43          0042  1 !                FREE_TEMP_VM
;    44          0043  1 !                EXCEPTION
;    45          0044  1 !                END_OF_FILE
;    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:  June-1980
;    55          0054  1 !
;    56          0055  1 !
;    57          0056  1 ! Modified by:
;    58          0057  1 !
;    59          0058  1 !       V03-010 JWT0194         Jim Teague              31-Aug-1984
;    60          0059  1 !               Fix problem with missing data on UDF input.
;    61          0060  1 !
;    62          0061  1 !       V03-009 JWT0185         Jim Teague              28-Jun-1984
;    63          0062  1 !               Make FTN carriage control improvements:
;    64          0063  1 !
;    65          0064  1 !                       - NEVER just toss away FTN ccl bytes
;    66          0065  1 !                       - provide visually equivalent conversion
;    67          0066  1 !                               between FTN <---> PRN files.
;    68          0067  1 !                       - provide a visually equivalent
;    69          0068  1 !                               FTN ---> STM conversion.
;    70          0069  1 !
;    71          0070  1 !       V03-008 KBT0470         Keith B. Thompson       21-Jan-1983
;    72          0071  1 !               Add RSZ error code to exception
;    73          0072  1 !
;    74          0073  1 !       V03-007 KBT0466         Keith B. Thompson       21-Jan-1983
;    75          0074  1 !               Fix conv$$free_temp_vm
;    76          0075  1 !
;    77          0076  1 !       V03-006 KBT0434         Keith B. Thompson       16-Dec-1982
;    78          0077  1 !               Fix some of the carriage control checking
;    79          0078  1 !
;    80          0079  1 !       V03-005 KBT0387         Keith B. Thompson       27-Oct-1982
;    81          0080  1 !               Have get_vm and get_temp_vm zero the space allocated
;    82          0081  1 !
;    83          0082  1 !       V03-004 KBT0381         Keith B. Thompson       25-Oct-1982
;    84          0083  1 !               Make RMS$_SEQ error non-fatal
;    85          0084  1 !
;    86          0085  1 !       V03-003 KBT0376         Keith B. Thompson       22-Oct-1982
;    87          0086  1 !               Replace conv$v_dcl with conv$v_signal and make vm buffers
```

```
;    88          0087  1 !                 bigger
;    89          0088  1 !
;    90          0089  1 !       V03-002 KBT0346         Keith B. Thompson       4-Oct-1982
;    91          0090  1 !                 Use new linkage definitions
;    92          0091  1 !
;    93          0092  1 !       V03-001 KBT0016         Keith Thompson          18-Mar-1982
;    94          0093  1 !                 Add control-y processing in get_record and fix call to
;    95          0094  1 !                 conv$$rms_read_error in conv$$exception
;    96          0095  1 !
;    97          0096  1 !****
```

```
;   99          0097  1
;  100          0098  1 PSECT
;  101          0099  1         OWN     = _CONV$OWN     (PIC),
;  102          0100  1         GLOBAL  = _CONV$GLOBAL  (PIC),
;  103          0101  1         PLIT    = _CONV$PLIT    (SHARE,PIC),
;  104          0102  1         CODE    = _CONV$CODE    (SHARE,PIC);
;  105          0103  1
;  106          0104  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  107          0105  1 LIBRARY 'SRC$:CONVERT';
;  108          0106  1
;  109          0107  1 DEFINE_ERROR_CODES;
;  110          0108  1
;  111          0109  1 EXTERNAL ROUTINE
;  112          0110  1         LIB$GET_VM              : ADDRESSING_MODE(GENERAL),
;  113          0111  1         LIB$FREE_VM             : ADDRESSING_MODE(GENERAL),
;  114          0112  1         CONV$$SORT_PRIMARY      : CL$SORT_PRIMARY,
;  115          0113  1         CONV$$FAST_LOAD         : CL$FAST_LOAD,
;  116          0114  1         CONV$$RMS_READ_ERROR    : NOVALUE;
;  117          0115  1
;  118          0116  1 FORWARD ROUTINE
;  119          0117  1         CONV$$GET_RECORD        : CL$GET_RECORD,
;  120          0118  1         CONV$$GET_VM            : CL$GET_VM,
;  121          0119  1         CONV$$FREE_TEMP_VM      : CL$FREE_TEMP_VM       NOVALUE,
;  122          0120  1         CONV$$EXCEPTION,
;  123          0121  1         CONV$$END_OF_FILE       : NOVALUE,
;  124          0122  1         CONV$$PUT_RECORD,
;  125          0123  1         CONV$$MAP_FTN_CCL;
;  126          0124  1
;  127          0125  1 EXTERNAL
;  128          0126  1         CONV$GL_APPEND          : LONG,
;  129          0127  1         CONV$GL_CREATE          : LONG,
;  130          0128  1         CONV$GL_EXC             : LONG,
;  131          0129  1         CONV$GL_EXIT            : LONG,
;  132          0130  1         CONV$GL_FAST            : LONG,
;  133          0131  1         CONV$GL_FIX             : LONG,
;  134          0132  1         CONV$GL_MERGE           : LONG,
;  135          0133  1         CONV$GL_PAD             : LONG,
;  136          0134  1         CONV$GL_PAD_CHAR        : LONG,
;  137          0135  1         CONV$GL_SHARE           : LONG,
;  138          0136  1         CONV$GL_SORT            : LONG,
;  139          0137  1         CONV$GL_TRUNCATE        : LONG,
;  140          0138  1
;  141          0139  1         CONV$AB_FLAGS           : BLOCK [ ,BYTE ],
;  142          0140  1         CONV$GL_VALID_COUNT,
;  143          0141  1         CONV$GL_EXCEPT_COUNT,
;  144          0142  1         CONV$GB_CURRENT_FILE    : BYTE,
;  145          0143  1         CONV$GW_OUT_MRS         : WORD,
;  146          0144  1         CONV$GW_UDF_MRS         : WORD,
;  147          0145  1         CONV$GW_IN_REC_SIZ      : SIGNED WORD,
;  148          0146  1         CONV$GW_OUT_REC_SIZ     : SIGNED WORD,
;  149          0147  1         CONV$GL_RECORD_PTR,
;  150          0148  1         CONV$GL_RECORD_COUNT,
;  151          0149  1         CONV$GW_MAX_REC_SIZ     : WORD,
;  152          0150  1         CONV$GL_REC_BUF_PTR,
;  153          0151  1         CONV$GL_VFC_BUF_PTR,
;  154          0152  1         CONV$GL_RFA_BUFFER,
;  155          0153  1
```

```
;  156     0154  1          CONV$AB_RFA_FAB          : $FAB_DECL,
;  157     0155  1          CONV$AB_RFA_RAB          : $RAB_DECL,
;  158     0156  1          CONV$AB_IN_FAB           : $FAB_DECL,
;  159     0157  1          CONV$AB_IN_RAB           : $RAB_DECL,
;  160     0158  1          CONV$AB_IN_NAM           : $NAM_DECL,
;  161     0159  1          CONV$AB_OUT_FAB          : $FAB_DECL,
;  162     0160  1          CONV$AB_OUT_RAB          : $RAB_DECL,
;  163     0161  1          CONV$AB_OUT_NAM          : $NAM_DECL,
;  164     0162  1          CONV$AB_EXC_RAB          : $RAB_DECL,
;  165     0163  1          CONV$GL_STM_BUF,
;  166     0164  1          CONV$GL_STM_REC_LEN;
;  167     0165  1
;  168     0166  1      OWN
;  169     0167  1          DYN_AREA_CNT,
;  170     0168  1          DYN_AREA_ADDR    : VECTOR [ 32,LONG ],
;  171     0169  1          DYN_AREA_SIZE    : VECTOR [ 32,LONG ],
;  172     0170  1
;  173     0171  1          TMP_AREA_CNT,
;  174     0172  1          TMP_AREA_ADDR    : VECTOR [ 32,LONG ],
;  175     0173  1          TMP_AREA_SIZE    : VECTOR [ 32,LONG ],
;  176     0174  1
;  177     0175  1          BYTE_COUNT       : SIGNED LONG,
;  178     0176  1          N_RFAS,
;  179     0177  1          REC_ADJUST       : SIGNED LONG;          ! Record size adjustment
```

```
;  181          0178  1 %SBTTL  'CONVERT'
;  182          0179  1 GLOBAL ROUTINE  CONV$$CONVERT =
;  183          0180  1 !++
;  184          0181  1 !
;  185          0182  1 ! Functional Description:
;  186          0183  1 !
;  187          0184  1 !       Sets up the record buffer pointers and any format conversion,
;  188          0185  1 !       sorts the input file if needed and loads the output file.
;  189          0186  1 !
;  190          0187  1 !       How the buffers work:
;  191          0188  1 !
;  192          0189  1 !       The main record buffer:
;  193          0190  1 !
;  194          0191  1 !               ----------------------------------------//----------------
;  195          0192  1 !               !               !                                       !
;  196          0193  1 !               !               !                                       !
;  197          0194  1 !               ----------------------------------------//----------------
;  198          0195  1 !               ^               ^
;  199          0196  1 !               !               !
;  200          0197  1 !          VFC_BUF_PTR     REC_BUF_PTR
;  201          0198  1 !
;  202          0199  1 !
;  203          0200  1 !       No /FIXED_CONTROL or neither or both files VFC
;  204          0201  1 !
;  205          0202  1 !       Neither or both files FTN
;  206          0203  1 !
;  207          0204  1 ! INPUT RAB:   RHB             UBF
;  208          0205  1 !               !               !
;  209          0206  1 !               ----------------------------------------//----------------
;  210          0207  1 !               !               !                                       !
;  211          0208  1 !               !               !                                       !
;  212          0209  1 !               ----------------------------------------//----------------
;  213          0210  1 !               !               !
;  214          0211  1 ! OUTPUT RAB:  RHB             RBF
;  215          0212  1 !
;  216          0213  1 !
;  217          0214  1 !       Input file FTN
;  218          0215  1 !
;  219          0216  1 ! INPUT RAB:   RHB             UBF
;  220          0217  1 !               !               !
;  221          0218  1 !               ----------------------------------------//----------------
;  222          0219  1 !               !               !       !                               !
;  223          0220  1 !               !               !       !                               !
;  224          0221  1 !               ----------------------------------------//----------------
;  225          0222  1 !               !                       !
;  226          0223  1 ! OUTPUT RAB:  RHB                     RBF
;  227          0224  1 !
;  228          0225  1 !
;  229          0226  1 !       Output file FTN
;  230          0227  1 !
;  231          0228  1 ! INPUT RAB:   RHB             UBF
;  232          0229  1 !               !               !
;  233          0230  1 !               ----------------------------------------//----------------
;  234          0231  1 !               !       !       !                                       !
;  235          0232  1 !               !       !  20   !                                       !
;  236          0233  1 !               ----------------------------------------//----------------
;  237          0234  1 !               !       !
```

```
;  238     0235  1 ! OUTPUT RAB:  RHB     RBF
;  239     0236  1 !
;  240     0237  1 !
;  241     0238  1 !       Ouput file PRN input file not VFC
;  242     0239  1 !
;  243     0240  1 ! INPUT RAB:   RHB             UBF
;  244     0241  1 !               |               |
;  245     0242  1 !               ----------------------------------------//----------------
;  246     0243  1 !               |                                                       |
;  247     0244  1 !               |   01  :  8D                                           |
;  248     0245  1 !               ----------------------------------------//----------------
;  249     0246  1 !               |               |
;  250     0247  1 ! OUTPUT RAB:  RHB             RBF
;  251     0248  1 !
;  252     0249  1 !
;  253     0250  1 !       With /FIXED_CONTROL
;  254     0251  1 !
;  255     0252  1 !       Input file VFC, output file not:
;  256     0253  1 !
;  257     0254  1 ! INPUT RAB:   RHB             UBF
;  258     0255  1 !               |               |
;  259     0256  1 !               ----------------------------------------//----------------
;  260     0257  1 !               |               |                                       |
;  261     0258  1 !               |               |                                       |
;  262     0259  1 !               ----------------------------------------//----------------
;  263     0260  1 !               |
;  264     0261  1 ! OUTPUT RAB:  RBF
;  265     0262  1 !
;  266     0263  1 !
;  267     0264  1 !       Output file VFC, input file not:
;  268     0265  1 !
;  269     0266  1 ! INPUT RAB:   UBF
;  270     0267  1 !               |
;  271     0268  1 !               ----------------------------------------//----------------
;  272     0269  1 !               |               |                                       |
;  273     0270  1 !               |               |                                       |
;  274     0271  1 !               ----------------------------------------//----------------
;  275     0272  1 !               |               |
;  276     0273  1 ! OUTPUT RAB:  RHB             RBF
;  277     0274  1 !
;  278     0275  1 !
;  279     0276  1 ! Calling Sequence:
;  280     0277  1 !
;  281     0278  1 !       CONV$$CONVERT()
;  282     0279  1 !
;  283     0280  1 ! Input Parameters:
;  284     0281  1 !       none
;  285     0282  1 !
;  286     0283  1 ! Implicit Inputs:
;  287     0284  1 !       none
;  288     0285  1 !
;  289     0286  1 ! Output Parameters:
;  290     0287  1 !       none
;  291     0288  1 !
;  292     0289  1 ! Implicit Outputs:
;  293     0290  1 !       none
;  294     0291  1 !
```

```
;  295     0292  1 ! Routine Value:
;  296     0293  1 !
;  297     0294  1 !       CONV$_SUCCESS or error code
;  298     0295  1 !
;  299     0296  1 ! Routines Called:
;  300     0297  1 !
;  301     0298  1 !       CONV$$SORT_PRIMARY
;  302     0299  1 !       CONV$$FAST_LOAD
;  303     0300  1 !       CONV$$GET_RECORD
;  304     0301  1 !       $PUT
;  305     0302  1 !       CONV$$EXCEPTION
;  306     0303  1 !       CONV$$END_OF_FILE
;  307     0304  1 !       CONV$$FREE_TEMP_VM
;  308     0305  1 !
;  309     0306  1 ! Side Effects:
;  310     0307  1 !       none
;  311     0308  1 !
;  312     0309  1 !--
;  313     0310  1
;  314     0311  2     BEGIN
;  315     0312  2
;  316     0313  2     DEFINE_KEY_DESC_GLOBAL;
;  317     0314  2
;  318     0315  2 !******************
;  319     0316  2
;  320     0317  2
;  321     0318  2     CONV$AB_FLAGS [ CONV$V_MAPFTN ] = 0;
;  322     0319  2
;  323     0320  2 !******************
;  324     0321  2
;  325     0322  2     ! Setup Buffer Pointers etc.
;  326     0323  2     !
;  327     0324  2     ! If the FIX Option is on and only one of the files
;  328     0325  2     ! has VFC format will pointers be moved
;  329     0326  2     !
;  330     0327  2     IF .CONV$GL_FIX AND
;  331     0328  3        (( .CONV$AB_IN_FAB [ FAB$B_RFM ] EQL FAB$C_VFC ) XOR
;  332     0329  3         ( .CONV$AB_OUT_FAB [ FAB$B_RFM ] EQL FAB$C_VFC ))
;  333     0330  2     THEN
;  334     0331  2
;  335     0332  2         IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQL FAB$C_VFC
;  336     0333  2         THEN
;  337     0334  3             BEGIN
;  338     0335  3
;  339     0336  3             ! The Input File is VFC
;  340     0337  3             !
;  341     0338  3             REC_ADJUST = .CONV$AB_IN_FAB [ FAB$B_FSZ ];
;  342     0339  3             CONV$AB_IN_RAB [ RAB$C_RAB ] = .CONV$GL_VFC_BUF_PTR;
;  343     0340  3             CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$GL_REC_BUF_PTR;
;  344     0341  3             CONV$AB_OUT_RAB [ RAB$C_RBF ] = .CONV$GL_VFC_BUF_PTR
;  345     0342  3             END
;  346     0343  2         ELSE
;  347     0344  3             BEGIN
;  348     0345  3
;  349     0346  3             ! The Output File is VFC
;  350     0347  3             !
;  351     0348  3             REC_ADJUST = - .CONV$AB_OUT_FAB [ FAB$B_FSZ ];
```

```
;  352     0349  3              CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$GL_VFC_BUF_PTR;
;  353     0350  3              CONV$AB_OUT_RAB [ RAB$L_RHB ] = .CONV$GL_VFC_BUF_PTR;
;  354     0351  3              CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_REC_BUF_PTR
;  355     0352  3              END
;  356     0353  2          ELSE
;  357     0354  3              BEGIN
;  358     0355  3
;  359     0356  3              ! Either they Both are or are not VFC files
;  360     0357  3              !
;  361     0358  3              REC_ADJUST = 0;
;  362     0359  3              CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$GL_REC_BUF_PTR;
;  363     0360  3              CONV$AB_IN_RAB [ RAB$L_RHB ] = .CONV$GL_VFC_BUF_PTR;
;  364     0361  3              CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_REC_BUF_PTR;
;  365     0362  3              CONV$AB_OUT_RAB [ RAB$L_RHB ] = .CONV$GL_VFC_BUF_PTR;
;  366     0363  3
;  367     0364  3              ! Take care of some special cases
;  368     0365  3              !
;  369     0366  3              ! If the input is FTN and the output isn't...
;  370     0367  3              !
;  371     0368  3              IF .CONV$AB_IN_FAB [ FAB$V_FTN ] AND NOT .CONV$AB_OUT_FAB [ FAB$V_FTN ]
;  372     0369  3              THEN
;  373     0370  3                  !
;  374     0371  3                  ! Output is PRN -- do a carriage control approximation
;  375     0372  3                  !
;  376     0373  3                  IF .CONV$AB_OUT_FAB [ FAB$V_PRN ]
;  377     0374  3                  THEN
;  378     0375  4                      BEGIN
;  379     0376  4                      CONV$AB_FLAGS [ CONV$V_MAPFTN ] = CONV$C_FTNPRN;                    ! (1)
;  380     0377  4                      CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$AB_OUT_RAB [ RAB$L_RBF ] + 1;
;  381     0378  4                      REC_ADJUST = -1;
;  382     0379  4                      END
;  383     0380  3                  ELSE
;  384     0381  3                      !
;  385     0382  3                      ! Output is CR:STM -- do visual approximation
;  386     0383  3                      !
;  387     0384  3                      IF .CONV$AB_OUT_FAB [ FAB$B_RFM ] EQL FAB$C_STM
;  388     0385  3                      THEN
;  389     0386  4                          BEGIN
;  390     0387  4                          CONV$AB_FLAGS [ CONV$V_MAPFTN ] = CONV$C_FTNSTM ;    ! (2)
;  391     0388  4                          CONV$GL_STM_BUF = CONV$$GET_VM ( STM_BUF_SIZ );
;  392     0389  4                          CONV$GL_STM_REC_LEN = 0;
;  393     0390  4                          CONV$AB_FLAGS [ CONV$V_FIRST_REC ] = 1;
;  394     0391  4                          CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$AB_IN_RAB [ RAB$L_UBF ] + 1;
;  395     0392  4                          REC_ADJUST = 0;
;  396     0393  3                          END;
;  397     0394  3
;  398     0395  3          ! If the output file is FTN and the input file is not ...
;  399     0396  3          !
;  400     0397  3          IF .CONV$AB_OUT_FAB [ FAB$V_FTN ] AND NOT .CONV$AB_IN_FAB [ FAB$V_FTN ]
;  401     0398  3          THEN
;  402     0399  3              !
;  403     0400  3              ! If PRN --> FTN, then do an approximation...
;  404     0401  3              !
;  405     0402  3              IF .CONV$AB_IN_FAB [ FAB$V_PRN ]
;  406     0403  3              THEN
;  407     0404  4                  BEGIN
;  408     0405  4                  CONV$AB_FLAGS [ CONV$V_MAPFTN ] = CONV$C_PRNFTN;                    ! (3)
```

```
;  409     0406  4              CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$AB_IN_RAB [ RAB$L_UBF ] + 1;
;  410     0407  4              REC_ADJUST = 1;
;  411     0408  4              END
;  412     0409  3          ELSE
;  413     0410  4              BEGIN
;  414     0411  4              LOCAL RECORD_BUFFER : REF VECTOR [ ,BYTE ];
;  415     0412  4
;  416     0413  4              RECORD_BUFFER = .CONV$AB_IN_RAB [ RAB$L_UBF ];
;  417     0414  4
;  418     0415  4              RECORD_BUFFER [ 0 ] = 20;                       ! 20 = ASCII space
;  419     0416  4
;  420     0417  4              CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$AB_IN_RAB [ RAB$L_UBF ] + 1;
;  421     0418  4              REC_ADJUST = 1
;  422     0419  4
;  423     0420  3              END;
;  424     0421  3
;  425     0422  3      ! If the output file is PRN and the input is not VFC
;  426     0423  3      ! then put LF - CR control info into the VFC field of the output
;  427     0424  3      ! NOTE: VMS print files are "018D" in stead of "8A8D" which would
;  428     0425  3      !       be more reasonable?
;  429     0426  3      !
;  430     0427  3      IF .CONV$AB_OUT_FAB [ FAB$V_PRN ] AND
;  431     0428  4                              ( .CONV$AB_IN_FAB [ FAB$B_RFM ] NEQU FAB$C_VFC )
;  432     0429  3      THEN
;  433     0430  4          BEGIN
;  434     0431  4
;  435     0432  4          LOCAL RECORD_BUFFER : REF VECTOR [ ,BYTE ];
;  436     0433  4
;  437     0434  4          RECORD_BUFFER = .CONV$AB_OUT_RAB [ RAB$L_RHB ];
;  438     0435  4
;  439     0436  4          RECORD_BUFFER [ 0 ] = %X'01';                   ! One LF
;  440     0437  4          RECORD_BUFFER [ 1 ] = %X'8D'                    ! CR
;  441     0438  4
;  442     0439  3          END;
;  443     0440  3
;  444     0441  2          END;
;  445     0442  2
;  446     0443  2      ! The exception record is always the record pointed to by the output rab
;  447     0444  2      !
;  448     0445  2      CONV$AB_EXC_RAB [ RAB$L_RBF ] = .CONV$AB_OUT_RAB [ RAB$L_RBF ];
;  449     0446  2
;  450     0447  2      ! Save the pointer to the output record
;  451     0448  2      !
;  452     0449  2      CONV$GL_RECORD_PTR = .CONV$AB_OUT_RAB [ RAB$L_RBF ];
;  453     0450  2
;  454     0451  2      ! Initialize some Variables
;  455     0452  2      !
;  456     0453  2      CONV$AB_IN_RAB [ RAB$W_USZ ] = .CONV$GW_MAX_REC_SIZ;
;  457     0454  2      BYTE_COUNT = 0;
;  458     0455  2
;  459     0456  2      ! If SORT is on then sort the records on the output primary key
;  460     0457  2      !
;  461     0458  2      IF .CONV$GL_SORT
;  462     0459  2      THEN
;  463     0460  2          RET_ON_ERROR( CONV$$SORT_PRIMARY() );
;  464     0461  2
;  465     0462  3      BEGIN                ! STATUS local
```

```
;  466          0463  3
;  467          0464  3      LOCAL       STATUS  : LONG;
;  468          0465  3
;  469          0466  3      ! If FAST then call FAST_LOAD, otherwise do it the slow way
;  470          0467  3      !
;  471          0468  3      IF .CONV$GL_FAST
;  472          0469  3      THEN
;  473          0470  3          STATUS = CONV$$FAST_LOAD()
;  474          0471  3      ELSE
;  475          0472  4          BEGIN
;  476          0473  4
;  477          0474  4          ! Restore the output rab pointer (sort_primary destroys it)
;  478          0475  4          !
;  479          0476  4          CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_RECORD_PTR;
;  480          0477  4
;  481          0478  4          ! Enter Main Loop
;  482          0479  4          !
;  483          0480  4          ! Loop untill an error or end of file
;  484          0481  4          !
;  485          0482  5          WHILE ( STATUS = CONV$$GET_RECORD() )
;  486          0483  4          DO
;  487          0484  4              IF .CONV$AB_FLAGS [ CONV$V_MAPFTN ] NEQ CONV$C_FTNSTM
;  488          0485  4              THEN
;  489          0486  5                  IF NOT ( STATUS = CONV$$PUT_RECORD () )
;  490          0487  4                  THEN
;  491          0488  4                      EXITLOOP;
;  492          0489  4
;  493          0490  4          ! Finish off this file
;  494          0491  4          !
;  495          0492  4          CONV$$END_OF_FILE()
;  496          0493  4
;  497          0494  3          END;
;  498          0495  3
;  499          0496  3      ! Deallocate all of the temporary memory used by this run
;  500          0497  3      !
;  501          0498  3      CONV$$FREE_TEMP_VM();
;  502          0499  3
;  503          0500  3      ! If end of file thats normal
;  504          0501  3      !
;  505          0502  4      IF ( .STATUS EQL RMS$_EOF )
;  506          0503  3      THEN
;  507          0504  3          STATUS = CONV$_SUCCESS;
;  508          0505  3
;  509          0506  3      RETURN .STATUS
;  510          0507  3
;  511          0508  3      END                     ! STATUS local
;  512          0509  1      END;


                                                                .TITLE  CONV$MAIN VAX-11 CONVERT
                                                                .IDENT  \V04-000\

                                                                .PSECT  _CONV$OWN,NOEXE,  PIC,2

                                                    00000 DYN_AREA_CNT:
                                                                .BLKB   4
                                                    00004 DYN_AREA_ADDR:
```

```
                        .BLKB   128
00084 DYN_AREA_SIZE:
                        .BLKB   128
00104 IMP_AREA_CNT:
                        .BLKB   4
00108 TMP_AREA_ADDR:
                        .BLKB   128
00188 TMP_AREA_SIZE:
                        .BLKB   128
00208 BYTE_COUNT:
                        .BLKB   4
0020C N_RFAS: .BLKB     4
00210 REC_ADJUST:
                        .BLKB   4

                        .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$GET_VM, LIB$FREE_VM
                        .EXTRN  CONV$$SORT_PRIMARY
                        .EXTRN  CONV$$FAST_LOAD
                        .EXTRN  CONV$$RMS_READ_ERROR
                        .EXTRN  CONV$GL_APPEND, CONV$GL_CREATE
                        .EXTRN  CONV$GL_EXC, CONV$GL_EXIT
                        .EXTRN  CONV$GL_FAST, CONV$GL_FIX
                        .EXTRN  CONV$GL_MERGE, CONV$GL_PAD
                        .EXTRN  CONV$GL_PAD_CHAR
                        .EXTRN  CONV$GL_SHARE, CONV$GL_SORT
                        .EXTRN  CONV$GL_TRUNCATE
                        .EXTRN  CONV$AB_FLAGS, CONV$GL_VALID_COUNT
                        .EXTRN  CONV$GL_EXCEPT_COUNT
                        .EXTRN  CONV$GB_CURRENT_FILE
                        .EXTRN  CONV$GW_OUT_MRS
                        .EXTRN  CONV$GW_UDF_MRS
                        .EXTRN  CONV$GW_IN_REC_SIZ
```

```
                                        .EXTRN  CONV$GW_OUT_REC_SIZ
                                        .EXTRN  CONV$GL_RECORD_PTR
                                        .EXTRN  CONV$GL_RECORD_COUNT
                                        .EXTRN  CONV$GW_MAX_REC_SIZ
                                        .EXTRN  CONV$GL_REC_BUF_PTR
                                        .EXTRN  CONV$GL_VFC_BUF_PTR
                                        .EXTRN  CONV$GL_RFA_BUFFER
                                        .EXTRN  CONV$AB_RFA_FAB
                                        .EXTRN  CONV$AB_RFA_RAB
                                        .EXTRN  CONV$AB_IN_FAB, CONV$AB_IN_RAB
                                        .EXTRN  CONV$AB_IN_NAM, CONV$AB_OUT_FAB
                                        .EXTRN  CONV$AB_OUT_RAB
                                        .EXTRN  CONV$AB_OUT_NAM
                                        .EXTRN  CONV$AB_EXC_RAB
                                        .EXTRN  CONV$GL_STM_BUF
                                        .EXTRN  CONV$GL_STM_REC_LEN

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

                          0FFC 00000    .ENTRY  CONV$$CONVERT, Save R2,R3,R4,R5,R6,R7,R8,-  ; 0179
                                                R9,R10,R11                                  ;
58      0000G   CF  9E 00002            MOVAB   CONV$AB_FLAGS+2, R8                         ;
57      0000G   CF  9E 00007            MOVAB   CONV$AB_OUT_FAB+30, R7                      ;
56      0000G   CF  9E 0000C            MOVAB   CONV$AB_IN_FAB+31, R6                       ;
55      0000G   CF  9E 00011            MOVAB   CONV$AB_IN_RAB+36, R5                       ;
54      0000'   CF  9E 00016            MOVAB   REC_ADJUST, R4                              ;
53      0000G   CF  9E 0001B            MOVAB   CONV$AB_OUT_RAB+40, R3                      ;
68      0180    8F  AA 00020            BICW2   #384, CONV$AB_FLAGS+2                       ; 0318
5B      0000G   CF  D0 00025            MOVL    CONV$GL_VFC_BUF_PTR, R11                    ; 0339
52      0000G   CF  D0 0002A            MOVL    CONV$GL_REC_BUF_PTR, R2                     ; 0340
41      0000G   CF  E9 0002F            BLBC    CONV$GL_FIX, 5$                             ; 0327
                51  D4 00034            CLRL    R1                                          ; 0328
03              66  91 00036            CMPB    CONV$AB_IN_FAB+31, #3                       ;
                02  12 00039            BNEQ    1$                                          ;
                51  D6 0003B            INCL    R1                                          ;
                50  D4 0003D 1$:         CLRL    R0                                          ; 0329
03      01      A7  91 0003F            CMPB    CONV$AB_OUT_FAB+31, #3                      ;
                02  12 00043            BNEQ    2$                                          ;
                50  D6 00045            INCL    R0                                          ;
50              51  C0 00047 2$:         ADDL2   R1, R0                                      ;
28              50  E9 0004A            BLBC    R0, 5$                                      ;
03              66  91 0004D            CMPB    CONV$AB_IN_FAB+31, #3                       ; 0332
                10  12 00050            BNEQ    3$                                          ;
64      20      A6  9A 00052            MOVZBL  CONV$AB_IN_FAB+63, REC_ADJUST               ; 0338
08  A5                  5B  D0 00056    MOVL    R11, CONV$AB_IN_RAB+44                      ; 0339
65                      52  D0 0005A    MOVL    R2, CONV$AB_IN_RAB+36                       ; 0340
63                      5B  D0 0005D    MOVL    R11, CONV$AB_OUT_RAB+40                     ; 0341
                11  11 00060            BRB     4$                                          ;
64      21      A7  9A 00062 3$:         MOVZBL  CONV$AB_OUT_FAB+63, REC_ADJUST              ; 0348
64              64  CE 00066            MNEGL   REC_ADJUST, REC_ADJUST                      ;
65                      5B  D0 00069    MOVL    R11, CONV$AB_IN_RAB+36                      ; 0349
04  A3                  5B  D0 0006C    MOVL    R11, CONV$AB_OUT_RAB+44                     ; 0350
63                      52  D0 00070    MOVL    R2, CONV$AB_OUT_RAB+40                      ; 0351
                7E  11 00073 4$:         BRB     12$                                         ; 0332
                64  D4 00075 5$:         CLRL    REC_ADJUST                                  ; 0358
65                      52  D0 00077    MOVL    R2, CONV$AB_IN_RAB+36                       ; 0359
08  A5                  5B  D0 0007A    MOVL    R11, CONV$AB_IN_RAB+44                      ; 0360
```

```
                        63              52  D0 0007E          MOVL    R2, CONV$AB_OUT_RAB+40                  ; 0361
                     04 A3              5B  D0 00081          MOVL    R11, CONV$AB_OUT_RAB+44                 ; 0362
                        3A        FF    A6  E9 00085          BLBC    CONV$AB_IN_FAB+30, 7$                   ; 0368
                        3A              67  E8 00089          BLBS    CONV$AB_OUT_FAB+30, 8$                  ;
                     0C                 67  02 E1 0008C          BBC     #2, CONV$AB_OUT_FAB+30, 6$              ; 0373
                68   02                 07  01 F0 00090          INSV    #1, #7, #2, CONV$AB_FLAGS+2             ; 0376
                                        63  D6 00095          INCL    CONV$AB_OUT_RAB+40                      ; 0377
                                        64  01 CE 00097          MNEGL   #1, REC_ADJUST                          ; 0378
                                        27  11 0009A          BRB     7$                                      ; 0373
                                        04    01  A7  91 0009C 6$:    CMPB    CONV$AB_OUT_FAB+31, #4          ; 0384
                                        21  12 000A0          BNEQ    7$                                      ;
                68   02                 07  02 F0 000A2          INSV    #2, #7, #2, CONV$AB_FLAGS+2             ; 0387
                                        7E  7FFE  8F  3C 000A7          MOVZWL  #32766, -(SP)                   ; 0388
                                          0000V 30  000AC          BSBW    CONV$$GET_VM                    ;
                                        5E              04  C0 000AF          ADDL2   #4, SP                  ;
                                     0000G  CF          50  D0 000B2          MOVL    R0, CONV$GL_STM_BUF     ;
                                            0000G  CF   D4  000B7          CLRL    CONV$GL_STM_REC_LEN             ; 0389
                                        01  A8          04  88 000BB          BISB2   #4, CONV$AB_FLAGS+3             ; 0390
                                                       65  D6 000BF          INCL    CONV$AB_IN_RAB+36               ; 0391
                                                       64  D4 000C1          CLRL    REC_ADJUST                      ; 0392
                                        1B                 67  E9 000C3 7$:    BLBC    CONV$AB_OUT_FAB+30, 11$         ; 0397
                                        17    FF    A6  E8 000C6 8$:    BLBS    CONV$AB_IN_FAB+30, 11$          ;
                                        64              01  D0 000CA          MOVL    #1, REC_ADJUST                  ; 0407
                     07    FF  A6                      02  E1 000CD          BBC     #2, CONV$AB_IN_FAB+30, 9$       ; 0402
                                        68    0180  8F  A8 000D2          BISW2   #384, CONV$AB_FLAGS+2           ; 0405
                                                       06  11 000D7          BRB     10$                             ; 0402
                                        50                 65  D0 000D9 9$:    MOVL    CONV$AB_IN_RAB+36, RECORD_BUFFER ; 0413
                                        60                 14  90 000DC          MOVB    #20, (RECORD_BUFFER)            ; 0415
                                                           65  D6 000DF 10$:    INCL    CONV$AB_IN_RAB+36               ; 0406
                     0E                 67                  02  E1 000E1 11$:    BBC     #2, CONV$AB_OUT_FAB+30, 12$     ; 0427
                                        03                 66  91 000E5          CMPB    CONV$AB_IN_FAB+31, #3           ; 0428
                                                           09  13 000E8          BEQL    12$                             ;
                                        50    04    A3  D0 000EA          MOVL    CONV$AB_OUT_RAB+44, RECORD_BUFFER ; 0434
                                        60    8D01  8F  B0 000EE          MOVW    #36097, (RECORD_BUFFER)         ; 0436
                                     0000G  CF          63  D0 000F3 12$:    MOVL    CONV$AB_OUT_RAB+40, CONV$AB_EXC_RAB+40 ; 0445
                                     0000G  CF          63  D0 000F8          MOVL    CONV$AB_OUT_RAB+40, CONV$GL_RECORD_PTR ; 0449
                                       FC   A5    0000G  CF  B0 000FD          MOVW    CONV$GW_MAX_REC_SIZ, CONV$AB_IN_RAB+32 ; 0453
                                              F8   A4  D4 00103          CLRL    BYTE_COUNT                      ; 0454
                                        06    0000G  CF  E9 00106          BLBC    CONV$GL_SORT, 13$               ; 0458
                                                 0000G 30  0010B          BSBW    CONV$$SORT_PRIMARY              ; 0460
                                        44              50  E9 0010E          BLBC    STATUS, 19$                     ;
                                        08    0000G  CF  E9 00111 13$:    BLBC    CONV$GL_FAST, 14$               ; 0468
                                                 0000G 30  00116          BSBW    CONV$$FAST_LOAD                 ; 0470
                                        52              50  D0 00119          MOVL    R0, STATUS                      ;
                                                        25  11 0011C          BRB     17$                             ;
                                        63    0000G  CF  D0 0011E 14$:    MOVL    CONV$GL_RECORD_PTR, CONV$AB_OUT_RAB+40 ; 0476
                                                 0000V 30  00123 15$:    BSBW    CONV$$GET_RECORD                ; 0482
                                        52              50  D0 00126          MOVL    R0, STATUS                      ;
                                        12              52  E9 00129          BLBC    STATUS, 16$                     ;
                02   68                 02                  07  ED 0012C          CMPZV   #7, #2, CONV$AB_FLAGS+2, #2     ; 0484
                                                           F0  13 00131          BEQL    15$                             ;
                                     0000V  CF          00  FB 00133          CALLS   #0, CONV$$PUT_RECORD            ; 0486
                                        52              50  D0 00138          MOVL    R0, STATUS                      ;
                                        E5              52  E8 0013B          BLBS    STATUS, 15$                     ;
                                     0000V  CF          00  FB 0013E 16$:    CALLS   #0, CONV$$END_OF_FILE           ; 0492
                                                 0000V 30  00143 17$:    BSBW    CONV$$FREE_TEMP_VM              ; 0498
                           0001827A 8F                   52  D1 00146          CMPL    STATUS, #98938          ; 0502
```

```
                                03  12 0014D           BNEQ    18$                                             ;
                           52   01  D0 0014F           MOVL    #1, STATUS                                      ; 0504
                           50   52  D0 00152 18$:      MOVL    STATUS, R0                                      ; 0506
                                    04 00155 19$:      RET                                                     ; 0509

; Routine Size:  342 bytes,    Routine Base:  _CONV$CODE + 0000
```

```
;  514          0510  1 %SBTTL  'GET_RECORD'
;  515          0511  1 GLOBAL ROUTINE  CONV$$GET_RECORD : CL$GET_RECORD =
;  516          0512  1 !++
;  517          0513  1 !
;  518          0514  1 ! Functional Description:
;  519          0515  1 !
;  520          0516  1 !       Gets a record from the input file and processes it
;  521          0517  1 !
;  522          0518  1 ! Calling Sequence:
;  523          0519  1 !
;  524          0520  1 !       CONV$$GET_RECORD()
;  525          0521  1 !
;  526          0522  1 ! Input Parameters:
;  527          0523  1 !       none
;  528          0524  1 !
;  529          0525  1 ! Implicit Inputs:
;  530          0526  1 !       none
;  531          0527  1 !
;  532          0528  1 ! Output Parameters:
;  533          0529  1 !       none
;  534          0530  1 !
;  535          0531  1 ! Implicit Outputs:
;  536          0532  1 !       none
;  537          0533  1 !
;  538          0534  1 ! Routine Value:
;  539          0535  1 !
;  540          0536  1 !       CONV$_SUCCESS or status returned by $GET
;  541          0537  1 !
;  542          0538  1 ! Routines Called:
;  543          0539  1 !
;  544          0540  1 !       $READ
;  545          0541  1 !       CONV$$RMS_READ_ERROR    - By RMS as an AST
;  546          0542  1 !       $GET
;  547          0543  1 !       CONV$$EXCEPTION
;  548          0544  1 !
;  549          0545  1 ! Side Effects:
;  550          0546  1 !       none
;  551          0547  1 !
;  552          0548  1 !--
;  553          0549  1
;  554          0550  2     BEGIN
;  555          0551  2
;  556          0552  2     LABEL
;  557          0553  2         GET_REC,
;  558          0554  2         EXC_REC;
;  559          0555  2
;  560          0556  2     OWN
;  561          0557  2         RFA_IDX         : LONG;
;  562          0558  2
;  563          0559  2     LOCAL
;  564          0560  2         STATUS          : LONG;
;  565          0561  2
;  566          0562  2     ! RFA Vector declarations
;  567          0563  2     !
;  568          0564  2     BIND
;  569          0565  2         RFA0    = .CONV$GL_RFA_BUFFER : RFA0_VECTOR [ 256 ],
;  570          0566  2         RFA4    = .CONV$GL_RFA_BUFFER : RFA4_VECTOR [ 256 ];
```

```
;   571         0567  2
;   572         0568  2      STATUS = _SET;
;   573         0569  2
;   574         0570  2      ! Loop while ok (we could fail from a fatal exception)
;   575         0571  2      !
;   576         0572  2      WHILE .STATUS
;   577         0573  2      DO
;   578         0574  3          BEGIN           ! Main While loop
;   579         0575  3
;   580         0576  3  GET_REC:
;   581         0577  4              BEGIN       ! GET_REC Block
;   582         0578  4
;   583         0579  4              ! If Input Format is UDF then we do some strange things
;   584         0580  4              !
;   585         0581  4              IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQLU FAB$C_UDF
;   586         0582  4              THEN
;   587         0583  5                  BEGIN   ! UDF Block
;   588         0584  5
;   589         0585  5                  ! If the Byte Count from last time is Positive then we must move
;   590         0586  5                  ! the Chracters Left Over to the Front of the Buffer
;   591         0587  5                  !
;   592         0588  5                  IF .BYTE_COUNT GEQ .CONV$GW_UDF_MRS
;   593         0589  5                  THEN
;   594         0590  6                      BEGIN
;   595         0591  6
;   596         0592  6                      ! Set status here because we dont do a READ
;   597         0593  6                      !
;   598         0594  6                      STATUS = CONV$_SUCCESS;
;   599         0595  6
;   600         0596  6                      ! Cut down BYTE_COUNT the size of the last record
;   601         0597  6                      !
;   602         0598  6                      BYTE_COUNT = .BYTE_COUNT - .CONV$GW_UDF_MRS;
;   603         0599  6
;   604         0600  6                      ! Move the extra characters to the front of the buffer
;   605         0601  6                      !
;   606         0602  6                      CH$MOVE ( .BYTE_COUNT,
;   607         0603  6                                .CONV$GL_REC_BUF_PTR + .CONV$GW_UDF_MRS,
;   608         0604  6                                .CONV$GL_REC_BUF_PTR )
;   609         0605  5                      END;
;   610         0606  5
;   611         0607  5                  ! Read some Blocks to Get the Record
;   612         0608  5                  !
;   613         0609  5                  WHILE .BYTE_COUNT LSS .CONV$GW_UDF_MRS
;   614         0610  5                  DO
;   615         0611  6                      BEGIN       ! READ While loop
;   616         0612  6
;   617         0613  6                      ! Point the RAB buffer to the position just after the last
;   618         0614  6                      ! character left over from last time
;   619         0615  6                      !
;   620         0616  6                      CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$GL_REC_BUF_PTR + .BYTE_COUNT;
;   621         0617  6
;   622       P 0618  8                      IF ( NOT ( STATUS = $READ( RAB=CONV$AB_IN_RAB,
;   623         0619  7                                                 ERR=CONV$$RMS_READ_ERROR ) ) )
;   624         0620  6                      THEN
;   625         0621  6
;   626         0622  6                          ! If Byte Count is > 0 then use that and give EOF
;   627         0623  6                          ! next time
```

```
;  628          0624  6                    !
;  629          0625  6                    IF .BYTE_COUNT GTR 0
;  630          0626  6                    THEN
;  631          0627  7                        BEGIN
;  632          0628  7                        CONV$GW_IN_REC_SIZ = .BYTE_COUNT;
;  633          0629  7                        BYTE_COUNT = 0;
;  634          0630  7                        STATUS = _SET;
;  635          0631  7                        LEAVE GET_REC
;  636          0632  7                        END
;  637          0633  6                    ELSE
;  638          0634  6                        RETURN .STATUS;
;  639          0635  6
;  640          0636  6                BYTE_COUNT = .BYTE_COUNT + .CONV$AB_IN_RAB [ RAB$W_RSZ ];
;  641          0637  6
;  642          0638  6                ! If we got a Short Record then we are also done
;  643          0639  6                !
;  644          0640  6                IF .CONV$AB_IN_RAB [ RAB$W_RSZ ] LSSU BLOCK_SIZE
;  645          0641  6                THEN
;  646          0642  6
;  647          0643  6                    ! Check to see if the short part put us over the limit
;  648          0644  6                    !
;  649          0645  6                    IF .BYTE_COUNT GTR .CONV$GW_UDF_MRS
;  650          0646  6                    THEN
;  651          0647  6                        EXITLOOP
;  652          0648  6                    ELSE
;  653          0649  7                        BEGIN
;  654          0650  7                        CONV$GW_IN_REC_SIZ = .BYTE_COUNT;
;  655          0651  7                        BYTE_COUNT = 0;
;  656          0652  7                        LEAVE GET_REC
;  657          0653  6                        END;
;  658          0654  6
;  659          0655  5                END;        ! READ While Loop
;  660          0656  5
;  661          0657  5                ! Get Ready for Next Time
;  662          0658  5                !
;  663          0659  5                CONV$GW_IN_REC_SIZ = .CONV$GW_UDF_MRS
;  664          0660  5
;  665          0661  5                END     ! UDF Block
;  666          0662  5
;  667          0663  5            ! For NON UDF Files do a Regular Get
;  668          0664  5            !
;  669          0665  4            ELSE
;  670          0666  5                BEGIN   ! Regular GET Block
;  671          0667  5
;  672          0668  5                ! If Reading a file by an RFA file first get the RFA
;  673          0669  5                !
;  674          0670  5                IF .CONV$AB_FLAGS [ CONV$V_RFA ]
;  675          0671  5                THEN
;  676          0672  6                    BEGIN
;  677          0673  6
;  678          0674  6                    ! If we ran out of RFAs get some more
;  679          0675  6                    !
;  680          0676  6                    IF .N_RFAS EQL 0
;  681          0677  6                    THEN
;  682        P 0678  8                        IF ( NOT ( STATUS = $READ( RAB=CONV$AB_RFA_RAB,
;  683          0679  7                                                   ERR=CONV$$RMS_READ_ERROR ) ) )
;  684          0680  6                        THEN
```

```
;  685          0681  6                    RETURN .STATUS
;  686          0682  6                ELSE
;  687          0683  7                    BEGIN
;  688          0684  7
;  689          0685  7                    ! RFAs are six bytes long so we can tell how
;  690          0686  7                    ! many there are
;  691          0687  7                    !
;  692          0688  7                    N_RFAS = .CONV$AB_RFA_RAB [ RAB$W_RSZ ] / 6;
;  693          0689  7                    RFA_IDX = 0
;  694          0690  7
;  695          0691  6                    END;
;  696          0692  6
;  697          0693  6                ! Get an RFA
;  698          0694  6                !
;  699          0695  6                CONV$AB_IN_RAB [ RAB$L_RFA0 ] = .RFA0 [ .RFA_IDX ];
;  700          0696  6                CONV$AB_IN_RAB [ RAB$W_RFA4 ] = .RFA4 [ .RFA_IDX ];
;  701          0697  6
;  702          0698  6                RFA_IDX = .RFA_IDX + 1;
;  703          0699  6                N_RFAS = .N_RFAS - 1
;  704          0700  6
;  705          0701  5                END;
;  706          0702  5
;  707          0703  5            ! Finally get a record!  If the i/o was complete under
;  708          0704  5            ! control-y just ignore it and try again
;  709          0705  5            !
;  710          0706  5            DO
;  711          0707  5
;  712          0708  5            ! rms_read_error will return if the error was end of file
;  713          0709  5            !
;  714        P 0710  8            ( IF ( NOT ( STATUS = $GET( RAB = CONV$AB_IN_RAB,
;  715          0711  7                                      ERR = CONV$$RMS_READ_ERROR ) ) )
;  716          0712  6                THEN
;  717          0713  6                    RETURN .STATUS )
;  718          0714  6
;  719          0715  5                WHILE .STATUS EQLU RMS$_CONTROLY;
;  720          0716  5
;  721          0717  5                ! Set the input record size
;  722          0718  5                !
;  723          0719  5                CONV$GW_IN_REC_SIZ = .CONV$AB_IN_RAB [ RAB$W_RSZ ]
;  724          0720  5
;  725          0721  4                END;    ! Regular GET Block
;  726          0722  4
;  727          0723  3            END;        ! GET_REC Block
;  728          0724  3
;  729          0725  3
;  730          0726  3            ! We have a record so count it
;  731          0727  3            !
;  732          0728  3            CONV$GL_RECORD_COUNT = .CONV$GL_RECORD_COUNT + 1;
;  733          0729  3
;  734          0730  3            ! If we need to do FTN carriage control mappings,
;  735          0731  3            !       then go for it...
;  736          0732  3            !
;  737          0733  3            IF .CONV$AB_FLAGS [ CONV$V_MAPFTN ] NEQ 0
;  738          0734  3            THEN
;  739          0735  4                IF NOT ( STATUS = CONV$$MAP_FTN_CCL () )
;  740          0736  3                THEN
;  741          0737  3                    RETURN .STATUS;
```

```
;   742     0738  3
;   743     0739  3          ! Make adjustments on the record size if necessary
;   744     0740  3          !
;   745     0741  3          CONV$GW_IN_REC_SIZ = .CONV$GW_IN_REC_SIZ + .REC_ADJUST;
;   746     0742  3
;   747     0743  3
;   748     0744  3 ! Check fo exception conditions ie. is the record to long to short ect...
;   749     0745  3 ! The exception record block has a series of test to check the valididy
;   750     0746  3 ! of the record if an exception record is found exit of the block will
;   751     0747  3 ! repeat the process of getting a record with out exiting.  If all checks
;   752     0748  3 ! suceed then we drop through an exit
;   753     0749  3 !
;   754     0750  3 EXC_PEC:
;   755     0751  4                 BEGIN       ! EXC_REC Block
;   756     0752  4
;   757     0753  4                 ! If record size is negative then the input record
;   758     0754  4                 ! is to short to fill the VFC portion of the output file
;   759     0755  4                 !
;   760     0756  4                 IF .CONV$GW_IN_REC_SIZ LSS 0
;   761     0757  4                 THEN
;   762     0758  5                     BEGIN
;   763     0759  5                     CONV$GW_IN_REC_SIZ = .CONV$GW_IN_REC_SIZ - .REC_ADJUST;
;   764     0760  5                     STATUS = CONV$$EXCEPTION( CONV$_VFC );
;   765     0761  5                     LEAVE EXC_REC
;   766     0762  4                     END;
;   767     0763  4
;   768     0764  4                 ! If Output File MRS is 0 then we check for index key size if necc.
;   769     0765  4                 !
;   770     0766  4                 IF .CONV$GW_OUT_MRS EQL 0
;   771     0767  4                 THEN
;   772     0768  4                     CONV$GW_OUT_REC_SIZ = .CONV$GW_IN_REC_SIZ
;   773     0769  4                 ELSE
;   774     0770  4
;   775     0771  4                     ! If the Record is Longer then MRS Check for Truncate or
;   776     0772  4                     ! Exception
;   777     0773  4                     !
;   778     0774  4                     IF .CONV$GW_IN_REC_SIZ GTR .CONV$GW_OUT_MRS
;   779     0775  4                     THEN
;   780     0776  4
;   781     0777  4                         ! If Truncate then set the Record's size to MRS
;   782     0778  4                         !
;   783     0779  4                         IF .CONV$GL_TRUNCATE
;   784     0780  4                         THEN
;   785     0781  4                             CONV$GW_OUT_REC_SIZ = .CONV$GW_OUT_MRS
;   786     0782  4                         ELSE
;   787     0783  5                             BEGIN
;   788     0784  5                             STATUS = CONV$$EXCEPTION( CONV$_RTL );
;   789     0785  5                             LEAVE EXC_REC
;   790     0786  5                             END
;   791     0787  4                     ELSE
;   792     0788  4
;   793     0789  4                         ! If the file is fixed an the record is short...
;   794     0790  4                         !
;   795     0791  4                         IF ( .CONV$AB_OUT_FAB [ FAB$B_RFM ] EQL FAB$C_FIX ) AND
;   796     0792  5                              ( .CONV$GW_IN_REC_SIZ LSS .CONV$GW_OUT_MRS )
;   797     0793  4                         THEN
;   798     0794  4
```

```
;   799     0795  4                     ! If PAD then pad the short record
;   800     0796  4                     !
;   801     0797  4                     IF .CONV$GL_PAD
;   802     0798  4                     THEN
;   803     0799  5                         BEGIN
;   804     0800  5                         CH$FILL ( .CONV$GL_PAD_CHAR,
;   805     0801  5                                   .CONV$GW_OUT_MRS - .CONV$GW_IN_REC_SIZ,
;   806     0802  5                                   .CONV$GL_REC_BUF_PTR + .CONV$GW_IN_REC_SIZ );
;   807     0803  5                         CONV$GW_OUT_REC_SIZ = .CONV$GW_OUT_MRS
;   808     0804  5                         END
;   809     0805  4                     ELSE
;   810     0806  5                         BEGIN
;   811     0807  5                         STATUS = CONV$$EXCEPTION( CONV$_RTS );
;   812     0808  5                         LEAVE EXC_REC
;   813     0809  5                         END
;   814     0810  4                 ELSE
;   815     0811  4                     CONV$GW_OUT_REC_SIZ = .CONV$GW_IN_REC_SIZ;
;   816     0812  4
;   817     0813  4             ! If we made it this far then we have an ok record
;   818     0814  4             !
;   819     0815  4             RETURN .STATUS
;   820     0816  4
;   821     0817  3             END;        ! EXC_REC Block
;   822     0818  3
;   823     0819  2         END;            ! Main While loop
;   824     0820  2
;   825     0821  2         RETURN .STATUS
;   826     0822  2
;   827     0823  1     END;


                                            .PSECT  _CONV$OWN,NOEXE,  PIC,2

                                    00214 RFA_IDX:.BLKB   4

                                            .EXTRN  SYS$READ, SYS$GET

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

                                    03FC    8F  BB 00000 CONV$$GET_RECORD::
                                                PUSHR   #^M<R2,R3,R4,R5,R6,R7,R8,R9>                    ; 0511
                                59  0000G   CF  D0 00004         MOVL    CONV$GL_RFA_BUFFER, R9          ; 0565
                                58  0000G   CF  D0 00009         MOVL    CONV$GL_RFA_BUFFER, R8          ; 0566
                                57          01  D0 0000E         MOVL    #1, STATUS                      ; 0568
                                03              57  E8 00011 1$:  BLBS    STATUS, 3$                      ; 0572
                                            01A0  31 00014 2$:  BRW     21$                             ;
                                0000G   CF      95 00017 3$:    TSTB    CONV$AB_IN_FAB+31               ; 0581
                                            03  13 0001B         BEQL    4$                              ;
                                          0092  31 0001D         BRW     9$                              ;
                                50  0000G   CF  3C 00020 4$:    MOVZWL  CONV$GW_UDF_MRS, R0             ; 0588
                                50  0000'    F  D1 00025         CMPL    BYTE_COUNT, R0                  ;
                                            13  19 0002A         BLSS    5$                              ;
                                57          01  D0 0002C         MOVL    #1, STATUS                      ; 0594
                          0000'   CF        50  C2 0002F         SUBL2   R0, BYTE_COUNT                  ; 0598
                0000G DF  0000GDF40     0000'   CF  28 00034    MOVC3   BYTE_COUNT, @CONV$GL_REC_BUF_PTR[R0], -  ; 0604
                                                                        @CONV$GL_REC_BUF_PTR            ;
```

```
0000' CF     0000G CF          10          00 ED 0003F 5$:   CMPZV   #0, #16, CONV$GW_UDF_MRS, BYTE_COUNT      ; 0609
                                           5F 15 00048       BLEQ    7$
             0000G CF     0000G CF     0000' CF C1 0004A      ADDL3   BYTE_COUNT, CONV$GL_REC_BUF_PTR, -       ; 0616
                                                                       CONV$AB_IN_RAB+36
                                     0000G CF 9F 00054       PUSHAB  CONV$$RMS_READ_ERROR                      ; 0619
                                     0000G CF 9F 00058       PUSHAB  CONV$AB_IN_RAB
                  00000000G  00              02 FB 0005C       CALLS   #2, SYS$READ
                             57              50 D0 00063       MOVL    R0, STATUS
                             15              57 E8 00066       BLBS    STATUS, 6$
                             50     0000' CF 50 D0 00069       MOVL    BYTE_COUNT, R0                           ; 0625
                                           A4 15 0006E       BLEQ    2$
                       0000G CF             50 B0 00070       MOVW    R0, CONV$GW_IN_REC_SIZ                   ; 0628
                                     0000' CF D4 00075       CLRL    BYTE_COUNT                               ; 0629
                             57              01 D0 00079       MOVL    #1, STATUS                               ; 0630
                                           32 11 0007C       BRB     8$                                       ; 0631
                             50     0000G CF 3C 0007E 6$:     MOVZWL  CONV$AB_IN_RAB+34, R0                    ; 0636
                       0000' CF             50 C0 00083       ADDL2   R0, BYTE_COUNT
                       0200  8F     0000G CF B1 00088       CMPW    CONV$AB_IN_RAB+34, #512                  ; 0640
                                           AE 1E 0008F       BGEQU   5$
0000' CF     0000G CF          10          00 ED 00091       CMPZV   #0, #16, CONV$GW_UDF_MRS, BYTE_COUNT      ; 0645
                                           0D 19 0009A       BLSS    7$
                       0000G CF     0000' CF B0 0009C       MOVW    BYTE_COUNT, CONV$GW_IN_REC_SIZ           ; 0650
                                     0000' CF D4 000A3       CLRL    BYTE_COUNT                               ; 0651
                                           7D 11 000A7       BRB     12$                                      ; 0652
                       0000G CF     0000G CF B0 000A9 7$:     MOVW    CONV$GW_UDF_MRS, CONV$GW_IN_REC_SIZ      ; 0659
                                           74 11 000B0 8$:   BRB     12$
                  49   0000G CF              04 E1 000B2 9$:   BBC     #4, CONV$AB_FLAGS+2, 11$                 ; 0670
                                     0000' CF D5 000B8       TSTL    N_RFAS                                   ; 0676
                                           24 12 000BC       BNEQ    10$
                                     0000G CF 9F 000BE       PUSHAB  CONV$$RMS_READ_ERROR                     ; 0679
                                     0000G CF 9F 000C2       PUSHAB  CONV$AB_RFA_RAB
                  00000000G  00              02 FB 000C6       CALLS   #2, SYS$READ
                             57              50 D0 000CD       MOVL    R0, STATUS
                             68              57 E9 000D0       BLBC    STATUS, 13$
                             50     0000G CF 3C 000D3       MOVZWL  CONV$AB_RFA_RAB+34, R0                   ; 0688
0000' CF                     50              06 C7 000D8       DIVL3   #6, R0, N_RFAS
                                     0000' CF D4 000DE       CLRL    RFA_IDX                                  ; 0689
       50    0000' CF                      06 C5 000E2 10$:   MULL3   #6, RFA_IDX, R0                          ; 0695
                                        6049 9F 000E8       PUSHAB  (R0)[R9]
                       0000G CF             9E D0 000EB       MOVL    @(SP)+, CONV$AB_IN_RAB+16
                                          04 A048 9F 000F0       PUSHAB  4(R0)[R8]                            ; 0696
                       0000G CF             9E B0 000F4       MOVW    @(SP)+, CONV$AB_IN_RAB+20
                                     0000' CF D6 000F9       INCL    RFA_IDX                                  ; 0698
                                     0000' CF D7 000FD       DECL    N_RFAS                                   ; 0699
                                     0000G CF 9F 00101 11$:    PUSHAB  CONV$$RMS_READ_ERROR                   ; 0711
                                     0000G CF 9F 00105         PUSHAB  CONV$AB_IN_RAB
                  00000000G  00              02 FB 00109       CALLS   #2, SYS$GET
                             57              50 D0 00110       MOVL    R0, STATUS
                             25              57 E9 00113       BLBC    STATUS, 13$
                  00010611   8F              57 D1 00116       CMPL    STATUS, #67089                           ; 0715
                                           E2 13 0011D       BEQL    11$
                       0000G CF     0000G CF B0 0011F       MOVW    CONV$AB_IN_RAB+34, CONV$GW_IN_REC_SIZ    ; 0719
                                     0000G CF D6 00126 12$:    INCL    CONV$GL_RECORD_COUNT                     ; 0728
                       0180  8F     0000G CF B3 0012A       BITW    CONV$AB_FLAGS+2, #384                    ; 0733
                                           0B 13 00131       BEQL    14$
                       0000V CF              00 FB 00133       CALLS   #0, CONV$$MAP_FTN_CCL                    ; 0735
                             57              50 D0 00138       MOVL    R0, STATUS
```

```
                                 79          57  E9 0013B 13$:     BLBC     STATUS, 21$                                    ;
                        0000G    CF    0000'  CF  A0 0013E 14$:     ADDW2    REC_ADJUST, CONV$GW_IN_REC_SIZ                 ; 0741
                                 50    0000G  CF  32 00145          CVTWL    CONV$GW_IN_REC_SIZ, R0                         ; 0756
                                              10  18 0014A          BGEQ     15$                                            ;
              0000G  CF          50    0000'  CF  A3 0014C          SUBW3    REC_ADJUST, R0, CONV$GW_IN_REC_SIZ             ; 0759
                                   00000000G  8F  DD 00154          PUSHL    #CONV$_VFC                                     ; 0760
                                              4B  11 0015A          BRB      19$                                            ;
                                 52    0000G  CF  32 0015C 15$:     CVTWL    CONV$GW_IN_REC_SIZ, R2                         ; 0768
                                 56    0000G  CF  3C 00161          MOVZWL   CONV$GW_OUT_MRS, R6                            ; 0766
                                              4A  13 00166          BEQL     20$                                            ;
                                 56           52  D1 00168          CMPL     R2, R6                                         ; 0774
                                              0D  15 0016B          BLEQ     16$                                            ;
                                 28    0000G  CF  E8 0016D          BLBS     CONV$GL_TRUNCATE, 17$                          ; 0779
                                   00000000G  8F  DD 00172          PUSHL    #CONV$_RTL                                     ; 0784
                                              2D  11 00178          BRB      19$                                            ;
                                 01    0000G  CF  91 0017A 16$:     CMPB     CONV$AB_OUT_FAB+31, #1                         ; 0791
                                              31  12 0017F          BNEQ     20$                                            ;
                                 56           52  D1 00181          CMPL     R2, R6                                         ; 0792
                                              2C  18 00184          BGEQ     20$                                            ;
                                 16    0000G  CF  E9 00186          BLBC     CONV$GL_PAD, 18$                               ; 0797
                       50        56           52  C3 0018B          SUBL3    R2, R6, R0                                     ; 0801
     50       0000G    CF        6E           00  2C 0018F          MOVC5    #0, (SP), CONV$GL_PAD_CHAR, R0, -              ; 0802
                                       0000GDF42     00196                   @CONV$GL_REC_BUF_PTR[R2]                       ;
                        0000G    CF           56  B0 0019A 17$:     MOVW     R6, CONV$GW_OUT_REC_SIZ                        ; 0803
                                              16  11 0019F          BRB      21$                                            ;
                                   00000000G  8F  DD 001A1 18$:     PUSHL    #CONV$_RTS                                     ; 0807
                        0000V    CF           01  FB 001A7 19$:     CALLS    #1, CONV$$EXCEPTION                            ;
                                 57           50  D0 001AC          MOVL     R0, STATUS                                     ;
                                            FE5F  31 001AF          BRW      1$                                             ; 0808
                        0000G    CF           52  B0 001B2 20$:     MOVW     R2, CONV$GW_OUT_REC_SIZ                        ; 0811
                                 50           57  D0 001B7 21$:     MOVL     STATUS, R0                                     ; 0821
                                        03FC  8F  BA 001BA          POPR     #^M<R2,R3,R4,R5,R6,R7,R8,R9>                   ; 0823
                                              05 001BE              RSB                                                     ;

; Routine Size:  447 bytes,    Routine Base:  _CONV$CODE + 0156
```

```
;  829          0824  1 %SBTTL  'GET_VM'
;  830          0825  1 GLOBAL ROUTINE  CONV$$GET_VM ( BYTES ) : CL$GET_VM =
;  831          0826  1 !++
;  832          0827  1 !
;  833          0828  1 ! Functional Description:
;  834          0829  1 !
;  835          0830  1 !       Allocates virtual memory and records it so that it can
;  836          0831  1 !       be returned by CONV$$FREE_VM
;  837          0832  1 !
;  838          0833  1 ! Calling Sequence:
;  839          0834  1 !
;  840          0835  1 !       CONV$$GET_VM( bytes )
;  841          0836  1 !
;  842          0837  1 ! Input Parameters:
;  843          0838  1 !
;  844          0839  1 !       bytes   - Number of bytes to allocate
;  845          0840  1 !
;  846          0841  1 ! Implicit Inputs:
;  847          0842  1 !       none
;  848          0843  1 !
;  849          0844  1 ! Output Parameters:
;  850          0845  1 !       none
;  851          0846  1 !
;  852          0847  1 ! Implicit Outputs:
;  853          0848  1 !       none
;  854          0849  1 !
;  855          0850  1 ! Routine Value:
;  856          0851  1 !
;  857          0852  1 !       R0 -    Address of start of memory allocated
;  858          0853  1 !
;  859          0854  1 ! Routines called:
;  860          0855  1 !
;  861          0856  1 !       LIB$GET_VM
;  862          0857  1 !
;  863          0858  1 ! Side Effects:
;  864          0859  1 !       none
;  865          0860  1 !
;  866          0861  1 !--
;  867          0862  1
;  868          0863  2     BEGIN
;  869          0864  2
;  870          0865  2     LOCAL       POINTER;
;  871          0866  2
;  872          0867  2     ! Allocate the memory and quit if errors
;  873          0868  2     !
;  874          0869  2     IF NOT LIB$GET_VM( BYTES,POINTER )
;  875          0870  2     THEN
;  876          0871  2         SIGNAL_STOP( CONV$_INSVIRMEM );
;  877          0872  2
;  878          0873  2     ! Keep track of allocated memory so we can give it back latter
;  879          0874  2     !
;  880          0875  2     DYN_AREA_SIZE [ .DYN_AREA_CNT ] = .BYTES;
;  881          0876  2     DYN_AREA_ADDR [ .DYN_AREA_CNT ] = .POINTER;
;  882          0877  2     DYN_AREA_CNT = .DYN_AREA_CNT + 1;
;  883          0878  2
;  884          0879  2     ! Zero the allocated space
;  885          0880  2     !
```

```
;  886          0881  2     CH$FILL( 0,.BYTES,.POINTER );
;  887          0882  2
;  888          0883  2     RETURN .POINTER
;  889          0884  2
;  890          0885  1     END;


                                        3E  BB 00000 CONV$$GET_VM::
                                                     PUSHR   #^M<R1,R2,R3,R4,R5>                          ; 0825
                              5E                04   C2 00002           SUBL2   #4, SP
                              5E                5E   DD 00005           PUSHL   SP                           ; 0869
                                          20    AE   9F 00007           PUSHAB  BYTES
                    00000000G   00              02   FB 0000A           CALLS   #2, LIB$GET_VM
                                00              50   E8 00011           BLBS    R0, 1$
                                  00000000G     8F   DD 00014           PUSHL   #CONV$_INSVIRMEM             ; 0871
                    00000000G   00              01   FB 0001A           CALLS   #1, LIB$STOP
                                50        0000' CF   D0 00021 1$:        MOVL    DYN_AREA_CNT, R0             ; 0875
                        0000'CF40           1C   AE   D0 00026           MOVL    BYTES, DYN_AREA_SIZE[R0]
                        0000'CF40                6E   D0 0002D           MOVL    POINTER, DYN_AREA_ADDR[R0]   ; 0876
                                          0000' CF   D6 00033           INCL    DYN_AREA_CNT                 ; 0877
      1C    AE              00                6E   00   2C 00037           MOVC5   #0, (SP), #0, BYTES, @POINTER ; 0881
                                             00   BE      0003D
                                50                8E   D0 0003F           MOVL    POINTER, R0                  ; 0883
                                                  3E   BA 00042           POPR    #^M<R1,R2,R3,R4,R5>          ; 0885
                                                       05 00044           RSB                                  ;

; Routine Size:  69 bytes,    Routine Base:  _CONV$CODE + 0315
```

```
;  892          0886  1 %SBTTL  'FREE_VM'
;  893          0887  1 GLOBAL ROUTINE CONV$$FREE_VM : CL$FREE_VM NOVALUE =
;  894          0888  1 !++
;  895          0889  1 !
;  896          0890  1 ! Functional Description:
;  897          0891  1 !
;  898          0892  1 !       Returns virtual memory allocated by CONV$$GET_VM
;  899          0893  1 !
;  900          0894  1 ! Calling Sequence:
;  901          0895  1 !
;  902          0896  1 !       CONV$$FREE_VM()
;  903          0897  1 !
;  904          0898  1 ! Input Parameters:
;  905          0899  1 !       none
;  906          0900  1 !
;  907          0901  1 ! Implicit Inputs:
;  908          0902  1 !       none
;  909          0903  1 !
;  910          0904  1 ! Output Parameters:
;  911          0905  1 !       none
;  912          0906  1 !
;  913          0907  1 ! Implicit Outputs:
;  914          0908  1 !       none
;  915          0909  1 !
;  916          0910  1 ! Routine Value:
;  917          0911  1 !       none
;  918          0912  1 !
;  919          0913  1 ! Routines called:
;  920          0914  1 !
;  921          0915  1 !       LIB$FREE_VM
;  922          0916  1 !
;  923          0917  1 ! Side Effects:
;  924          0918  1 !       none
;  925          0919  1 !
;  926          0920  1 !--
;  927          0921  1
;  928          0922  2     BEGIN
;  929          0923  2
;  930          0924  2     ! Deallocate memory
;  931          0925  2     !
;  932          0926  3     WHILE ( .DYN_AREA_CNT NEQ 0 )
;  933          0927  2     DO
;  934          0928  3         BEGIN
;  935          0929  3         DYN_AREA_CNT = .DYN_AREA_CNT - 1;
;  936          0930  3         LIB$FREE_VM(    DYN_AREA_SIZE [ .DYN_AREA_CNT ],
;  937          0931  3                         DYN_AREA_ADDR [ .DYN_AREA_CNT ] )
;  938          0932  2         END;
;  939          0933  2
;  940          0934  2     RETURN
;  941          0935  2
;  942          0936  1     END;


                                0000'  CF   D5 00000 CONV$$FREE_VM::
```

```
                                                        TSTL    DYN_AREA_CNT                            ; 0926
                                                1C  13  BEQL    1$                                      ;
                                        0000'   CF  D7  DECL    DYN_AREA_CNT                            ; 0929
                                    50  0000'   CF  D0  MOVL    DYN_AREA_CNT, R0                        ; 0931
                                        0000'CF40  DF  PUSHAL   DYN_AREA_ADDR[R0]                       ;
                                        0000'CF40  DF  PUSHAL   DYN_AREA_SIZE[R0]                       ; 0930
                            00000000G   00      02  FB  CALLS   #2, LIB$FREE_VM                        ; 0931
                                            DE  11  BRB     CONV$$FREE_VM                               ;
                                                05  RSB     1$:                                     ; 0936

; Routine Size:  35 bytes,    Routine Base:  _CONV$CODE + 035A
```

```
;  944          0937  1 %SBTTL  'GET_TEMP_VM'
;  945          0938  1 GLOBAL ROUTINE  CONV$$GET_TEMP_VM ( BYTES ) : CL$GET_TEMP_VM =
;  946          0939  1 !++
;  947          0940  1 !
;  948          0941  1 ! Functional Description:
;  949          0942  1 !
;  950          0943  1 !       Allocates virtual memory and records it so that it can
;  951          0944  1 !       be returned by CONV$$FREE_TEMP_VM
;  952          0945  1 !
;  953          0946  1 ! Calling Sequence:
;  954          0947  1 !
;  955          0948  1 !       CONV$$GET_TEMP_VM( bytes )
;  956          0949  1 !
;  957          0950  1 ! Input Parameters:
;  958          0951  1 !
;  959          0952  1 !       bytes   - Number of bytes to allocate
;  960          0953  1 !
;  961          0954  1 ! Implicit Inputs:
;  962          0955  1 !       none
;  963          0956  1 !
;  964          0957  1 ! Output Parameters:
;  965          0958  1 !       none
;  966          0959  1 !
;  967          0960  1 ! Implicit Outputs:
;  968          0961  1 !       none
;  969          0962  1 !
;  970          0963  1 ! Routine Value:
;  971          0964  1 !
;  972          0965  1 !       R0 -    Address of start of memory allocated
;  973          0966  1 !
;  974          0967  1 ! Routines called:
;  975          0968  1 !
;  976          0969  1 !       LIB$GET_VM
;  977          0970  1 !
;  978          0971  1 ! Side Effects:
;  979          0972  1 !       none
;  980          0973  1 !
;  981          0974  1 !--
;  982          0975  1
;  983          0976  2     BEGIN
;  984          0977  2
;  985          0978  2     LOCAL       POINTER;
;  986          0979  2
;  987          0980  2     ! Allocate the memory and quit if errors
;  988          0981  2     !
;  989          0982  2     IF NOT LIB$GET_VM( BYTES,POINTER )
;  990          0983  2     THEN
;  991          0984  2         SIGNAL_STOP( CONV$_INSVIRMEM );
;  992          0985  2
;  993          0986  2     ! Keep track of allocated memory so we can give it back latter
;  994          0987  2     !
;  995          0988  2     TMP_AREA_SIZE [ .TMP_AREA_CNT ] = .BYTES;
;  996          0989  2     TMP_AREA_ADDR [ .TMP_AREA_CNT ] = .POINTER;
;  997          0990  2     TMP_AREA_CNT = .TMP_AREA_CNT + 1;
;  998          0991  2
;  999          0992  2     ! Zero the allocated space
; 1000          0993  2     !
```

```
;  1001          0994  2     CH$FILL( 0,.BYTES,.POINTER );
;  1002          0995  2
;  1003          0996  2     RETURN .POINTER
;  1004          0997  2
;  1005          0998  1     END;


                                                3E  BB 00000 CONV$$GET_TEMP_VM::
                                                             PUSHR   #^M<R1,R2,R3,R4,R5>                             ; 0938
                                      5E                04  C2 00002             SUBL2   #4, SP                      ;
                                                        5E  DD 00005             PUSHL   SP                          ; 0982
                                                 20     AE  9F 00007             PUSHAB  BYTES                       ;
                                00000000G         00        02  FB 0000A             CALLS   #2, LIB$GET_VM              ;
                                                  0D        50  E8 00011             BLBS    R0, 1$                      ;
                                                     00000000G  8F  DD 00014             PUSHL   #CONV$_INSVIRMEM            ; 0984
                                00000000G         00        01  FB 0001A             CALLS   #1, LIB$STOP                ;
                                                  50     0000'  CF  D0 00021 1$:     MOVL    TMP_AREA_CNT, R0            ; 0988
                                    0000'CF40            1C  AE  D0 00026             MOVL    BYTES, TMP_AREA_SIZE[R0]    ;
                                    0000'CF40            6E  D0 0002D             MOVL    POINTER, TMP_AREA_ADDR[R0]  ; 0989
                                                     0000'  CF  D6 00033             INCL    TMP_AREA_CNT                ; 0990
    1C  AE              00                6E              00  2C 00037             MOVC5   #0, (SP), #0, BYTES, @POINTER ; 0994
                                                        00  BE     0003D            ;
                                      50                8E  D0 0003F             MOVL    POINTER, R0                 ; 0996
                                                        3E  BA 00042             POPR    #^M<R1,R2,R3,R4,R5>         ; 0998
                                                            05 00044             RSB                                 ;

; Routine Size:  69 bytes,    Routine Base:  _CONV$CODE + 037D
```

```
;  1007          0999  1 %SBTTL  'FREE_TEMP_VM'
;  1008          1000  1 GLOBAL ROUTINE CONV$$FREE_TEMP_VM : CL$FREE_TEMP_VM NOVALUE =
;  1009          1001  1 !++
;  1010          1002  1 !
;  1011          1003  1 ! Functional Description:
;  1012          1004  1 !
;  1013          1005  1 !       Returns virtual memory allocated by CONV$$GET_TEMP_VM
;  1014          1006  1 !
;  1015          1007  1 ! Calling Sequence:
;  1016          1008  1 !
;  1017          1009  1 !       CONV$$FREE_TEMP_VM()
;  1018          1010  1 !
;  1019          1011  1 ! Input Parameters:
;  1020          1012  1 !       none
;  1021          1013  1 !
;  1022          1014  1 ! Implicit Inputs:
;  1023          1015  1 !       none
;  1024          1016  1 !
;  1025          1017  1 ! Output Parameters:
;  1026          1018  1 !       none
;  1027          1019  1 !
;  1028          1020  1 ! Implicit Outputs:
;  1029          1021  1 !       none
;  1030          1022  1 !
;  1031          1023  1 ! Routine Value:
;  1032          1024  1 !       none
;  1033          1025  1 !
;  1034          1026  1 ! Routines called:
;  1035          1027  1 !
;  1036          1028  1 !       LIB$FREE_VM
;  1037          1029  1 !
;  1038          1030  1 ! Side Effects:
;  1039          1031  1 !       none
;  1040          1032  1 !
;  1041          1033  1 !--
;  1042          1034  1
;  1043          1035  2     BEGIN
;  1044          1036  2
;  1045          1037  2     ! Deallocate memory
;  1046          1038  2     !
;  1047          1039  3     WHILE ( .TMP_AREA_CNT NEQ 0 )
;  1048          1040  2     DO
;  1049          1041  3         BEGIN
;  1050          1042  3         TMP_AREA_CNT = .TMP_AREA_CNT - 1;
;  1051          1043  3         LIB$FREE_VM(    TMP_AREA_SIZE [ .TMP_AREA_CNT ],
;  1052          1044  3                         TMP_AREA_ADDR [ .TMP_AREA_CNT ] )
;  1053          1045  2         END;
;  1054          1046  2
;  1055          1047  2     RETURN
;  1056          1048  2
;  1057          1049  1     END;


                                0000'  CF   D5 00000 CONV$$FREE_TEMP_VM::
```

```
                                                        TSTL    TMP_AREA_CNT                            ; 1039
                                                1C  13  BEQL    1$                                      ;
                                        0000'   CF  D7  DECL    TMP_AREA_CNT                            ; 1042
                                    50  0000'   CF  D0  MOVL    TMP_AREA_CNT, R0                        ; 1044
                                        0000'CF40  DF  PUSHAL   TMP_AREA_ADDR[R0]                       ;
                                        0000'CF40  DF  PUSHAL   TMP_AREA_SIZE[R0]                       ; 1043
                            00000000G   00      02  FB  CALLS   #2, LIB$FREE_VM                        ; 1044
                                                DE  11  BRB     CONV$$FREE_TEMP_VM                      ;
                                                05  1$: RSB                                            ; 1049

; Routine Size:  35 bytes,    Routine Base:  _CONV$CODE + 03C2
```

```
;  1059          1050  1 %SBTTL  'EXCEPTION'
;  1060          1051  1 GLOBAL ROUTINE  CONV$$EXCEPTION ( CODE : BLOCK [ 4,BYTE ] ) =
;  1061          1052  1 !++
;  1062          1053  1 !
;  1063          1054  1 ! Functional Description:
;  1064          1055  1 !
;  1065          1056  1 !       Logs an exception record and if an exception file was specified
;  1066          1057  1 !       writes the record into it
;  1067          1058  1 !
;  1068          1059  1 ! Calling Sequence:
;  1069          1060  1 !
;  1070          1061  1 !       CONV$$EXCEPTION( code )
;  1071          1062  1 !
;  1072          1063  1 ! Input Parameters:
;  1073          1064  1 !
;  1074          1065  1 !       code    - Convert exception code
;  1075          1066  1 !
;  1076          1067  1 ! Implicit Inputs:
;  1077          1068  1 !       none
;  1078          1069  1 !
;  1079          1070  1 ! Output Parameters:
;  1080          1071  1 !       none
;  1081          1072  1 !
;  1082          1073  1 ! Implicit Outputs:
;  1083          1074  1 !       none
;  1084          1075  1 !
;  1085          1076  1 ! Routine Value:
;  1086          1077  1 !
;  1087          1078  1 !       CONV$_SUCCESS or
;  1088          1079  1 !       CONV$_FATALEXC  - if fatal exception
;  1089          1080  1 !
;  1090          1081  1 ! Routines Called:
;  1091          1082  1 !
;  1092          1083  1 !       CONV$$RMS_READ_ERROR
;  1093          1084  1 !       $PUTMSG
;  1094          1085  1 !       $PUT
;  1095          1086  1 !
;  1096          1087  1 ! Side Effects:
;  1097          1088  1 !       none
;  1098          1089  1 !
;  1099          1090  1 !--
;  1100          1091  1
;  1101          1092  2     BEGIN
;  1102          1093  2
;  1103          1094  2     OWN
;  1104          1095  2         MESSAGE_VECTOR  : VECTOR [ 2,LONG ] INITIAL ( 1,0 );
;  1105          1096  2
;  1106          1097  2     ! If this is a rms error the see if it is one of the recoverable ones
;  1107          1098  2     ! If it is not then call rms_read_error with the output rab which
;  1108          1099  2     ! will contain the error code in the sts field
;  1109          1100  2     !
;  1110          1101  2     IF .CODE [ STS$V_FAC_NO ] EQLU 1            ! RMS facility code is 1
;  1111          1102  2     THEN
;  1112          1103  2         SELECTONEU .CODE OF
;  1113          1104  2         SET
;  1114          1105  2             [ RMS$_DUP ] : CODE = CONV$_DUP;
;  1115          1106  2
```

```
;  1116          1107  2              [ RMS$_SEQ ] : CODE = CONV$_SEQ;
;  1117          1108  2
;  1118          1109  2              [ RMS$_KEY ] : CODE = CONV$_KEY;
;  1119          1110  2
;  1120          1111  2              [ RMS$_RSZ ] : CODE = CONV$_RSZ;
;  1121          1112  2
;  1122          1113  2              [ OTHERWISE ]: CONV$$RMS_READ_ERROR ( CONV$AB_OUT_RAB );
;  1123          1114  2
;  1124          1115  2          TES;
;  1125          1116  2
;  1126          1117  2      ! If we are to signal errors then output an exception message
;  1127          1118  2      !
;  1128          1119  2      IF .CONV$AB_FLAGS [ CONV$V_SIGNAL ]
;  1129          1120  2      THEN
;  1130          1121  3          BEGIN
;  1131          1122  3
;  1132          1123  3          MESSAGE_VECTOR [ 1 ] = .CODE;
;  1133          1124  3
;  1134          1125  4          $PUTMSG( MSGVEC=MESSAGE_VECTOR )
;  1135          1126  4
;  1136          1127  2          END;
;  1137          1128  2
;  1138          1129  2      ! If there is an exception file put the record into it
;  1139          1130  2      !
;  1140          1131  2      IF .CONV$GL_EXC
;  1141          1132  2      THEN
;  1142          1133  3          BEGIN
;  1143          1134  3
;  1144          1135  3          ! Stuff the size of the record which caused the exception
;  1145          1136  3          !
;  1146          1137  3          CONV$AB_EXC_RAB [ RAB$W_RSZ ] = .CONV$GW_IN_REC_SIZ;
;  1147          1138  3
;  1148          1139  3          ! Write the exception record
;  1149          1140  3          !
;  1150          1141  4          $PUT ( RAB=CONV$AB_EXC_RAB )
;  1151          1142  4
;  1152          1143  2          END;
;  1153          1144  2
;  1154          1145  2      ! Count the exception
;  1155          1146  2      !
;  1156          1147  2      CONV$GL_EXCEPT_COUNT = .CONV$GL_EXCEPT_COUNT + 1;
;  1157          1148  2
;  1158          1149  2      ! If /EXIT was specified exit with fatal excpetion
;  1159          1150  2      ! else return normal
;  1160          1151  2      !
;  1161          1152  2      IF .CONV$GL_EXIT
;  1162          1153  2      THEN
;  1163          1154  2          RETURN CONV$_FATALEXC
;  1164          1155  2      ELSE
;  1165          1156  2          RETURN CONV$_SUCCESS;
;  1166          1157  2
;  1167          1158  1      END;


                                                                .PSECT  _CONV$OWN,NOEXE,  PIC,2
```

```
                        00000000  00000001  00218 MESSAGE_VECTOR:
                                                  .LONG    1, 0                                          ;

                                                  .EXTRN   SYS$PUTMSG, SYS$PUT

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

                                 0000 00000       .ENTRY   CONV$$EXCEPTION, Save nothing                 ; 1051
   01      06   AC          0C     00  ED 00002   CMPZV    #0, #12, CODE+2, #1                           ; 1101
                                   59  12 00008   BNEQ     5$                                            ;
                   50         04   AC  D0 0000A   MOVL     CODE, R0                                      ; 1103
         000184EC  8F              50  D1 0000E   CMPL     R0, #99564                                    ; 1105
                                   0A  12 00015   BNEQ     1$                                            ;
               04  AC 00000000G    8F  D0 00017   MOVL     #CONV$_DUP, CODE                              ;
                                   42  11 0001F   BRB      5$                                            ;
         000186AC  8F              50  D1 00021 1$: CMPL   R0, #100012                                   ; 1107
                                   0A  12 00028   BNEQ     2$                                            ;
               04  AC 00000000G    8F  D0 0002A   MOVL     #CONV$_SEQ, CODE                              ;
                                   2F  11 00032   BRB      5$                                            ;
         00018594  8F              50  D1 00034 2$: CMPL   R0, #99732                                    ; 1109
                                   0A  12 0003B   BNEQ     3$                                            ;
               04  AC 00000000G    8F  D0 0003D   MOVL     #CONV$_KEY, CODE                              ;
                                   1C  11 00045   BRB      5$                                            ;
         000186A4  8F              50  D1 00047 3$: CMPL   R0, #100004                                   ; 1111
                                   0A  12 0004E   BNEQ     4$                                            ;
               04  AC 00000000G    8F  D0 00050   MOVL     #CONV$_RSZ, CODE                              ;
                                   09  11 00058   BRB      5$                                            ;
                         0000G     CF  9F 0005A 4$: PUSHAB CONV$AB_OUT_RAB                               ; 1113
             0000G CF              01  FB 0005E   CALLS    #1, CONV$$RMS_READ_ERROR                      ;
                   15    0000G     CF  E9 00063 5$: BLBC   CONV$AB_FLAGS, 6$                             ; 1119
             0000' CF         04   AC  D0 00068   MOVL     CODE, MESSAGE_VECTOR+4                        ; 1123
                                   7E  7C 0006E   CLRQ     -(SP)                                         ; 1125
                                   7E  D4 00070   CLRL     -(SP)                                         ;
                         0000'     CF  9F 00072   PUSHAB   MESSAGE_VECTOR                                ;
         00000000G 00              04  FB 00076   CALLS    #4, SYS$PUTMSG                                ;
                   12    0000G     CF  E9 0007D 6$: BLBC   CONV$GL_EXC, 7$                               ; 1131
             0000G CF    0000G     CF  B0 00082   MOVW     CONV$GW_IN_REC_SIZ, CONV$AB_EXC_RAB+34        ; 1137
                         0000G     CF  9F 00089   PUSHAB   CONV$AB_EXC_RAB                               ; 1141
         00000000G 00              01  FB 0008D   CALLS    #1, SYS$PUT                                   ;
                         0000G     CF  D6 00094 7$: INCL   CONV$GL_EXCEPT_COUNT                          ; 1147
                   08    0000G     CF  E9 00098   BLBC     CONV$GL_EXIT, 8$                              ; 1152
                   50 00000000G    8F  D0 0009D   MOVL     #CONV$_FATALEXC, R0                           ; 1156
                                   04     000A4   RET                                                    ;
                   50              01  D0 000A5 8$: MOVL   #1, R0                                        ;
                                   04     000A8   RET                                                    ; 1158

; Routine Size:  169 bytes,    Routine Base:  _CONV$CODE + 03E5
```

```
;  1169          1159  1 %SBTTL  'END_OF_FILE'
;  1170          1160  1 GLOBAL ROUTINE  CONV$$END_OF_FILE : NOVALUE =
;  1171          1161  1 !++
;  1172          1162  1 !
;  1173          1163  1 ! Functional Description:
;  1174          1164  1 !
;  1175          1165  1 !       Closes the current input file
;  1176          1166  1 !
;  1177          1167  1 ! Calling Sequence:
;  1178          1168  1 !
;  1179          1169  1 !       CONV$$END_OF_FILE
;  1180          1170  1 !
;  1181          1171  1 ! Input Parameters:
;  1182          1172  1 !       none
;  1183          1173  1 !
;  1184          1174  1 ! Implicit Inputs:
;  1185          1175  1 !       none
;  1186          1176  1 !
;  1187          1177  1 ! Output Parameters:
;  1188          1178  1 !       none
;  1189          1179  1 !
;  1190          1180  1 ! Implicit Outputs:
;  1191          1181  1 !       none
;  1192          1182  1 !
;  1193          1183  1 ! Routine Value:
;  1194          1184  1 !       none
;  1195          1185  1 !
;  1196          1186  1 ! Routines Called:
;  1197          1187  1 !
;  1198          1188  1 !       $DISCONNECT
;  1199          1189  1 !       $CLOSE
;  1200          1190  1 !       $ERASE
;  1201          1191  1 !
;  1202          1192  1 ! Side Effects:
;  1203          1193  1 !       none
;  1204          1194  1 !
;  1205          1195  1 !--
;  1206          1196  1
;  1207          1197  2     BEGIN
;  1208          1198  2
;  1209          1199  2     ! Disconnect the input RAB (the FAB closed depends on which was open)
;  1210          1200  2     !
;  1211          1201  2     $DISCONNECT ( RAB=CONV$AB_IN_RAB );
;  1212          1202  2
;  1213          1203  2     ! If the real input file was open close it
;  1214          1204  2     !
;  1215          1205  2     IF .CONV$AB_FLAGS [ CONV$V_IN ]
;  1216          1206  2     THEN
;  1217          1207  3         BEGIN
;  1218          1208  3         $CLOSE( FAB=CONV$AB_IN_FAB );
;  1219          1209  3         CONV$AB_FLAGS [ CONV$V_IN ] = _CLEAR
;  1220          1210  2         END;
;  1221          1211  2
;  1222          1212  2     ! If the RFA file was used, either for the primary sort or secondary sorts
;  1223          1213  2     ! then close it.
;  1224          1214  2     !
;  1225          1215  2     IF .CONV$AB_FLAGS [ CONV$V_RFA ] OR .CONV$AB_FLAGS [ CONV$V_SOR ]
```

```
;  1226          1216  2     THEN
;  1227          1217  3         BEGIN
;  1228          1218  3         $DISCONNECT( RAB=CONV$AB_RFA_RAB );
;  1229          1219  3         $CLOSE( FAB=CONV$AB_RFA_FAB );
;  1230          1220  3         $ERASE( FAB=CONV$AB_RFA_FAB );
;  1231          1221  3         CONV$AB_FLAGS [ CONV$V_RFA ] = _CLEAR
;  1232          1222  2         END;
;  1233          1223  2
;  1234          1224  2     ! Calculate Totals
;  1235          1225  2     !
;  1236          1226  2     CONV$GL_VALID_COUNT = .CONV$GL_RECORD_COUNT - .CONV$GL_EXCEPT_COUNT;
;  1237          1227  2
;  1238          1228  2     RETURN
;  1239          1229  2
;  1240          1230  1     END;


                                                        .EXTRN  SYS$DISCONNECT, SYS$CLOSE
                                                        .EXTRN  SYS$ERASE

                                        001C 00000      .ENTRY  CONV$$END_OF_FILE, Save R2,R3,R4                ; 1160
                    54 00000000G    00   9E 00002      MOVAB   SYS$CLOSE, R4                                   ;
                    53 00000000G    00   9E 00009      MOVAB   SYS$DISCONNECT, R3                              ;
                    52      0000G   CF   9E 00010      MOVAB   CONV$AB_FLAGS+2, R2                             ;
                            0000G   CF   9F 00015      PUSHAB  CONV$AB_IN_RAB                                  ; 1201
                    63              01   FB 00019      CALLS   #1, SYS$DISCONNECT                              ;
                    0A              62   E9 0001C      BLBC    CONV$AB_FLAGS+2, 1$                             ; 1205
                            0000G   CF   9F 0001F      PUSHAB  CONV$AB_IN_FAB                                  ; 1208
                    64              01   FB 00023      CALLS   #1, SYS$CLOSE                                   ;
                    62              01   8A 00026      BICB2   #1, CONV$AB_FLAGS+2                             ; 1209
                04  62              04   E0 00029 1$:  BBS     #4, CONV$AB_FLAGS+2, 2$                         ; 1215
                1C  62              03   E1 0002D      BBC     #3, CONV$AB_FLAGS+2, 3$                         ;
                            0000G   CF   9F 00031 2$:  PUSHAB  CONV$AB_RFA_RAB                                 ; 1218
                    63              01   FB 00035      CALLS   #1, SYS$DISCONNECT                              ;
                            0000G   CF   9F 00038      PUSHAB  CONV$AB_RFA_FAB                                 ; 1219
                    64              01   FB 0003C      CALLS   #1, SYS$CLOSE                                   ;
                            0000G   CF   9F 0003F      PUSHAB  CONV$AB_RFA_FAB                                 ; 1220
                 00000000G  00           01   FB 00043      CALLS   #1, SYS$ERASE                           ;
                    62              10   8A 0004A      BICB2   #16, CONV$AB_FLAGS+2                            ; 1221
        0000G   CF      0000G   CF      0000G   CF   C3 0004D 3$:  SUBL3   CONV$GL_EXCEPT_COUNT, -             ; 1226
                                                                           CONV$GL_RECORD_COUNT, CONV$GL_VALID_COUNT
                                        04 00057        RET                                                     ; 1230

; Routine Size:  88 bytes,    Routine Base:  _CONV$CODE + 048E

; 1241          1231  1
```

```
;  1243          1232  1 %SBTTL  'MAP_FTN_CCL'
;  1244          1233  1 ROUTINE CONV$$MAP_FTN_CCL =
;  1245          1234  1 !++
;  1246          1235  1 !
;  1247          1236  1 ! Functional Description:
;  1248          1237  1 !
;  1249          1238  1 !       Does conversion from/to FTN files based on the CONV$V_MAPFTN bits.
;  1250          1239  1 !       Output for FTN --> STM files is done from this routine, since
;  1251          1240  1 !       we need to know 1) what the old record ended with, and 2) what
;  1252          1241  1 !       the new record begins with before we can $PUT a record.
;  1253          1242  1 !
;  1254          1243  1 ! Calling Sequence:
;  1255          1244  1 !
;  1256          1245  1 !       CONV$$MAP_FTN_CCL()
;  1257          1246  1 !
;  1258          1247  1 ! Input Parameters:
;  1259          1248  1 !
;  1260          1249  1 !       None.
;  1261          1250  1 !
;  1262          1251  1 ! Implicit Inputs:
;  1263          1252  1 !
;  1264          1253  1 !       CONV$AB_FLAGS
;  1265          1254  1 !       CONV$AB_IN_RAB
;  1266          1255  1 !       CONV$AB_OUT_RAB
;  1267          1256  1 !       CONV$GL_STM_BUF
;  1268          1257  1 !       CONV$GL_STM_REC_LEN
;  1269          1258  1 !
;  1270          1259  1 ! Output Parameters:
;  1271          1260  1 !
;  1272          1261  1 !       none
;  1273          1262  1 !
;  1274          1263  1 ! Implicit Outputs:
;  1275          1264  1 !
;  1276          1265  1 !       none
;  1277          1266  1 !
;  1278          1267  1 ! Routine Value:
;  1279          1268  1 !
;  1280          1269  1 !       Success, unless PUT_RECORD complains
;  1281          1270  1 !
;  1282          1271  1 ! Routines Called:
;  1283          1272  1 !
;  1284          1273  1 !       PUT_RECORD
;  1285          1274  1 !
;  1286          1275  1 ! Side Effects:
;  1287          1276  1 !
;  1288          1277  1 !       The STM buffer may be $PUT to the output file.  The new
;  1289          1278  1 !       input record is moved to the STM buffer.
;  1290          1279  1 !
;  1291          1280  1 !--
;  1292          1281  2     BEGIN
;  1293          1282  2     LOCAL
;  1294          1283  2         STATUS;
;  1295          1284  2
;  1296          1285  2     BIND
;  1297          1286  2         CCLBYTE_IN   = .CONV$AB_IN_RAB  [ RAB$L_UBF ] : VECTOR [,BYTE],
;  1298          1287  2         CCLBYTE_OUT  = .CONV$AB_OUT_RAB [ RAB$L_RBF ] : BYTE,
;  1299          1288  2         PRINTCCC_IN  = .CONV$AB_IN_RAB  [ RAB$L_RHB ] : VECTOR [,WORD],
```

```
;  1300     1289  2          PRINTCCL_OUT = .CONV$AB_OUT_RAB [ RAB$L_RHB ] : WORD,
;  1301     1290  2          INPUT_BUFFER_PTR = .CONV$AB_IN_RAB  [ RAB$L_UBF ] : VECTOR [,BYTE];
;  1302     1291  2
;  1303     1292  2      !
;  1304     1293  2      ! See what the intention is for conversion.  Low bit set means
;  1305     1294  2      !   either FTN --> PRN or PRN --> FTN
;  1306     1295  2      !
;  1307     1296  2      IF .CONV$AB_FLAGS [ CONV$V_MAPFTN ]
;  1308     1297  2      THEN
;  1309     1298  3          BEGIN
;  1310     1299  3          IF .CONV$AB_FLAGS [ CONV$V_MAPFTN ] EQL CONV$C_FTNPRN
;  1311     1300  3          THEN
;  1312     1301  3              !
;  1313     1302  3              ! Convert from FORTRAN to printer format
;  1314     1303  3              !
;  1315     1304  3              SELECTONE .CCLBYTE_IN[0] OF SET
;  1316     1305  3
;  1317     1306  3                  [%C'1']   :     PRINTCCL_OUT = %x'8D8C';
;  1318     1307  3
;  1319     1308  3                  [%C'0']   :     PRINTCCL_OUT = %X'8D02';
;  1320     1309  3
;  1321     1310  3                  [%C'+']   :     PRINTCCL_OUT = %X'8D00';
;  1322     1311  3
;  1323     1312  3                  [%C'$']   :     PRINTCCL_OUT = %X'0001';
;  1324     1313  3
;  1325     1314  3                  [%X'00']  :     PRINTCCL_OUT = %X'0000';
;  1326     1315  3
;  1327     1316  3                  [OTHERWISE] :   PRINTCCL_OUT = %X'8D01';
;  1328     1317  3
;  1329     1318  3              TES
;  1330     1319  3
;  1331     1320  3          ELSE
;  1332     1321  3              !
;  1333     1322  3              ! Convert from printer format to FORTRAN
;  1334     1323  3              !
;  1335     1324  3              SELECTONE .PRINTCCL_IN[0] OF SET
;  1336     1325  3
;  1337     1326  3                  [%X'8D8C']  :   CCLBYTE_OUT = %C'1';
;  1338     1327  3
;  1339     1328  3                  [%X'8D02']  :   CCLBYTE_OUT = %C'0';
;  1340     1329  3
;  1341     1330  3                  [%X'8D00']  :   CCLBYTE_OUT = %C'+';
;  1342     1331  3
;  1343     1332  3                  [%X'0001']  :   CCLBYTE_OUT = %C'$';
;  1344     1333  3
;  1345     1334  3                  [%X'0000']  :   CCLBYTE_OUT = %X'00';
;  1346     1335  3
;  1347     1336  3                  [OTHERWISE] :   CCLBYTE_OUT = %C' ';
;  1348     1337  3
;  1349     1338  3              TES;
;  1350     1339  3          END
;  1351     1340  3
;  1352     1341  2      ELSE
;  1353     1342  2          !
;  1354     1343  2          ! Convert from FTN to STM -- visual equivalence...
;  1355     1344  2          !
;  1356     1345  3          BEGIN
```

```
;  1357          1346  3          LOCAL
;  1358          1347  3              STREAM_BUFFER_PTR : REF VECTOR [,BYTE],
;  1359          1348  3              NEW_RECORD_POINTER,
;  1360          1349  3              PUT,
;  1361          1350  3              NEW_REC_SIZ;
;  1362          1351  3
;  1363          1352  3          STREAM_BUFFER_PTR = .CONV$GL_STM_BUF + . CONV$GL_STM_REC_LEN;
;  1364          1353  3
;  1365          1354  3          SELECTONE .CCLBYTE_IN [0] OF SET
;  1366          1355  3
;  1367          1356  4             [%C'1']  : ( IF .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
;  1368          1357  4                          THEN
;  1369          1358  5                              BEGIN
;  1370          1359  5                              STREAM_BUFFER_PTR [0] = %X'0D';
;  1371          1360  5                              STREAM_BUFFER_PTR = .STREAM_BUFFER_PTR + 1;
;  1372          1361  5                              CONV$GL_STM_REC_LEN = .CONV$GL_STM_REC_LEN + 1;
;  1373          1362  4                              END;
;  1374          1363  4                          PUT =  CLEAR;
;  1375          1364  4                          NEW_RECORD_POINTER = INPUT_BUFFER_PTR ;
;  1376          1365  4                          INPUT_BUFFER_PTR [0]  = %X'0C';
;  1377          1366  4                          REC_ADJUST = 0;
;  1378          1367  4
;  1379          1368  3                          CONV$AB_FLAGS [ CONV$V_LAST_CR ] = _SET; );
;  1380          1369  3
;  1381          1370  4               [%C'0']   : ( IF  .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
;  1382          1371  4                          THEN
;  1383          1372  5                              BEGIN
;  1384          1373  5                              PUT =  SET;
;  1385          1374  5                              NEW_RECORD_POINTER = INPUT_BUFFER_PTR ;
;  1386          1375  5                              INPUT_BUFFER_PTR [0] = %X'0A';
;  1387          1376  5                              REC_ADJUST = 0;
;  1388          1377  5                              END
;  1389          1378  4                          ELSE
;  1390          1379  5                              BEGIN
;  1391          1380  5                              PUT =  CLEAR;
;  1392          1381  5                              NEW_RECORD_POINTER = INPUT_BUFFER_PTR - 1;
;  1393          1382  5                              INPUT_BUFFER_PTR [-1] = %X'0A';
;  1394          1383  5                              INPUT_BUFFER_PTR [0]  = %X'0A';
;  1395          1384  5                              REC_ADJUST = 1;
;  1396          1385  4                              END;
;  1397          1386  4
;  1398          1387  3                          CONV$AB_FLAGS [ CONV$V_LAST_CR ] = _SET; );
;  1399          1388  3
;  1400          1389  4                [%C'+']  : ( IF .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
;  1401          1390  4                          THEN
;  1402          1391  5                              BEGIN
;  1403          1392  5                              STREAM_BUFFER_PTR [0] = %X'0D';
;  1404          1393  5                              STREAM_BUFFER_PTR = .STREAM_BUFFER_PTR + 1;
;  1405          1394  5                              CONV$GL_STM_REC_LEN = .CONV$GL_STM_REC_LEN + 1;
;  1406          1395  4                              END;
;  1407          1396  4                          PUT =  CLEAR;
;  1408          1397  4                          NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
;  1409          1398  4                          REC_ADJUST = - 1;
;  1410          1399  3                          CONV$AB_FLAGS [ CONV$V_LAST_CR ] = _SET; );
;  1411          1400  3
;  1412          1401  4                [%C'$']  : ( IF .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
;  1413          1402  4                          THEN
```

```
;  1414     1403  5                        BEGIN
;  1415     1404  5                        PUT =  SET;
;  1416     1405  5                        NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
;  1417     1406  5                        REC_ADJUST = -1;
;  1418     1407  5                        END
;  1419     1408  4                    ELSE
;  1420     1409  5                        BEGIN
;  1421     1410  5                        PUT =  CLEAR;
;  1422     1411  5                        IF .CONV$AB_FLAGS [ CONV$V_FIRST_REC ]
;  1423     1412  5                        THEN
;  1424     1413  6                            BEGIN
;  1425     1414  6                            NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
;  1426     1415  6                            REC_ADJUST = -1;
;  1427     1416  6                            END
;  1428     1417  5                        ELSE
;  1429     1418  6                            BEGIN
;  1430     1419  6                            NEW_RECORD_POINTER = INPUT_BUFFER_PTR ;
;  1431     1420  6                            INPUT_BUFFER_PTR [0] = %X'0A';
;  1432     1421  6                            REC_ADJUST = 0;
;  1433     1422  5                            END;
;  1434     1423  4                        END;
;  1435     1424  4
;  1436     1425  3                    CONV$AB_FLAGS [ CONV$V_LAST_CR  ] = _CLEAR );
;  1437     1426  3
;  1438     1427  4            [%X'00'] : ( PUT =  CLEAR;
;  1439     1428  4                        IF .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
;  1440     1429  4                        THEN
;  1441     1430  5                            BEGIN
;  1442     1431  5                            STREAM_BUFFER_PTR [0] = %X'0D';
;  1443     1432  5                            STREAM_BUFFER_PTR = .STREAM_BUFFER_PTR + 1;
;  1444     1433  5                            CONV$GL_STM_REC_LEN = .CONV$GL_STM_REC_LEN + 1;
;  1445     1434  4                            END;
;  1446     1435  4                        NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
;  1447     1436  4                        REC_ADJUST = -1;
;  1448     1437  3                        CONV$AB_FLAGS [ CONV$V_LAST_CR ] = _CLEAR );
;  1449     1438  3
;  1450     1439  4          [OTHERWISE] : ( IF NOT .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
;  1451     1440  4                        THEN
;  1452     1441  5                            BEGIN
;  1453     1442  5                            PUT =  CLEAR;
;  1454     1443  5                            IF .CONV$AB_FLAGS [ CONV$V_FIRST_REC ]
;  1455     1444  5                            THEN
;  1456     1445  6                                BEGIN
;  1457     1446  6                                NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
;  1458     1447  6                                REC_ADJUST = -1;
;  1459     1448  6                                END
;  1460     1449  5                            ELSE
;  1461     1450  6                                BEGIN
;  1462     1451  6                                NEW_RECORD_POINTER = INPUT_BUFFER_PTR ;
;  1463     1452  6                                INPUT_BUFFER_PTR [0] = %X'0A';
;  1464     1453  6                                REC_ADJUST = 0;
;  1465     1454  5                                END;
;  1466     1455  5                            END
;  1467     1456  4                        ELSE
;  1468     1457  5                            BEGIN
;  1469     1458  5                            PUT =  SET;
;  1470     1459  5                            NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
```

```
;  1471          1460  5                        REC_ADJUST = -1;
;  1472          1461  4                        END;
;  1473          1462  3                    CONV$AB_FLAGS [ CONV$V_LAST_CR ] = _SET; );
;  1474          1463  3            TES;
;  1475          1464  3
;  1476          1465  3            !
;  1477          1466  3            ! If we can $PUT the STM buffer, then do it.  Adjust
;  1478          1467  3            !       buffer values appropriately to reflect what
;  1479          1468  3            !       happens here.
;  1480          1469  3
;  1481          1470  3            NEW_REC_SIZ = .CONV$GW_IN_REC_SIZ + .REC_ADJUST;
;  1482          1471  3
;  1483          1472  3            IF .PUT  OR
;  1484          1473  4                 ((.CONV$GL_STM_REC_LEN + .NEW_REC_SIZ)  GTRU  STM_BUF_SIZ )
;  1485          1474  3            THEN
;  1486          1475  4                BEGIN
;  1487          1476  4                CONV$GW_OUT_REC_SIZ = .CONV$GL_STM_REC_LEN ;
;  1488          1477  4                CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_STM_BUF;
;  1489          1478  5                IF NOT ( STATUS = CONV$$PUT_RECORD () )
;  1490          1479  4                THEN
;  1491          1480  4                    RETURN .STATUS;
;  1492          1481  4                !
;  1493          1482  4                ! Reset STM buffer pointer to beginning of buffer;
;  1494          1483  4                !   zero current STM record length.
;  1495          1484  4                !
;  1496          1485  4                STREAM_BUFFER_PTR = .CONV$GL_STM_BUF;
;  1497          1486  4                CONV$GL_STM_REC_LEN = 0;
;  1498          1487  3                END;
;  1499          1488  3
;  1500          1489  3            CH$MOVE ( .NEW_REC_SIZ, .NEW_RECORD_POINTER, .STREAM_BUFFER_PTR);
;  1501          1490  3            CONV$GL_STM_REC_LEN = .CONV$GL_STM_REC_LEN + .NEW_REC_SIZ ;
;  1502          1491  3            CONV$GW_OUT_REC_SIZ = .CONV$GL_STM_REC_LEN;
;  1503          1492  3            CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_STM_BUF;
;  1504          1493  2            END;
;  1505          1494  2
;  1506          1495  2        RETURN CONV$_SUCCESS;
;  1507          1496  2
;  1508          1497  1        END;


                                                OFFC 00000 CONV$$MAP_FTN_CCL:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1233
                                          5B    0000G  CF  9E 00002          MOVAB   CONV$GL_STM_BUF, R11          ;
                                          5A    0000G  CF  9E 00007          MOVAB   CONV$AB_OUT_RAB+40, R10        ;
                                          59    0000'  CF  9E 0000C          MOVAB   REC_ADJUST, R9                ;
                                          58    0000G  CF  9E 00011          MOVAB   CONV$GL_STM_REC_LEN, R8       ;
                                          57    0000G  CF  9E 00016          MOVAB   CONV$AB_FLAGS+2, R7           ;
                                          50    0000G  CF  D0 0001B          MOVL    CONV$AB_IN_RAB+36, R0         ; 1286
                                          53           6A  D0 00020          MOVL    CONV$AB_OUT_RAB+40, R3        ; 1287
                                          52      04   AA  D0 00023          MOVL    CONV$AB_OUT_RAB+44, R2        ; 1289
                                                       67  95 00027          TSTB    CONV$AB_FLAGS+2               ; 1296
                                                       03  19 00029          BLSS    1$                            ;
                                                     0088  31 0002B          BRW     14$                           ;
        01              67              02             07  ED 0002E 1$:      CMPZV   #7, #2, CONV$AB_FLAGS+2, #1   ; 1299
```

```
                              40  12 00033          BNEQ    7$                                      ;
                   51         60  9A 00035          MOVZBL  (R0), R1                                ; 1304
                   31         51  91 00038          CMPB    R1, #49                                 ; 1306
                              07  12 0003B          BNEQ    2$                                      ;
                   62   8D8C  8F  B0 0003D          MOVW    #-29300, (R2)                           ;
                              6F  11 00042          BRB     13$                                     ;
                   30         51  91 00044  2$:     CMPB    R1, #48                                 ; 1308
                              07  12 00047          BNEQ    3$                                      ;
                   62   8D02  8F  B0 00049          MOVW    #-29438, (R2)                           ;
                              63  11 0004E          BRB     13$                                     ;
                   2B         51  91 00050  3$:     CMPB    R1, #43                                 ; 1310
                              07  12 00053          BNEQ    4$                                      ;
                   62   8D00  8F  B0 00055          MOVW    #-29440, (R2)                           ;
                              57  11 0005A          BRB     13$                                     ;
                   24         51  91 0005C  4$:     CMPB    R1, #36                                 ; 1312
                              05  12 0005F          BNEQ    5$                                      ;
                   62         01  B0 00061          MOVW    #1, (R2)                                ;
                              4D  11 00064          BRB     13$                                     ;
                              51  D5 00066  5$:     TSTL    R1                                      ; 1314
                              04  12 00068          BNEQ    6$                                      ;
                              62  B4 0006A          CLRW    (R2)                                    ;
                              45  11 0006C          BRB     13$                                     ;
                   62   8D01  8F  B0 0006E  6$:     MOVW    #-29439, (R2)                           ; 1316
                              3E  11 00073          BRB     13$                                     ; 1304
                   51  0000G  DF  3C 00075  7$:     MOVZWL  @CONV$AB_IN_RAB+44, R1                  ; 1324
             8D8C  8F         51  B1 0007A          CMPW    R1, #36236                              ; 1326
                              05  12 0007F          BNEQ    8$                                      ;
                   63         31  90 00081          MOVB    #49, (R3)                               ;
                              2D  11 00084          BRB     13$                                     ;
             8D02  8F         51  B1 00086  8$:     CMPW    R1, #36098                              ; 1328
                              05  12 0008B          BNEQ    9$                                      ;
                   63         30  90 0008D          MOVB    #48, (R3)                               ;
                              21  11 00090          BRB     13$                                     ;
             8D00  8F         51  B1 00092  9$:     CMPW    R1, #36096                              ; 1330
                              05  12 00097          BNEQ    10$                                     ;
                   63         2B  90 00099          MOVB    #43, (R3)                               ;
                              15  11 0009C          BRB     13$                                     ;
                   01         51  B1 0009E  10$:    CMPW    R1, #1                                  ; 1332
                              05  12 000A1          BNEQ    11$                                     ;
                   63         24  90 000A3          MOVB    #36, (R3)                               ;
                              0B  11 000A6          BRB     13$                                     ;
                              51  D5 000A8  11$:    TSTL    R1                                      ; 1334
                              04  12 000AA          BNEQ    12$                                     ;
                              63  94 000AC          CLRB    (R3)                                    ;
                              03  11 000AE          BRB     13$                                     ;
                   63         20  90 000B0  12$:    MOVB    #32, (R3)                               ; 1336
                        00F1  31 000B3  13$:        BRW     34$                                     ; 1296
       54          6B         68  C1 000B6  14$:    ADDL3   CONV$GL_STM_REC_LEN, CONV$GL_STM_BUF, - ; 1352
                                                            STREAM_BUFFER_PTR
                   51         60  9A 000BA          MOVZBL  (R0), R1                                ; 1354
                   31         51  91 000BD          CMPB    R1, #49                                 ; 1356
                              15  12 000C0          BNEQ    16$                                     ;
       05    01    A7         01  E1 000C2          BBC     #1, CONV$AB_FLAGS+3, 15$                ;
                   84         0D  90 000C7          MOVB    #13, (STREAM_BUFFER_PTR)+               ; 1359
                              68  D6 000CA          INCL    CONV$GL_STM_REC_LEN                     ; 1361
                              52  D4 000CC  15$:    CLRL    PUT                                     ; 1363
                   53         50  D0 000CE          MOVL    R0, NEW_RECORD_POINTER                  ; 1364
```

```
                                60              0C  90 000D1          MOVB    #12, (R0)                               ; 1365
                              0082              31  31 000D4          BRW     27$                                     ; 1366
                                30              51  91 000D7 16$:     CMPB    R1, #48                                 ; 1370
                                                1B  12 000DA          BNEQ    18$                                     ;
                     05     01  A7              01  E1 000DC          BBC     #1, CONV$AB_FLAGS+3, 17$                ;
                                52              01  D0 000E1          MOVL    #1, PUT                                 ; 1373
                                                6D  11 000E4          BRB     26$                                     ; 1374
                                                52  D4 000E6 17$:     CLRL    PUT                                     ; 1380
                                53          FF  A0  9E 000E8          MOVAB   -1(R0), NEW_RECORD_POINTER              ; 1381
                             FF A0      0A0A    8F  B0 000EC          MOVW    #2570, -1(R0)                           ; 1382
                                69              01  D0 000F2          MOVL    #1, REC_ADJUST                          ; 1384
                                                70  11 000F5          BRB     30$                                     ; 1387
                                2B              51  91 000F7 18$:     CMPB    R1, #43                                 ; 1389
                                                0E  12 000FA          BNEQ    20$                                     ;
                     05     01  A7              01  E1 000FC          BBC     #1, CONV$AB_FLAGS+3, 19$                ;
                                84              0D  90 00101          MOVB    #13, (STREAM_BUFFER_PTR)+               ; 1392
                                                68  D6 00104          INCL    CONV$GL_STM_REC_LEN                     ; 1394
                                                52  D4 00106 19$:     CLRL    PUT                                     ; 1396
                                                56  11 00108          BRB     29$                                     ; 1397
                                24              51  91 0010A 20$:     CMPB    R1, #36                                 ; 1401
                                                1B  12 0010D          BNEQ    22$                                     ;
                     05     01  A7              01  E1 0010F          BBC     #1, CONV$AB_FLAGS+3, 21$                ;
                                52              01  D0 00114          MOVL    #1, PUT                                 ; 1404
                                                21  11 00117          BRB     23$                                     ; 1405
                                                52  D4 00119 21$:     CLRL    PUT                                     ; 1410
                     1A     01  A7              02  E0 0011B          BBS     #2, CONV$AB_FLAGS+3, 23$                ; 1411
                                53              50  D0 00120          MOVL    R0, NEW_RECORD_POINTER                  ; 1419
                                60              0A  90 00123          MOVB    #10, (R0)                               ; 1420
                                                69  D4 00126          CLRL    REC_ADJUST                              ; 1421
                                                17  11 00128          BRB     24$                                     ; 1425
                                                51  D5 0012A 22$:     TSTL    R1                                      ; 1427
                                                19  12 0012C          BNEQ    25$                                     ;
                                                52  D4 0012E          CLRL    PUT                                     ;
                     05     01  A7              01  E1 00130          BBC     #1, CONV$AB_FLAGS+3, 23$                ; 1428
                                84              0D  90 00135          MOVB    #13, (STREAM_BUFFER_PTR)+               ; 1431
                                                68  D6 00138          INCL    CONV$GL_STM_REC_LEN                     ; 1433
                                53          01  A0  9E 0013A 23$:     MOVAB   1(R0), NEW_RECORD_POINTER               ; 1435
                                69              01  CE 0013E          MNEGL   #1, REC_ADJUST                          ; 1436
                                01  A7              02  8A 00141 24$:     BICB2   #2, CONV$AB_FLAGS+3                     ; 1437
                                                24  11 00145          BRB     31$                                     ;
                     11     01  A7              01  E0 00147 25$:     BBS     #1, CONV$AB_FLAGS+3, 28$                ; 1439
                                                52  D4 0014C          CLRL    PUT                                     ; 1442
                     0D     01  A7              02  E0 0014E          BBS     #2, CONV$AB_FLAGS+3, 29$                ; 1443
                                53              50  D0 00153 26$:     MOVL    R0, NEW_RECORD_POINTER                  ; 1451
                                60              0A  90 00156          MOVB    #10, (R0)                               ; 1452
                                                69  D4 00159 27$:     CLRL    REC_ADJUST                              ; 1453
                                                0A  11 0015B          BRB     30$                                     ; 1439
                                52              01  D0 0015D 28$:     MOVL    #1, PUT                                 ; 1458
                                53          01  A0  9E 00160 29$:     MOVAB   1(R0), NEW_RECORD_POINTER               ; 1459
                                69              01  CE 00164          MNEGL   #1, REC_ADJUST                          ; 1460
                                01  A7              02  88 00167 30$:     BISB2   #2, CONV$AB_FLAGS+3                     ; 1462
                                56          0000G   CF  32 0016B 31$:     CVTWL   CONV$GW_IN_REC_SIZ, NEW_REC_SIZ         ; 1470
                                56              69  C0 00170          ADDL2   REC_ADJUST, NEW_REC_SIZ                 ;
                                0D              52  E8 00173          BLBS    PUT, 32$                                ; 1472
                     50                         68  C1 00176          ADDL3   NEW_REC_SIZ, CONV$GL_STM_REC_LEN, R0    ; 1473
                        00007FFE                8F  50  D1 0017A          CMPL    R0, #32766                              ;
                                                15  1B 00181          BLEQU   33$                                     ;
```

```
                          0000G  CF          68  B0 00183 32$:    MOVW    CONV$GL_STM_REC_LEN, CONV$GW_OUT_REC_SIZ     ; 1476
                                 6A          6B  D0 00188         MOVL    CONV$GL_STM_BUF, CONV$AB_OUT_RAB+40          ; 1477
                          0000V  CF          00  FB 0018B         CALLS   #0, CONV$$PUT_RECORD                         ; 1478
                                 17          50  E9 00190         BLBC    STATUS, 35$                                  ;
                                 54          6B  D0 00193         MOVL    CONV$GL_STM_BUF, STREAM_BUFFER_PTR           ; 1485
                                             68  D4 00196         CLRL    CONV$GL_STM_REC_LEN                          ; 1486
                            64               63  56  28 00198 33$:    MOVC3   NEW_REC_SIZ, (NEW_RECORD_POINTER), -         ; 1489
                                                                             (STREAM_BUFFER_PTR)                          ;
                                 68          56  C0 0019C         ADDL2   NEW_REC_SIZ, CONV$GL_STM_REC_LEN             ; 1490
                          0000G  CF          68  B0 0019F         MOVW    CONV$GL_STM_REC_LEN, CONV$GW_OUT_REC_SIZ     ; 1491
                                 6A          6B  D0 001A4         MOVL    CONV$GL_STM_BUF, CONV$AB_OUT_RAB+40          ; 1492
                                 50          01  D0 001A7 34$:    MOVL    #1, R0                                       ; 1495
                                             04  001AA 35$:    RET                                                  ; 1497

; Routine Size:  427 bytes,    Routine Base:  _CONV$CODE + 04E6
```

```
;  1510          1498  1 %SBTTL  'PUT_RECORD'
;  1511          1499  1 GLOBAL ROUTINE CONV$$PUT_RECORD  =
;  1512          1500  1 !++
;  1513          1501  1 !
;  1514          1502  1 ! Functional Description:
;  1515          1503  1 !
;  1516          1504  1 !       $PUT record to output file.
;  1517          1505  1 !
;  1518          1506  1 ! Calling Sequence:
;  1519          1507  1 !
;  1520          1508  1 !       CONV$$PUT_RECORD ()
;  1521          1509  1 !
;  1522          1510  1 ! Input Parameters:
;  1523          1511  1 !
;  1524          1512  1 !       None.
;  1525          1513  1 !
;  1526          1514  1 ! Implicit Inputs:
;  1527          1515  1 !
;  1528          1516  1 !       CONV$AB_OUT_RAB
;  1529          1517  1 !
;  1530          1518  1 ! Output Parameters:
;  1531          1519  1 !
;  1532          1520  1 !       none
;  1533          1521  1 !
;  1534          1522  1 ! Implicit Outputs:
;  1535          1523  1 !
;  1536          1524  1 !       none
;  1537          1525  1 !
;  1538          1526  1 ! Routine Value:
;  1539          1527  1 !
;  1540          1528  1 !       See CONV$$EXCEPTION
;  1541          1529  1 !
;  1542          1530  1 ! Routines Called:
;  1543          1531  1 !
;  1544          1532  1 !       CONV$$EXCEPTION
;  1545          1533  1 !
;  1546          1534  1 ! Side Effects:
;  1547          1535  1 !
;  1548          1536  1 !       The record is $PUT to the output file
;  1549          1537  1 !
;  1550          1538  1 !--
;  1551          1539  2     BEGIN
;  1552          1540  2     LOCAL
;  1553          1541  2         STATUS;
;  1554          1542  2
;  1555          1543  2     ! Set the size of the record to OUT_REC_SIZE which is set by GET_RECORD
;  1556          1544  2     !
;  1557          1545  2     CONV$AB_OUT_RAB [ RAB$W_RSZ ] = .CONV$GW_OUT_REC_SIZ;
;  1558          1546  2
;  1559          1547  2     ! Write the record, if there was an error call the exception handler
;  1560          1548  2     !
;  1561          1549  3     IF NOT ( STATUS = $PUT ( RAB=CONV$AB_OUT_RAB ) )
;  1562          1550  2     THEN
;  1563          1551  2         ! If the exception handler returned with an error
;  1564          1552  2         ! then it was fatal
;  1565          1553  2         !
;  1566          1554  2         STATUS = CONV$$EXCEPTION( .STATUS );
```

```
; 1567          1555  2
; 1568          1556  2     RETURN .STATUS;
; 1569          1557  2
; 1570          1558  1     END;


                                                0000 00000          .ENTRY  CONV$$PUT_RECORD, Save nothing          ; 1499
                                    0000G   CF  B0 00002          MOVW    CONV$GW_OUT_REC_SIZ, CONV$AB_OUT_RAB+34 ; 1545
                                    0000G   CF  9F 00009          PUSHAB  CONV$AB_OUT_RAB                         ; 1549
                            00000000G   00      01 FB 0000D       CALLS   #1, SYS$PUT                             ;
                                        07      50 E8 00014       BLBS    STATUS, 1$                              ;
                                                50 DD 00017       PUSHL   STATUS                                  ; 1554
                                FD36    CF      01 FB 00019       CALLS   #1, CONV$$EXCEPTION                     ;
                                                04 0001E 1$:      RET                                             ; 1558

; Routine Size:  31 bytes,    Routine Base:  _CONV$CODE + 0691

; 1571          1559  1
; 1572          1560  0 END ELUDOM


                                                                  .EXTRN  LIB$STOP

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;  _CONV$OWN                     544  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _CONV$CODE                   1712  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        38         0        1000        00:01.9
;  _$255$DUA28:[CONV.SRC]CONVERT.L32;1    165        27        16          17        00:00.2


;                                  COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CONVMAIN/OBJ=OBJ$:CONVMAIN MSRC$:CONVMAIN/UPDATE=(ENH$:CONVMAIN)
```

```
; Size:          1712 code + 544 data bytes
; Run Time:        00:34.8
; Elapsed Time:    01:50.6
; Lines/CPU Min:    2693
; Lexemes/CPU-Min: 14608
; Memory Used:  217 pages
; Compilation Complete
```

0066 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RECLDCL
LIS

RECLREC
LIS

RECLCTRL
LIS

RECLRMSIO
LIS

CONVMSG
LIS

CONVMAIN
LIS

CONVSORT
LIS

CONVVEC
LIS