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

FILEID**CONVMAIN

N 2

```
  CCCCCCCC    OOOOOO    NN      NN  VV      VV  MM      MM    AAAAAA      IIIIII    NN      NN
  CCCCCCCC    OOOOOO    NN      NN  VV      VV  MM      MM    AAAAAA      IIIIII    NN      NN
CC          OO      OO  NN      NN  VV      VV  MMMM  MMMM  AA      AA      II      NN      NN
CC          OO      OO  NN      NN  VV      VV  MMMM  MMMM  AA      AA      II      NN      NN
CC          OO      OO  NNNN    NN  VV      VV  MM  MM  MM  AA      AA      II      NNNN    NN
CC          OO      OO  NNNN    NN  VV      VV  MM  MM  MM  AA      AA      II      NNNN    NN
CC          OO      OO  NN  NN  NN  VV      VV  MM      MM  AA      AA      II      NN  NN  NN
CC          OO      OO  NN  NN  NN  VV      VV  MM      MM  AA      AA      II      NN  NN  NN
CC          OO      OO  NN    NNNN  VV      VV  MM      MM  AAAAAAAAAA      II      NN    NNNN
CC          OO      OO  NN    NNNN  VV      VV  MM      MM  AAAAAAAAAA      II      NN    NNNN
CC          OO      OO  NN      NN    VV  VV    MM      MM  AA      AA      II      NN      NN    ..
CC          OO      OO  NN      NN    VV  VV    MM      MM  AA      AA      II      NN      NN    ..
  CCCCCCCC    OOOOOO    NN      NN      VV      MM      MM  AA      AA    IIIIII    NN      NN    ..
  CCCCCCCC    OOOOOO    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
```

```
0 %TITLE  'VAX-11 CONVERT'
0 MODULE  CONV$MAIN         ( IDENT='V04-000',
0                           OPTLEVEL=3
0                           ) =
0
1 BEGIN
1
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
```

```
1 !++
1 !
1 ! Facility:      VAX-11 CONVERT
1 !
1 ! Abstract:      CONVERT main routines
1 !
1 ! Contents:
1 !                CONVERT
1 !                GET_RECORD
1 !                GET_VM
1 !                FREE_VM
1 !                GET_TEMP_VM
1 !                FREE_TEMP_VM
1 !                EXCEPTION
1 !                END_OF_FILE
1 !
1 ! Environment:
1 !
1 !                VAX/VMS Operating System
1 !
1 !--
1
1 !
1 ! Author:        Keith B Thompson        Creation date:  June-1980
1 !
1 !
1 ! Modified by:
1 !
1 !       V03-010 JWT0194         Jim Teague              31-Aug-1984
1 !               Fix problem with missing data on UDF input.
1 !
1 !       V03-009 JWT0185         Jim Teague              28-Jun-1984
1 !               Make FTN carriage control improvements:
1 !
1 !                       - NEVER just toss away FTN ccl bytes
1 !                       - provide visually equivalent conversion
1 !                               between FTN <---> PRN files.
1 !                       - provide a visually equivalent
1 !                               FTN ---> STM conversion.
1 !
1 !       V03-008 KBT0470         Keith B. Thompson       21-Jan-1983
1 !               Add RSZ error code to exception
1 !
1 !       V03-007 KBT0466         Keith B. Thompson       21-Jan-1983
1 !               Fix conv$$free_temp_vm
1 !
1 !       V03-006 KBT0434         Keith B. Thompson       16-Dec-1982
1 !               Fix some of the carriage control checking
1 !
1 !       V03-005 KBT0387         Keith B. Thompson       27-Oct-1982
1 !               Have get_vm and get_temp_vm zero the space allocated
1 !
1 !       V03-004 KBT0381         Keith B. Thompson       25-Oct-1982
1 !               Make RMS$_SEQ error non-fatal
1 !
1 !       V03-003 KBT0376         Keith B. Thompson       22-Oct-1982
1 !               Replace conv$v_dcl with conv$v_signal and make vm buffers
```

```
1 !                     bigger
1 !
1 !     V03-002 KBT0346         Keith B. Thompson       4-Oct-1982
1 !             Use new linkage definitions
1 !
1 !     V03-001 KBT0016         Keith Thompson          18-Mar-1982
1 !             Add control-y processing in get_record and fix call to
1 !             conv$$rms_read_error in conv$$exception
1 !
1 !****
```

```
PSECT
        OWN     = _CONV$OWN         (PIC),
        GLOBAL  = _CONV$GLOBAL      (PIC),
        PLIT    = _CONV$PLIT        (SHARE,PIC),
        CODE    = _CONV$CODE        (SHARE,PIC);

LIBRARY 'SYS$LIBRARY:LIB.L32';
LIBRARY 'SRC$:CONVERT';

DEFINE_ERROR_CODES;

EXTERNAL ROUTINE
        LIB$GET_VM              : ADDRESSING_MODE(GENERAL),
        LIB$FREE_VM             : ADDRESSING_MODE(GENERAL),
        CONV$$SORT_PRIMARY      : CL$SORT_PRIMARY,
        CONV$$FAST_LOAD         : CL$FAST_LOAD,
        CONV$$RMS_READ_ERROR    : NOVALUE;

FORWARD ROUTINE
        CONV$$GET_RECORD        : CL$GET_RECORD,
        CONV$$GET_VM            : CL$GET_VM,
        CONV$$FREE_TEMP_VM      : CL$FREE_TEMP_VM       NOVALUE,
        CONV$$EXCEPTION,
        CONV$$END_OF_FILE       : NOVALUE,
        CONV$$PUT_RECORD,
        CONV$$MAP_FTN_CCL;

EXTERNAL
        CONV$GL_APPEND          : LONG,
        CONV$GL_CREATE          : LONG,
        CONV$GL_EXC             : LONG,
        CONV$GL_EXIT            : LONG,
        CONV$GL_FAST            : LONG,
        CONV$GL_FIX             : LONG,
        CONV$GL_MERGE           : LONG,
        CONV$GL_PAD             : LONG,
        CONV$GL_PAD_CHAR        : LONG,
        CONV$GL_SHARE           : LONG,
        CONV$GL_SORT            : LONG,
        CONV$GL_TRUNCATE        : LONG,

        CONV$AB_FLAGS           : BLOCK [ ,BYTE ],
        CONV$GL_VALID_COUNT,
        CONV$GL_EXCEPT_COUNT,
        CONV$GB_CURRENT_FILE    : BYTE,
        CONV$GW_OUT_MRS         : WORD,
        CONV$GW_UDF_MRS         : WORD,
        CONV$GW_IN_REC_SIZ      : SIGNED WORD,
        CONV$GW_OUT_REC_SIZ     : SIGNED WORD,
        CONV$GL_RECORD_PTR,
        CONV$GL_RECORD_COUNT,
        CONV$GW_MAX_REC_SIZ     : WORD,
        CONV$GL_REC_BUF_PTR,
        CONV$GL_VFC_BUF_PTR,
        CONV$GL_RFA_BUFFER,
```

```
1          CONV$AB_RFA_FAB          : $FAB_DECL,
1          CONV$AB_RFA_RAB          : $RAB_DECL,
1          CONV$AB_IN_FAB           : $FAB_DECL,
1          CONV$AB_IN_RAB           : $RAB_DECL,
1          CONV$AB_IN_NAM           : $NAM_DECL,
1          CONV$AB_OUT_FAB          : $FAB_DECL,
1          CONV$AB_OUT_RAB          : $RAB_DECL,
1          CONV$AB_OUT_NAM          : $NAM_DECL,
1          CONV$AB_EXC_RAB          : $RAB_DECL,
1          CONV$GL_STM_BUF,
1          CONV$GL_STM_REC_LEN;
1
1   OWN
1          DYN_AREA_CNT,
1          DYN_AREA_ADDR   : VECTOR [ 32,LONG ],
1          DYN_AREA_SIZE   : VECTOR [ 32,LONG ],
1
1          TMP_AREA_CNT,
1          TMP_AREA_ADDR   : VECTOR [ 32,LONG ],
1          TMP_AREA_SIZE   : VECTOR [ 32,LONG ],
1
1          BYTE_COUNT      : SIGNED LONG,
1          N_RFAS,
1          REC_ADJUST      : SIGNED LONG;          ! Record size adjustment
```

```
1 %SBTTL  'CONVERT'
1 GLOBAL ROUTINE  CONV$$CONVERT =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Sets up the record buffer pointers and any format conversion,
1 !       sorts the input file if needed and loads the output file.
1 !
1 !       How the buffers work:
1 !
1 !       The main record buffer:
1 !
1 !             ---------------------------------------//-----------------
1 !             |                   |                                   |
1 !             ---------------------------------------//-----------------
1 !             ^                   ^
1 !             |                   |
1 !          VFC_BUF_PTR        REC_BUF_PTR
1 !
1 !
1 !       No /FIXED_CONTROL or neither or both files VFC
1 !
1 !       Neither or both files FTN
1 !
1 ! INPUT RAB:   RHB                UBF
1 !             |                   |
1 !             ---------------------------------------//-----------------
1 !             |                   |                                   |
1 !             |                   |                                   |
1 !             ---------------------------------------//-----------------
1 !             |                   |
1 ! OUTPUT RAB:  RHB                RBF
1 !
1 !
1 !       Input file FTN
1 !
1 ! INPUT RAB:   RHB                UBF
1 !             |                   |
1 !             ---------------------------------------//-----------------
1 !             |                   |         |                         |
1 !             |                   |         |                         |
1 !             ---------------------------------------//-----------------
1 !             |                             |
1 ! OUTPUT RAB:  RHB                          RBF
1 !
1 !
1 !       Output file FTN
1 !
1 ! INPUT RAB:   RHB                UBF
1 !             |                   |
1 !             ---------------------------------------//-----------------
1 !             |         |         |                                   |
1 !             |         |    20   |                                   |
1 !             ---------------------------------------//-----------------
1 !             |         |
```

```
! OUTPUT RAB:  RHB      RBF
!
!
!       Ouput file PRN input file not VFC
!
! INPUT RAB:   RHB                UBF
!               :                  :
!              ---------------------------------------//-----------------
!              :                  :                                     :
!              :   01  :  8D      :                                     :
!              ---------------------------------------//-----------------
!              :                  :
! OUTPUT RAB:  RHB                RBF
!
!
!       With /FIXED_CONTROL
!
!       Input file VFC, output file not:
!
! INPUT RAB:   RHB                UBF
!               :                  :
!              ---------------------------------------//-----------------
!              :                  :                                     :
!              :                  :                                     :
!              ---------------------------------------//-----------------
!              :
! OUTPUT RAB:  RBF
!
!
!       Output file VFC, input file not:
!
! INPUT RAB:   UBF
!               :
!              ---------------------------------------//-----------------
!              :                  :                                     :
!              :                  :                                     :
!              ---------------------------------------//-----------------
!              :                  :
! OUTPUT RAB:  RHB                RBF
!
!
! Calling Sequence:
!
!       CONV$$CONVERT()
!
! Input Parameters:
!       none
!
! Implicit Inputs:
!       none
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
```

```
1 ! Routine Value:
1 !
1 !       CONV$_SUCCESS or error code
1 !
1 ! Routines Called:
1 !
1 !       CONV$$SORT_PRIMARY
1 !       CONV$$FAST_LOAD
1 !       CONV$$GET_RECORD
1 !       $PUT
1 !       CONV$$EXCEPTION
1 !       CONV$$END_OF_FILE
1 !       CONV$$FREE_TEMP_VM
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2   BEGIN
2
2   DEFINE_KEY_DESC_GLOBAL;
2
2 !*******************
2
2
2   CONV$AB_FLAGS [ CONV$V_MAPFTN ] = 0;
2
2 !*******************
2
2     ! Setup Buffer Pointers etc.
2     !
2     ! If the FIX Option is on and only one of the files
2     ! has VFC Format will pointers be moved
2     !
2     IF .CONV$GL_FIX AND
3       (( .CONV$AB_IN_FAB [ FAB$B_RFM ] EQL FAB$C_VFC ) XOR
3        ( .CONV$AB_OUT_FAB [ FAB$B_RFM ] EQL FAB$C_VFC ))
2     THEN
2
2         IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQL FAB$C_VFC
2         THEN
3             BEGIN
3
3             ! The Input File is VFC
3             !
3             REC_ADJUST = .CONV$AB_IN_FAB [ FAB$B_FSZ ];
3             CONV$AB_IN_RAB [ RAB$L_RHB ] = .CONV$GL_VFC_BUF_PTR;
3             CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$GL_REC_BUF_PTR;
3             CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_VFC_BUF_PTR
3             END
2         ELSE
3             BEGIN
3
3             ! The Output File is VFC
3             !
3             REC_ADJUST = - .CONV$AB_OUT_FAB [ FAB$B_FSZ ];
```

```
                    CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$GL_VFC_BUF_PTR;
                    CONV$AB_OUT_RAB [ RAB$L_RHB ] = .CONV$GL_VFC_BUF_PTR;
                    CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_REC_BUF_PTR
                    END
            ELSE
                BEGIN

                ! Either they Both are or are not VFC files
                !
                REC_ADJUST = 0;
                CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$GL_REC_BUF_PTR;
                CONV$AB_IN_RAB [ RAB$L_RHB ] = .CONV$GL_VFC_BUF_PTR;
                CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_REC_BUF_PTR;
                CONV$AB_OUT_RAB [ RAB$L_RHB ] = .CONV$GL_VFC_BUF_PTR;

                ! Take care of some special cases
                !
                ! If the input is FTN and the output isn't...
                !
                IF .CONV$AB_IN_FAB [ FAB$V_FTN ] AND NOT .CONV$AB_OUT_FAB [ FAB$V_FTN ]
                THEN
                    !
                    ! Output is PRN -- do a carriage control approximation
                    !
                    IF .CONV$AB_OUT_FAB [ FAB$V_PRN ]
                    THEN
                        BEGIN
                        CONV$AB_FLAGS [ CONV$V_MAPFTN ] = CONV$C_FTNPRN;                        ! (1)
                        CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$AB_OUT_RAB [ RAB$L_RBF ] + 1;
                        REC_ADJUST = -1;
                        END
                    ELSE
                        !
                        ! Output is CR:STM -- do visual approximation
                        !
                        IF .CONV$AB_OUT_FAB [ FAB$B_RFM ] EQL FAB$C_STM
                        THEN
                            BEGIN
                            CONV$AB_FLAGS [ CONV$V_MAPFTN ] = CONV$C_FTNSTM ;   ! (2)
                            CONV$GL_STM_BUF = CONV$$GET_VM ( STM_BUF_SIZ );
                            CONV$GL_STM_REC_LEN = 0;
                            CONV$AB_FLAGS [ CONV$V_FIRST_REC ] = 1;
                            CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$AB_IN_RAB [ RAB$L_UBF ] + 1;
                            REC_ADJUST = 0;
                            END;

                ! If the output file is FTN and the input file is not ...
                !
                IF .CONV$AB_OUT_FAB [ FAB$V_FTN ] AND NOT .CONV$AB_IN_FAB [ FAB$V_FTN ]
                THEN
                    !
                    ! If PRN --> FTN, then do an approximation...
                    !
                    IF .CONV$AB_IN_FAB [ FAB$V_PRN ]
                    THEN
                        BEGIN
                        CONV$AB_FLAGS [ CONV$V_MAPFTN ] = CONV$C_PRNFTN;                        ! (3)
```

```
                CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$AB_IN_RAB [ RAB$L_UBF ] + 1;
                REC_ADJUST = 1;
                END
            ELSE
                BEGIN
                LOCAL RECORD_BUFFER : REF VECTOR [ ,BYTE ];

                RECORD_BUFFER = .CONV$AB_IN_RAB [ RAB$L_UBF ];

                RECORD_BUFFER [ 0 ] = 20;                    ! 20 = ASCII space

                CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$AB_IN_RAB [ RAB$L_UBF ] + 1;
                REC_ADJUST = 1

                END;

        ! If the output file is PRN and the input is not VFC
        ! then put LF - CR control info into the VFC field of the output
        ! NOTE: VMS print files are "018D" in stead of "8A8D" which would
        !       be more reasonable?
        !
        IF .CONV$AB_OUT_FAB [ FAB$V_PRN ] AND
                        ( .CONV$AB_IN_FAB [ FAB$B_RFM ] NEQU FAB$C_VFC )
        THEN
            BEGIN

            LOCAL RECORD_BUFFER : REF VECTOR [ ,BYTE ];

            RECORD_BUFFER = .CONV$AB_OUT_RAB [ RAB$L_RHB ];

            RECORD_BUFFER [ 0 ] = %X'01';                ! One LF
            RECORD_BUFFER [ 1 ] = %X'8D'                 ! CR

            END;

        END;

    ! The exception record is always the record pointed to by the output rab

    CONV$AB_EXC_RAB [ RAB$L_RBF ] = .CONV$AB_OUT_RAB [ RAB$L_RBF ];

    ! Save the pointer to the output record
    !
    CONV$GL_RECORD_PTR = .CONV$AB_OUT_RAB [ RAB$L_RBF ];

    ! Initialize some Variables
    !
    CONV$AB_IN_RAB [ RAB$W_USZ ] = .CONV$GW_MAX_REC_SIZ;
    BYTE_COUNT = 0;

    ! If SORT is on then sort the records on the output primary key
    !
    IF .CONV$GL_SORT
    THEN
        RET_ON_ERROR( CONV$$SORT_PRIMARY() );

    BEGIN               ! STATUS local
```

```
3
3       LOCAL       STATUS  : LONG;
3
3       ! If FAST then call FAST_LOAD, otherwise do it the slow way
3       !
3       IF .CONV$GL_FAST
3       THEN
3           STATUS = CONV$$FAST_LOAD()
3       ELSE
4           BEGIN
4
4           ! Restore the output rab pointer (sort_primary destroys it)
4           !
4           CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_RECORD_PTR;
4
4           ! Enter Main Loop
4           !
4           ! Loop untill an error or end of file
4           !
5           WHILE ( STATUS = CONV$$GET_RECORD() )
4           DO
4               IF .CONV$AB_FLAGS [ CONV$V_MAPFTN ] NEQ CONV$C_FTNSTM
4               THEN
5                   IF NOT ( STATUS = CONV$$PUT_RECORD () )
4                   THEN
4                       EXITLOOP;
4
4           ! Finish off this file
4           !
4           CONV$$END_OF_FILE()
4
3           END;
3
3       ! Deallocate all of the temporary memory used by this run
3       !
3       CONV$$FREE_TEMP_VM();
3
3       ! If end of file thats normal
3       !
4       IF ( .STATUS EQL RMS$_EOF )
3       THEN
3           STATUS = CONV$_SUCCESS;
3
3       RETURN .STATUS
3
3       END                     ! STATUS local
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 TMP_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
```

```
%SBTTL  'GET_RECORD'
GLOBAL ROUTINE  CONV$$GET_RECORD : CL$GET_RECORD =
!++
!
! Functional Description:
!
!       Gets a record from the input file and processes it
!
! Calling Sequence:
!
!       CONV$$GET_RECORD()
!
! Input Parameters:
!       none
!
! Implicit Inputs:
!       none
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!
!       CONV$_SUCCESS or status returned by $GET
!
! Routines Called:
!
!       $READ
!       CONV$$RMS_READ_ERROR    - By RMS as an AST
!       $GET
!       CONV$$EXCEPTION
!
! Side Effects:
!       none
!
!--

    BEGIN

    LABEL
        GET_REC,
        EXC_REC;

    OWN
        RFA_IDX         : LONG;

    LOCAL
        STATUS          : LONG;

    ! RFA Vector declarations
    !
    BIND
        RFA0    = .CONV$GL_RFA_BUFFER : RFA0_VECTOR [ 256 ];
        RFA4    = .CONV$GL_RFA_BUFFER : RFA4_VECTOR [ 256 ];
```

```
	STATUS = _SET;

	! Loop while ok (we could fail from a fatal exception)
	!
	WHILE .STATUS
	DO
	    BEGIN		! Main While loop

GET_REC:
		BEGIN		! GET_REC Block

		! If Input Format is UDF then we do some strange things
		!
		IF .CONV$AB_IN_FAB [ FAB$B_RFM ] EQLU FAB$C_UDF
		THEN
		    BEGIN	! UDF Block

		    ! If the Byte Count from last time is Positive then we must move
		    ! the Chracters Left Over to the Front of the Buffer
		    !
		    IF .BYTE_COUNT GEQ .CONV$GW_UDF_MRS
		    THEN
			BEGIN

			! Set status here because we dont do a READ
			!
			STATUS = CONV$_SUCCESS;

			! Cut down BYTE_COUNT the size of the last record
			!
			BYTE_COUNT = .BYTE_COUNT - .CONV$GW_UDF_MRS;

			! Move the extra characters to the front of the buffer
			!
			CH$MOVE ( .BYTE_COUNT,
				.CONV$GL_REC_BUF_PTR + .CONV$GW_UDF_MRS,
				.CONV$GL_REC_BUF_PTR )
			END;

		    ! Read some Blocks to Get the Record
		    !
		    WHILE .BYTE_COUNT LSS .CONV$GW_UDF_MRS
		    DO
			BEGIN		! READ While loop

			! Point the RAB buffer to the position just after the last
			! character left over from last time
			!
			CONV$AB_IN_RAB [ RAB$L_UBF ] = .CONV$GL_REC_BUF_PTR + .BYTE_COUNT;

			IF ( NOT ( STATUS = $READ( RAB=CONV$AB_IN_RAB,
						ERR=CONV$$RMS_READ_ERROR ) ) )
			THEN

			    ! If Byte Count is > 0 then use that and give EOF
			    ! next time
```

```
                        !
                        IF .BYTE_COUNT GTR 0
                        THEN
                            BEGIN
                            CONV$GW_IN_REC_SIZ = .BYTE_COUNT;
                            BYTE_COUNT = 0;
                            STATUS = _SET;
                            LEAVE GET_REC
                            END
                        ELSE
                            RETURN .STATUS;

                    BYTE_COUNT = .BYTE_COUNT + .CONV$AB_IN_RAB [ RAB$W_RSZ ];

                    ! If we got a Short Record then we are also done
                    !
                    IF .CONV$AB_IN_RAB [ RAB$W_RSZ ] LSSU BLOCK_SIZE
                    THEN

                        ! Check to see if the short part put us over the limit
                        !
                        IF .BYTE_COUNT GTR .CONV$GW_UDF_MRS
                        THEN
                            EXITLOOP
                        ELSE
                            BEGIN
                            CONV$GW_IN_REC_SIZ = .BYTE_COUNT;
                            BYTE_COUNT = 0;
                            LEAVE GET_REC
                            END;

                    END;            ! READ While Loop

                ! Get Ready for Next Time
                !
                CONV$GW_IN_REC_SIZ = .CONV$GW_UDF_MRS

                END     ! UDF Block

            ! For NON UDF Files do a Regular Get
            !
            ELSE
                BEGIN   ! Regular GET Block

                ! If Reading a file by an RFA file first get the RFA
                !
                IF .CONV$AB_FLAGS [ CONV$V_RFA ]
                THEN
                    BEGIN

                    ! If we ran out of RFAs get some more
                    !
                    IF .N_RFAS EQL 0
                    THEN
                        IF ( NOT ( STATUS = $READ( RAB=CONV$AB_RFA_RAB,
                                                   ERR=CONV$$RMS_READ_ERROR ) ) )
                        THEN
```

```
                                        RETURN .STATUS
                                    ELSE
                                        BEGIN

                                        ! RFAs are six bytes long so we can tell how
                                        ! many there are
                                        !
                                        N_RFAS = .CONV$AB_RFA_RAB [ RAB$W_RSZ ] / 6;
                                        RFA_IDX = 0

                                        END;

                                ! Get an RFA
                                !
                                CONV$AB_IN_RAB [ RAB$L_RFA0 ] = .RFA0 [ .RFA_IDX ];
                                CONV$AB_IN_RAB [ RAB$W_RFA4 ] = .RFA4 [ .RFA_IDX ];

                                RFA_IDX = .RFA_IDX + 1;
                                N_RFAS = .N_RFAS - 1

                                END;

                        ! Finally get a record!  If the i/o was complete under
                        ! control-y just ignore it and try again
                        !
                        DO

                        ! rms_read_error will return if the error was end of file
                        !
                        ( IF ( NOT ( STATUS = $GET( RAB = CONV$AB_IN_RAB,
                                                    ERR = CONV$$RMS_READ_ERROR ) ) )
                        THEN
                            RETURN .STATUS )

                        WHILE .STATUS EQLU RMS$_CONTROLY;

                        ! Set the input record size
                        !
                        CONV$GW_IN_REC_SIZ = .CONV$AB_IN_RAB [ RAB$W_RSZ ]

                        END;    ! Regular GET Block

                    END;        ! GET_REC Block

                ! We have a record so count it
                !
                CONV$GL_RECORD_COUNT = .CONV$GL_RECORD_COUNT + 1;

                ! If we need to do FTN carriage control mappings,
                !       then go for it...
                !
                IF .CONV$AB_FLAGS [ CONV$V_MAPFTN ] NEQ 0
                THEN
                    IF NOT ( STATUS = CONV$$MAP_FTN_CCL () )
                    THEN
                        RETURN .STATUS;
```

```
                ! Make adjustments on the record size if necessary
                !
                CONV$GW_IN_REC_SIZ = .CONV$GW_IN_REC_SIZ + .REC_ADJUST;

        ! Check fo exception conditions ie. is the record to long to short ect...
        ! The exception record block has a series of test to check the valididy
        ! of the record if an exception record is found exit of the block will
        ! repeat the process of getting a record with out exiting.  If all checks
        ! suceed then we drop through an exit
        !
        EXC_REC:
                    BEGIN        ! EXC_REC Block

                    ! If record size is negative then the input record
                    ! is to short to fill the VFC portion of the output file
                    !
                    IF .CONV$GW_IN_REC_SIZ LSS 0
                    THEN
                        BEGIN
                        CONV$GW_IN_REC_SIZ = .CONV$GW_IN_REC_SIZ - .REC_ADJUST;
                        STATUS = CONV$$EXCEPTION( CONV$_VFC );
                        LEAVE EXC_REC
                        END;

                    ! If Output File MRS is 0 then we check for index key size if necc.
                    !
                    IF .CONV$GW_OUT_MRS EQL 0
                    THEN
                        CONV$GW_OUT_REC_SIZ = .CONV$GW_IN_REC_SIZ
                    ELSE

                        ! If the Record is Longer then MRS Check for Truncate or
                        ! Exception
                        !
                        IF .CONV$GW_IN_REC_SIZ GTR .CONV$GW_OUT_MRS
                        THEN

                            ! If Truncate then set the Record's size to MRS
                            !
                            IF .CONV$GL_TRUNCATE
                            THEN
                                CONV$GW_OUT_REC_SIZ = .CONV$GW_OUT_MRS
                            ELSE
                                BEGIN
                                STATUS = CONV$$EXCEPTION( CONV$_RTL );
                                LEAVE EXC_REC
                                END
                        ELSE

                            ! If the file is fixed an the record is short...
                            !
                            IF ( .CONV$AB_OUT_FAB [ FAB$B_RFM ] EQL FAB$C_FIX ) AND
                               ( .CONV$GW_IN_REC_SIZ LSS .CONV$GW_OUT_MRS )
                            THEN
```

```
4                               ! If PAD then pad the short record
4                               !
4                               IF .CONV$GL_PAD
4                               THEN
5                                   BEGIN
5                                   CH$FILL ( .CONV$GL_PAD_CHAR,
5                                             .CONV$GW_OUT_MRS - .CONV$GW_IN_REC_SIZ,
5                                             .CONV$GL_REC_BUF_PTR + .CONV$GW_IN_REC_SIZ );
5                                   CONV$GW_OUT_REC_SIZ = .CONV$GW_OUT_MRS
5                                   END
4                               ELSE
5                                   BEGIN
5                                   STATUS = CONV$$EXCEPTION( CONV$_RTS );
5                                   LEAVE EXC_REC
5                                   END
4                           ELSE
4                               CONV$GW_OUT_REC_SIZ = .CONV$GW_IN_REC_SIZ;
4
4                   ! If we made it this far then we have an ok record
4                   !
4                   RETURN .STATUS
4
3                   END;            ! EXC_REC Block
3
2               END;                ! Main While loop
2
2               RETURN .STATUS
2
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'   CF   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 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
```

```
%SBTTL  'GET_VM'
GLOBAL ROUTINE  CONV$$GET_VM ( BYTES ) : CL$GET_VM =
!++
!
! Functional Description:
!
!       Allocates virtual memory and records it so that it can
!       be returned by CONV$$FREE_VM
!
! Calling Sequence:
!
!       CONV$$GET_VM( bytes )
!
! Input Parameters:
!
!       bytes   - Number of bytes to allocate
!
! Implicit Inputs:
!       none
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!
!       R0 -    Address of start of memory allocated
!
! Routines called:
!
!       LIB$GET_VM
!
! Side Effects:
!       none
!
!--

    BEGIN

    LOCAL       POINTER;

    ! Allocate the memory and quit if errors
    !
    IF NOT LIB$GET_VM( BYTES,POINTER )
    THEN
        SIGNAL_STOP( CONV$_INSVIRMEM );

    ! Keep track of allocated memory so we can give it back latter
    !
    DYN_AREA_SIZE [ .DYN_AREA_CNT ] = .BYTES;
    DYN_AREA_ADDR [ .DYN_AREA_CNT ] = .POINTER;
    DYN_AREA_CNT = .DYN_AREA_CNT + 1;

    ! Zero the allocated space
    !
```

```
;  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  DD 00005          PUSHL   SP                                   ; 0869
                                     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$_INSVIRMEM                     ; 0871
                        00000000G  00     01  FB 0001A          CALLS   #1, LIB$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
```

```
1 %SBTTL  'FREE_VM'
1 GLOBAL ROUTINE CONV$$FREE_VM : CL$FREE_VM NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Returns virtual memory allocated by CONV$$GET_VM
1 !
1 ! Calling Sequence:
1 !
1 !       CONV$$FREE_VM()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !       none
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !       none
1 !
1 ! Routines called:
1 !
1 !       LIB$FREE_VM
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     ! Deallocate memory
2     !
3     WHILE ( .DYN_AREA_CNT NEQ 0 )
2     DO
3         BEGIN
3         DYN_AREA_CNT = .DYN_AREA_CNT - 1;
3         LIB$FREE_VM(     DYN_AREA_SIZE [ .DYN_AREA_CNT ],
3                          DYN_AREA_ADDR [ .DYN_AREA_CNT ] )
2         END;
2
2     RETURN
2
1     END;


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

```
                                                          TSTL    DYN_AREA_CNT                    ; 0926
                                 1C  13 00004            BEQL    1$                              ;
                        0000'    CF  D7 00006            DECL    DYN_AREA_CNT                    ; 0929
              50        0000'    CF  D0 0000A            MOVL    DYN_AREA_CNT, R0                ; 0931
                        0000'CF40    DF 0000F            PUSHAL  DYN_AREA_ADDR[R0]               ;
                        0000'CF40    DF 00014            PUSHAL  DYN_AREA_SIZE[R0]               ; 0930
      00000000G  00                  02 FB 00019         CALLS   #2, LIB$FREE_VM                 ; 0931
                                 DE  11 00020            BRB     CONV$$FREE_VM                   ;
                                     05 00022 1$:        RSB                                     ; 0936

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

```
%SBTTL  'GET_TEMP_VM'
GLOBAL ROUTINE  CONV$$GET_TEMP_VM ( BYTES ) : CL$GET_TEMP_VM =
!++
!
! Functional Description:
!
!       Allocates virtual memory and records it so that it can
!       be returned by CONV$$FREE_TEMP_VM
!
! Calling Sequence:
!
!       CONV$$GET_TEMP_VM( bytes )
!
! Input Parameters:
!
!       bytes   - Number of bytes to allocate
!
! Implicit Inputs:
!       none
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!
!       R0 -    Address of start of memory allocated
!
! Routines called:
!
!       LIB$GET_VM
!
! Side Effects:
!       none
!
!--

    BEGIN

    LOCAL       POINTER;

    ! Allocate the memory and quit if errors
    !
    IF NOT LIB$GET_VM( BYTES,POINTER )
    THEN
        SIGNAL_STOP( CONV$_INSVIRMEM );

    ! Keep track of allocated memory so we can give it back latter
    !
    TMP_AREA_SIZE [ .TMP_AREA_CNT ] = .BYTES;
    TMP_AREA_ADDR [ .TMP_AREA_CNT ] = .POINTER;
    TMP_AREA_CNT = .TMP_AREA_CNT + 1;

    ! Zero the allocated space
    !
```

```
; 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
```

```
1 %SBTTL  'FREE_TEMP_VM'
1 GLOBAL ROUTINE CONV$$FREE_TEMP_VM : CL$FREE_TEMP_VM NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Returns virtual memory allocated by CONV$$GET_TEMP_VM
1 !
1 ! Calling Sequence:
1 !
1 !       CONV$$FREE_TEMP_VM()
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !       none
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !       none
1 !
1 ! Routines called:
1 !
1 !       LIB$FREE_VM
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     ! Deallocate memory
2     !
3     WHILE ( .TMP_AREA_CNT NEQ 0 )
2     DO
3         BEGIN
3         TMP_AREA_CNT = .TMP_AREA_CNT - 1;
3         LIB$FREE_VM(    TMP_AREA_SIZE [ .TMP_AREA_CNT ],
3                         TMP_AREA_ADDR [ .TMP_AREA_CNT ] )
2         END;
2
2     RETURN
2
1     END;


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

```
                                                     TSTL    TMP_AREA_CNT                ; 1039
                         1C  13 00004                BEQL    1$                          ;
               0000'     CF  D7 00006                DECL    TMP_AREA_CNT                ; 1042
       50      0000'     CF  D0 0000A                MOVL    TMP_AREA_CNT, R0            ; 1044
               0000'CF40     DF 0000F                PUSHAL  TMP_AREA_ADDR[R0]           ;
               0000'CF40     DF 00014                PUSHAL  TMP_AREA_SIZE[R0]           ; 1043
00000000G  00                02  FB 00019            CALLS   #2, LIB$FREE_VM             ; 1044
                         DE  11 00020                BRB     CONV$$FREE_TEMP_VM          ;
                             05 00022 1$:            RSB                                 ; 1049

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

```
%SBTTL  'EXCEPTION'
GLOBAL ROUTINE  CONV$$EXCEPTION ( CODE : BLOCK [ 4,BYTE ] ) =
!++
!
! Functional Description:
!
!       Logs an exception record and if an exception file was specified
!       writes the record into it
!
! Calling Sequence:
!
!       CONV$$EXCEPTION( code )
!
! Input Parameters:
!
!       code    - Convert exception code
!
! Implicit Inputs:
!       none
!
! Output Parameters:
!       none
!
! Implicit Outputs:
!       none
!
! Routine Value:
!
!       CONV$_SUCCESS or
!       CONV$_FATALEXC  - if fatal exception
!
! Routines Called:
!
!       CONV$$RMS_READ_ERROR
!       $PUTMSG
!       $PUT
!
! Side Effects:
!       none
!
!--

    BEGIN

    OWN
        MESSAGE_VECTOR  : VECTOR [ 2,LONG ] INITIAL ( 1,0 );

    ! If this is a rms error the see if it is one of the recoverable ones
    ! If it is not then call rms_read_error with the output rab which
    ! will contain the error code in the sts field
    !
    IF .CODE [ STS$V_FAC_NO ] EQLU 1            ! RMS facility code is 1
    THEN
        SELECTONEU .CODE OF
        SET
            [ RMS$_DUP ] : CODE = CONV$_DUP;

```

```
                [ RMS$_SEQ ] : CODE = CONV$_SEQ;

                [ RMS$_KEY ] : CODE = CONV$_KEY;

                [ RMS$_RSZ ] : CODE = CONV$_RSZ;

                [ OTHERWISE ]: CONV$$RMS_READ_ERROR ( CONV$AB_OUT_RAB );

            TES;

        ! If we are to signal errors then output an exception message
        !
        IF .CONV$AB_FLAGS [ CONV$V_SIGNAL ]
        THEN
            BEGIN

            MESSAGE_VECTOR [ 1 ] = .CODE;

            $PUTMSG( MSGVEC=MESSAGE_VECTOR )

            END;

        ! If there is an exception file put the record into it
        !
        IF .CONV$GL_EXC
        THEN
            BEGIN

            ! Stuff the size of the record which caused the exception
            !
            CONV$AB_EXC_RAB [ RAB$W_RSZ ] = .CONV$GW_IN_REC_SIZ;

            ! Write the exception record
            !
            $PUT ( RAB=CONV$AB_EXC_RAB )

            END;

        ! Count the exception
        !
        CONV$GL_EXCEPT_COUNT = .CONV$GL_EXCEPT_COUNT + 1;

        ! If /EXIT was specified exit with fatal excpetion
        ! else return normal
        !
        IF .CONV$GL_EXIT
        THEN
            RETURN CONV$_FATALEXC
        ELSE
            RETURN CONV$_SUCCESS;

        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
```

```
1 %SBTTL 'END_OF_FILE'
1 GLOBAL ROUTINE CONV$$END_OF_FILE : NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Closes the current input file
1 !
1 ! Calling Sequence:
1 !
1 !       CONV$$END_OF_FILE
1 !
1 ! Input Parameters:
1 !       none
1 !
1 ! Implicit Inputs:
1 !       none
1 !
1 ! Output Parameters:
1 !       none
1 !
1 ! Implicit Outputs:
1 !       none
1 !
1 ! Routine Value:
1 !       none
1 !
1 ! Routines Called:
1 !
1 !       $DISCONNECT
1 !       $CLOSE
1 !       $ERASE
1 !
1 ! Side Effects:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     ! Disconnect the input RAB (the FAB closed depends on which was open)
2     !
2     $DISCONNECT ( RAB=CONV$AB_IN_RAB );
2
2     ! If the real input file was open close it
2     !
2     IF .CONV$AB_FLAGS [ CONV$V_IN ]
2     THEN
3         BEGIN
3         $CLOSE( FAB=CONV$AB_IN_FAB );
3         CONV$AB_FLAGS [ CONV$V_IN ] = _CLEAR
2         END;
2
2     ! If the RFA file was used, either for the primary sort or secondary sorts
2     ! then close it.
2     !
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
```

```
1 %SBTTL  'MAP FTN CCL'
1 ROUTINE CONV$$MAP_FTN_CCL =
1 !++
1 !
1 ! Functional Description:
1 !
1 !       Does conversion from/to FTN files based on the CONV$V_MAPFTN bits.
1 !       Output for FTN --> STM files is done from this routine, since
1 !       we need to know 1) what the old record ended with, and 2) what
1 !       the new record begins with before we can $PUT a record.
1 !
1 ! Calling Sequence:
1 !
1 !       CONV$$MAP_FTN_CCL()
1 !
1 ! Input Parameters:
1 !
1 !       None.
1 !
1 ! Implicit Inputs:
1 !
1 !       CONV$AB_FLAGS
1 !       CONV$AB_IN_RAB
1 !       CONV$AB_OUT_RAB
1 !       CONV$GL_STM_BUF
1 !       CONV$GL_STM_REC_LEN
1 !
1 ! Output Parameters:
1 !
1 !       none
1 !
1 ! Implicit Outputs:
1 !
1 !       none
1 !
1 ! Routine Value:
1 !
1 !       Success, unless PUT_RECORD complains
1 !
1 ! Routines Called:
1 !
1 !       PUT_RECORD
1 !
1 ! Side Effects:
1 !
1 !       The STM buffer may be $PUT to the output file.  The new
1 !       input record is moved to the STM buffer.
1 !
1 !--
2     BEGIN
2     LOCAL
2         STATUS;
2
2     BIND
2         CCLBYTE_IN   = .CONV$AB_IN_RAB  [ RAB$L_UBF ] : VECTOR [,BYTE],
2         CCLBYTE_OUT  = .CONV$AB_OUT_RAB [ RAB$L_RBF ] : BYTE,
2         PRINTCCC_IN  = .CONV$AB_IN_RAB  [ RAB$L_RHB ] : VECTOR [,WORD],
```

```
        PRINTCCL_OUT = .CONV$AB_OUT_RAB [ RAB$L_RHB ] : WORD,
        INPUT_BUFFER_PTR = .CONV$AB_IN_RAB  [ RAB$L_UBF ] : VECTOR [,BYTE];

    !
    ! See what the intention is for conversion.  Low bit set means
    !   either FTN --> PRN or PRN --> FTN
    !
    IF .CONV$AB_FLAGS [ CONV$V_MAPFTN ]
    THEN
        BEGIN
        IF .CONV$AB_FLAGS [ CONV$V_MAPFTN ] EQL CONV$C_FTNPRN
        THEN
            !
            ! Convert from FORTRAN to printer format
            !
            SELECTONE .CCLBYTE_IN[0] OF SET

                [%C'1']    :     PRINTCCL_OUT = %x'8D8C';

                [%C'0']    :     PRINTCCL_OUT = %X'8D02';

                [%C'+']    :     PRINTCCL_OUT = %X'8D00';

                [%C'$']    :     PRINTCCL_OUT = %X'0001';

                [%X'00']   :     PRINTCCL_OUT = %X'0000';

              [OTHERWISE] :      PRINTCCL_OUT = %X'8D01';

            TES

        ELSE
            !
            ! Convert from printer format to FORTRAN
            !
            SELECTONE .PRINTCCL_IN[0] OF SET

                [%X'8D8C']  :   CCLBYTE_OUT = %C'1';

                [%X'8D02']  :   CCLBYTE_OUT = %C'0';

                [%X'8D00']  :   CCLBYTE_OUT = %C'+';

                [%X'0001']  :   CCLBYTE_OUT = %C'$';

                [%X'0000']  :   CCLBYTE_OUT = %X'00';

                [OTHERWISE] :   CCLBYTE_OUT = %C' ';

            TES;
        END
    ELSE
        !
        ! Convert from FTN to STM -- visual equivalence...
        !
        BEGIN
```

```
LOCAL
    STREAM_BUFFER_PTR : REF VECTOR [,BYTE],
    NEW_RECORD_POINTER,
    PUT,
    NEW_REC_SIZ;

STREAM_BUFFER_PTR = .CONV$GL_STM_BUF + . CONV$GL_STM_REC_LEN;

SELECTONE .CCLBYTE_IN [0] OF SET

   [%C'1']  : ( IF .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
                THEN
                    BEGIN
                    STREAM_BUFFER_PTR [0] = %X'0D';
                    STREAM_BUFFER_PTR = .STREAM_BUFFER_PTR + 1;
                    CONV$GL_STM_REC_LEN = .CONV$GL_STM_REC_LEN + 1;
                    END;
                PUT = _CLEAR;
                NEW_RECORD_POINTER = INPUT_BUFFER_PTR ;
                INPUT_BUFFER_PTR [0]  = %X'0C';
                REC_ADJUST = 0;

                CONV$AB_FLAGS [ CONV$V_LAST_CR ] = _SET; );

  [%C'0']   : ( IF  .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
                THEN
                    BEGIN
                    PUT = _SET;
                    NEW_RECORD_POINTER = INPUT_BUFFER_PTR ;
                    INPUT_BUFFER_PTR [0] = %X'0A';
                    REC_ADJUST = 0;
                    END
                ELSE
                    BEGIN
                    PUT = _CLEAR;
                    NEW_RECORD_POINTER = INPUT_BUFFER_PTR - 1;
                    INPUT_BUFFER_PTR [-1] = %X'0A';
                    INPUT_BUFFER_PTR [0]  = %X'0A';
                    REC_ADJUST = 1;
                    END;

                CONV$AB_FLAGS [ CONV$V_LAST_CR ] = _SET; );

   [%C'+']  : ( IF .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
                THEN
                    BEGIN
                    STREAM_BUFFER_PTR [0] = %X'0D';
                    STREAM_BUFFER_PTR = .STREAM_BUFFER_PTR + 1;
                    CONV$GL_STM_REC_LEN = .CONV$GL_STM_REC_LEN + 1;
                    END;
                PUT = _CLEAR;
                NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
                REC_ADJUST = - 1;
                CONV$AB_FLAGS [ CONV$V_LAST_CR ] = _SET; );

   [%C'$']  : ( IF .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
                THEN
```

```
                          BEGIN
                          PUT = _SET;
                          NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
                          REC_ADJUST = -1;
                          END
                      ELSE
                          BEGIN
                          PUT = _CLEAR;
                          IF .CONV$AB_FLAGS [ CONV$V_FIRST_REC ]
                          THEN
                              BEGIN
                              NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
                              REC_ADJUST = -1;
                              END
                          ELSE
                              BEGIN
                              NEW_RECORD_POINTER = INPUT_BUFFER_PTR ;
                              INPUT_BUFFER_PTR [0] = %X'0A';
                              REC_ADJUST = 0;
                              END;
                          END;

                      CONV$AB_FLAGS [ CONV$V_LAST_CR  ] = _CLEAR );

       [%X'00'] : ( PUT = _CLEAR;
                      IF .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
                      THEN
                          BEGIN
                          STREAM_BUFFER_PTR [0] = %X'0D';
                          STREAM_BUFFER_PTR = .STREAM_BUFFER_PTR + 1;
                          CONV$GL_STM_REC_LEN = .CONV$GL_STM_REC_LEN + 1;
                          END;
                      NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
                      REC_ADJUST = -1;
                      CONV$AB_FLAGS [ CONV$V_LAST_CR ] = _CLEAR );

    [OTHERWISE] : ( IF NOT .CONV$AB_FLAGS [ CONV$V_LAST_CR ]
                      THEN
                          BEGIN
                          PUT = _CLEAR;
                          IF .CONV$AB_FLAGS [ CONV$V_FIRST_REC ]
                          THEN
                              BEGIN
                              NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
                              REC_ADJUST = -1;
                              END
                          ELSE
                              BEGIN
                              NEW_RECORD_POINTER = INPUT_BUFFER_PTR ;
                              INPUT_BUFFER_PTR [0] = %X'0A';
                              REC_ADJUST = 0;
                              END;
                          END
                      ELSE
                          BEGIN
                          PUT = _SET;
                          NEW_RECORD_POINTER = INPUT_BUFFER_PTR + 1;
```

```
5                              REC_ADJUST = -1;
4                              END;
3                          CONV$AB_FLAGS [ CONV$V_LAST_CR ] = _SET; );
3           TES;
3
3           !
3           ! If we can $PUT the STM buffer, then do it.  Adjust
3           !       buffer values appropriately to reflect what
3           !       happens here.
3           !
3           NEW_REC_SIZ = .CONV$GW_IN_REC_SIZ + .REC_ADJUST;
3
3           IF .PUT  OR
4                   ((.CONV$GL_STM_REC_LEN + .NEW_REC_SIZ)  GTRU  STM_BUF_SIZ )
3           THEN
4               BEGIN
4               CONV$GW_OUT_REC_SIZ = .CONV$GL_STM_REC_LEN ;
4               CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_STM_BUF;
5               IF NOT ( STATUS = CONV$$PUT_RECORD () )
4               THEN
4                   RETURN .STATUS;
4               !
4               ! Reset STM buffer pointer to beginning of buffer;
4               !    zero current STM record length.
4               !
4               STREAM_BUFFER_PTR = .CONV$GL_STM_BUF;
4               CONV$GL_STM_REC_LEN = 0;
3               END;
3
3           CH$MOVE ( .NEW_REC_SIZ, .NEW_RECORD_POINTER, .STREAM_BUFFER_PTR);
3           CONV$GL_STM_REC_LEN = .CONV$GL_STM_REC_LEN + .NEW_REC_SIZ ;
3           CONV$GW_OUT_REC_SIZ = .CONV$GL_STM_REC_LEN;
3           CONV$AB_OUT_RAB [ RAB$L_RBF ] = .CONV$GL_STM_BUF;
2           END;
2
2       RETURN CONV$_SUCCESS;
2
1       END;
```

```
                                        0FFC 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 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          56  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
```

```
%SBTTL  'PUT_RECORD'
GLOBAL ROUTINE CONV$$PUT_RECORD  =
!++
!
! Functional Description:
!
!       $PUT record to output file.
!
! Calling Sequence:
!
!       CONV$$PUT_RECORD ()
!
! Input Parameters:
!
!       None.
!
! Implicit Inputs:
!
!       CONV$AB_OUT_RAB
!
! Output Parameters:
!
!       none
!
! Implicit Outputs:
!
!       none
!
! Routine Value:
!
!       See CONV$$EXCEPTION
!
! Routines Called:
!
!       CONV$$EXCEPTION
!
! Side Effects:
!
!       The record is $PUT to the output file
!
!--
    BEGIN
    LOCAL
        STATUS;

    ! Set the size of the record to OUT_REC_SIZE which is set by GET_RECORD
    !
    CONV$AB_OUT_RAB [ RAB$W_RSZ ] = .CONV$GW_OUT_REC_SIZ;

    ! Write the record, if there was an error call the exception handler
    !
    IF NOT ( STATUS = $PUT ( RAB=CONV$AB_OUT_RAB ) )
    THEN
        ! If the exception handler returned with an error
        ! then it was fatal
        !
        STATUS = CONV$$EXCEPTION( .STATUS );
```

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


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