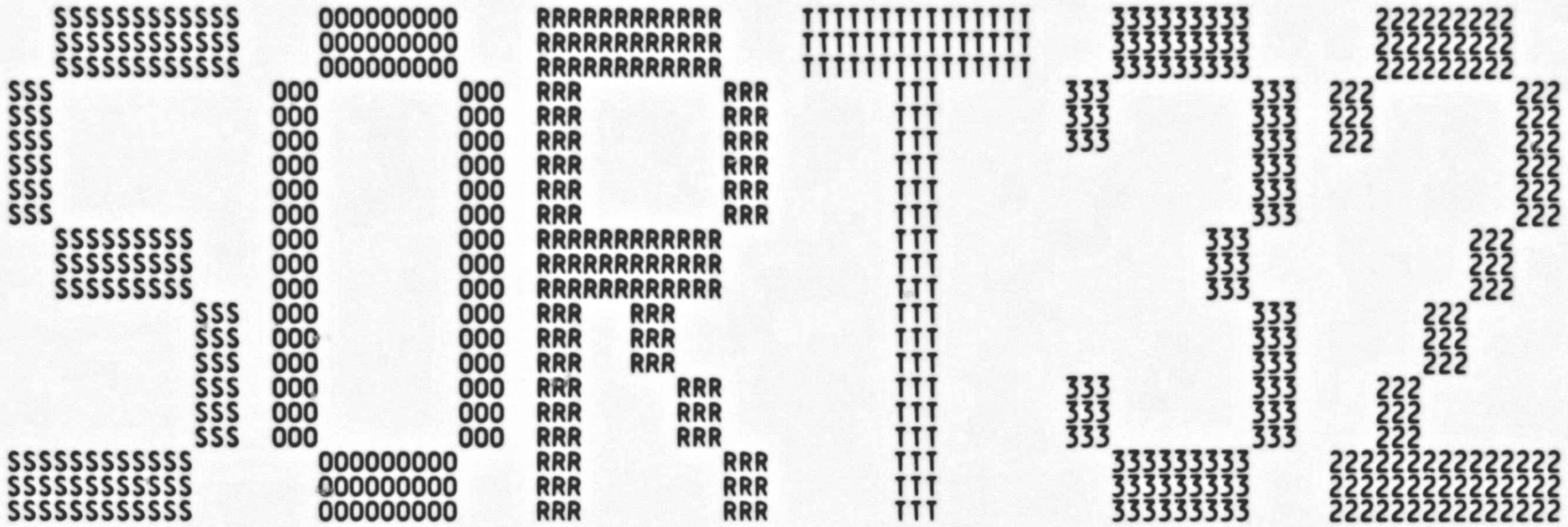
SSSSSSSSSSS 000000000 RRRRRRRRRRR TTTTTTTTTTTTTTT 333333333 222222222
SSSSSSSSSSS 000000000 RRRRRRRRRRR TTTTTTTTTTTTTTT 333333333 222222222
SSSSSSSSSSS 000000000 RRRRRRRRRRR TTTTTTTTTTTTTTT 333333333 222222222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 222 222
SSS 000 000 RRR RRR TTT 333 222
SSS 000 000 RRR RRR TTT 333 222
SSS 000 000 RRR RRR TTT 333 222
SSSSSSSSS 000 000 RRRRRRRRRRR TTT 333 222
SSSSSSSSS 000 000 RRRRRRRRRRR TTT 333 222
SSSSSSSSS 000 000 RRRRRRRRRRR TTT 333 222
SSS 000 000 RRR RRR TTT 333 222
SSS 000 000 RRR RRR TTT 333 222
SSS 000 000 RRR RRR TTT 333 222
SSS 000 000 RRR RRR TTT 333 333 222
SSS 000 000 RRR RRR TTT 333 333 222
SSS 000 000 RRR RRR TTT 333 333 222
SSSSSSSSSSS 000000000 RRR RRR TTT 333333333 222222222222222
SSSSSSSSSSS 000000000 RRR RRR TTT 333333333 222222222222222
SSSSSSSSSSS 000000000 RRR RRR TTT 333333333 222222222222222

FILEID**SORSPCUTI

```
  SSSSSSSS    OOOOOO    RRRRRRRR      SSSSSSSS  PPPPPPPP     CCCCCCCC  UU      UU  TTTTTTTTTT    IIIIII
  SSSSSSSS    OOOOOO    RRRRRRRR      SSSSSSSS  PPPPPPPP     CCCCCCCC  UU      UU  TTTTTTTTTT    IIIIII
SS           OO      OO  RR      RR  SS        PP      PP  CC          UU      UU      TT        II
SS           OO      OO  RR      RR  SS        PP      PP  CC          UU      UU      TT        II
SS           OO      OO  RR      RR  SS        PP      PP  CC          UU      UU      TT        II
SS           OO      OO  RR      RR  SS        PP      PP  CC          UU      UU      TT        II
  SSSSSS     OO      OO  RRRRRRRR      SSSSSS  PPPPPPPP    CC          UU      UU      TT        II
  SSSSSS     OO      OO  RRRRRRRR      SSSSSS  PPPPPPPP    CC          UU      UU      TT        II
        SS   OO      OO  RR   RR              SS PP          CC          UU      UU      TT        II
        SS   OO      OO  RR   RR              SS PP          CC          UU      UU      TT        II
        SS   OO      OO  RR    RR            SS PP          CC          UU      UU      TT        II          ....
        SS   OO      OO  RR    RR            SS PP          CC          UU      UU      TT        II          ....
SSSSSSSS      OOOOOO    RR      RR  SSSSSSSS  PP            CCCCCCCCC UUUUUUUUUU      TT      IIIIII        ....
SSSSSSSS      OOOOOO    RR      RR  SSSSSSSS  PP            CCCCCCCCC UUUUUUUUUU      TT      IIIIII        ....

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

```
;     1          0001  0 MODULE SOR$SPEC_UTIL (
;     2          0002  0                 IDENT = 'V04-000'       ! File: SORSPCUTI.B32 Edit: PDG3024
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**************************************************************************************
;     7          0007  1 !*                                                                                    *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                           *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                             *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                               *
;    11          0011  1 !*                                                                                    *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED            *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE           *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER           *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY           *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY           *
;    17          0017  1 !*  TRANSFERRED.                                                                       *
;    18          0018  1 !*                                                                                    *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE           *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT           *
;    21          0021  1 !*  CORPORATION.                                                                       *
;    22          0022  1 !*                                                                                    *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS           *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                           *
;    25          0025  1 !*                                                                                    *
;    26          0026  1 !*                                                                                    *
;    27          0027  1 !**************************************************************************************
;    28          0028  1 !
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 !
;    32          0032  1 ! FACILITY:     VAX-11 SORT/MERGE
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       This module contains support routines for specification file features.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  VAX/VMS user mode
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: Peter D Gilbert, CREATION DATE: 25-Aug-1982
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 !       T03-015         Original
;    45          0045  1 !       T03-016 Put linkages on SOR$$COMPARE.  Rework the way TDT entries are
;    46          0046  1 !               processed.  Return 0 if no tie-breaking and the strings collate
;    47          0047  1 !               as equal.  PDG 13-Dec-1982
;    48          0048  1 !       T03-017 Some fixes in SOR$$COLL_CMP and GET_COLL.  PDG 28-Dec-1982
;    49          0049  1 !       T03-018 Add checks for short records in SOR$$TDT and SOR$$REFORM.
;    50          0050  1 !               PDG 3-Jan-1983
;    51          0051  1 !       T03-019 Remove collating sequence stuff (from this module).
;    52          0052  1 !               PDG 26-Jan-1983
;    53          0053  1 !       T03-020 Change CH$COPY to use a pad character.  PDG 8-Feb-1983
;    54          0054  1 !       T03-021 Use KFT_NDE_SIZ for length in internal node.  PDG 12-Feb-1983
;    55          0055  1 !       T03-022 Check the KFT_BUILD flag in SOR$$REFORM.  PDG 10-May-1983
;    56          0056  1 !       T03-023 Make CFT_CON_ADR relative.  PDG 25-Jan-1984
;    57          0057  1 !       T03-024 WHILE_FAIL_ is now defined in SRTSPC.REQ.  PDG 1-Feb-1984
```

```
;   58          0058  1 !--
```

```
;    60          0059  1 !       Require files
;    61          0060  1 !
;    62          0061  1 %IF %BLISS(BLISS32) %THEN
;    63          0062  1 REQUIRE 'SRC$:COM';                ! Common definitions for VAX-11 SORT/MERGE
;    64          0132  1 %FI
;    65          0133  1 LIBRARY 'SRC$:SRTSPC';             ! Define symbols for spec file processing
;    66          0134  1
;    67          0135  1 !       Linkage declarations
;    68          0136  1 !
;    69          0137  1 LITERAL
;    70          0138  1     PT_REG = %BLISS16(4) %BLISS32(4),
;    71          0139  1     ST_REG = %BLISS16(3) %BLISS32(3),
;    72          0140  1     XX_REG = 2;                                ! Parameter register
;    73          0141  1
;    74          0142  1 !       Routine declarations
;    75          0143  1 !
;    76          0144  1 FORWARD ROUTINE
;    77          0145  1     SOR$$TDT:           CA_LINKAGE,     ! Evaluate a test
;    78          0146  1     SOR$$RDT:           CA_LINKAGE,     ! Determine record type
;    79          0147  1     SOR$$REFORM:        CA_LINKAGE;     ! Reformat a record
;    80          0148  1
;    81          0149  1 !       Macro declarations
;    82          0150  1 !
;    83        L 0151  1 %IF NOT %DECLARED(%QUOTE BASE_)
;    84        U 0152  1 %THEN
;    85        U 0153  1 MACRO
;    86        U 0154  1     BASE_ =     0, 0, 0, 0 %;
;    87          0155  1 %FI
```

```
;    89          0156  1 GLOBAL ROUTINE SOR$$TDT
;    90          0157  1         (
;    91          0158  1         INPREC: REF VECTOR[2],          ! Length/address of input record
;    92          0159  1         TDTPTR: REF TDT_TAB[]           ! Test definitions
;    93        L 0160  1 %IF %BLISS(BLISS16)
;    94        U 0161  1 %THEN
;    95        U 0162  1         ,
;    96        U 0163  1         COMP:   REF VECTOR[,BYTE],      ! Addr of routine to do simple compares
;    97        U 0164  1         FDT:    REF FDT_TAB[],          ! Field definition table
;    98          0165  1         CFT:    REF CFT_TAB[]           ! Constant definition table
;    99          0166  1 %FI
;   100          0167  1         ):      CA_LINKAGE =
;   101          0168  1 !++
;   102          0169  1 ! FUNCTIONAL DESCRIPTION:
;   103          0170  1 !
;   104          0171  1 !       This routine evaluates a test.
;   105          0172  1 !
;   106          0173  1 ! FORMAL PARAMETERS:
;   107          0174  1 !
;   108          0175  1 !       As described above.
;   109          0176  1 !
;   110          0177  1 !       Note that COMP, FDT, and CFT could be bound to the locations
;   111          0178  1 !       in the context area that hold the addresses.
;   112          0179  1 !
;   113          0180  1 ! IMPLICIT INPUTS:
;   114          0181  1 !
;   115          0182  1 !       NONE
;   116          0183  1 !
;   117          0184  1 ! IMPLICIT OUTPUTS:
;   118          0185  1 !
;   119          0186  1 !       NONE
;   120          0187  1 !
;   121          0188  1 ! ROUTINE VALUE:
;   122          0189  1 !
;   123          0190  1 !       0       indicates test failed
;   124          0191  1 !       1       indicates test passed
;   125          0192  1 !
;   126          0193  1 ! SIDE EFFECTS:
;   127          0194  1 !
;   128          0195  1 !       NONE
;   129          0196  1 !
;   130          0197  1 !--
;   131          0198  2     BEGIN
;   132          0199  2     LOCAL
;   133          0200  2         TDT:    REF TDT_TAB[],  ! Local pointer to test descriptions
;   134          0201  2         RES;                    ! Running total/result
;   135          0202  2
;   136          0203  2     CA_AREA_(CA);
;   137          0204  2
;   138          0205  3     BEGIN
;   139        L 0206  3 %IF %BLISS(BLISS32)
;   140          0207  3 %THEN
;   141          0208  3     BIND
;   142          0209  3         FDT = CA[CA_FDT_ADR]: REF FDT_TAB[],    ! Field definition table
;   143          0210  3         CFT = CA[CA_CFT_ADR]: REF CFT_TAB[];    ! Constant definition table
;   144          0211  3     EXTERNAL ROUTINE
;   145          0212  3         SOR$$COMPARE:   CA_LINKAGE;
```

```
;  146     0213  3     BIND
;  147     0214  3         COMP = SOR$$COMPARE: VECTOR[,BYTE];     ! Addr of comparison routine
;  148     0215  3 %FI
;  149     0216  3
;  150     0217  3         !+
;  151     0218  3         ! The test definition table consists of simple comparisons, and a flag
;  152     0219  3         ! indicating whether the result of this comparison should be ANDed or
;  153     0220  3         ! ORed with the running total/result.  Another flag indicates whether
;  154     0221  3         ! this is the last simple comparison (there is always at least one).
;  155     0222  3         !
;  156     0223  3         ! For example, the following:
;  157     0224  3         !
;  158     0225  3         ! CONTINUE  CMP1 XXX
;  159     0226  3         ! CONTINUE  CMP2 OP2
;  160     0227  3         !           CMP3 OP3
;  161     0228  3         !
;  162     0229  3         ! corresponds to the condition: ((CMP1 OP2 CMP2) OP3 CMP3)
;  163     0230  3         !-
;  164     0231  3
;  165     0232  3
;  166     0233  3         ! Get the address of the first test entry
;  167     0234  3         !
;  168     0235  3         TDT = TDTPTR[0, BASE_];
;  169     0236  3
;  170     0237  3         WHILE 1 DO
;  171     0238  4             BEGIN
;  172     0239  4             LOCAL
;  173     0240  4                 FDT_IX,             ! Index into FDT (or CFT) table
;  174     0241  4                 TYPE,               ! Data type of the operands
;  175     0242  4                 FLD1: VECTOR[2],    ! Length/address of first operand
;  176     0243  4                 FLD2: VECTOR[2];    ! Length/address of first operand
;  177     0244  4
;  178     0245  4             IF .TDT[0,TDT_TRUE] THEN RETURN 1;
;  179     0246  4
;  180     0247  4             !+
;  181     0248  4             ! Find the datatypes, lengths, and addresses of the fields/data
;  182     0249  4             ! to be compared.
;  183     0250  4             !-
;  184     0251  4
;  185     0252  4             ! The first field is always a field (not a constant)
;  186     0253  4             !
;  187     0254  4             FDT_IX = .TDT[0,TDT_FLD_ONE];
;  188     0255  4             TYPE = .FDT[.FDT_IX, FDT_TYPE];
;  189     0256  4             FLD1[0] = .FDT[.FDT_IX, FDT_FLD_SIZ];
;  190     0257  4             FLD1[1] = .FDT[.FDT_IX, FDT_FLD_POS];
;  191     0258  4             IF .FLD1[0] + .FLD1[1] GTRU .INPREC[0]
;  192     0259  4             THEN
;  193     0260  5                 BEGIN
;  194     0261  5                 !
;  195     0262  5                 ! If this is not a text field, it's an error
;  196     0263  5                 !
;  197     0264  5                 IF .TYPE NEQ DT_T THEN RETURN 0;
;  198     0265  5                 FLD1[0] = .INPREC[0] - .FLD1[1];
;  199     0266  5                 IF .FLD1[0] LSS 0 THEN FLD1[0] = 0;
;  200     0267  4                 END;
;  201     0268  4             FLD1[1] = .FLD1[1] + .INPREC[1];
;  202     0269  4
```

```
;  203          0270  4          ! The second field may be a field or a constant
;  204          0271  4          !
;  205          0272  4          FDT_IX = .TDT[0,TDT_FLD_TWO];
;  206          0273  4          IF .TDT[0,TDT_CONSTANT]
;  207          0274  4          THEN
;  208          0275  5              BEGIN
;  209          0276  5              FLD2[0] = .CFT[.FDT_IX, CFT_CON_LEN];
;  210          0277  5              FLD2[1] = .CFT[.FDT_IX, CFT_CON_ADR] + CFT[0, BASE_];
;  211          0278  5              END
;  212          0279  4          ELSE
;  213          0280  5              BEGIN
;  214          0281  5              FLD2[0] = .FDT[.FDT_IX, FDT_FLD_SIZ];
;  215          0282  5              FLD2[1] = .FDT[.FDT_IX, FDT_FLD_POS];
;  216          0283  5              IF .FLD2[0] + .FLD2[1] GTRU .INPREC[0]
;  217          0284  5              THEN
;  218          0285  6                  BEGIN
;  219          0286  6                  !
;  220          0287  6                  ! If this is not a text field, it's an error
;  221          0288  6                  !
;  222          0289  6                  IF .FDT[.FDT_IX, FDT_TYPE] NEQ DT_T THEN RETURN 0;
;  223          0290  6                  FLD2[0] = .INPREC[0] - .FLD2[1];
;  224          0291  6                  IF .FLD2[0] LSS 0 THEN FLD2[0] = 0;
;  225          0292  5                  END;
;  226          0293  5              FLD2[1] = .FLD2[1] + .INPREC[1];
;  227          0294  4              END;
;  228          0295  4
;  229          0296  4          IF
;  230          0297  5              BEGIN
;  231          0298  5              LOCAL CMP;
;  232          0299  5
;  233          0300  5              CMP = CA_LINKAGE(COMP[0], .TYPE, FLD1[0], FLD2[0]);
;  234          0301  5
;  235          0302  5              SELECTONE .CMP OF
;  236          0303  5                  SET
;  237          0304  5                  [0]:     .TDT[0,TDT_EQL];
;  238          0305  5                  [+1]:    .TDT[0,TDT_GTR];
;  239          0306  5                  [-1]:    .TDT[0,TDT_LSS];
;  240          0307  5                  [OTHERWISE]:    RETURN .CMP;    ! Unexpected
;  241          0308  5                  TES
;  242          0309  5              END
;  243          0310  4          THEN
;  244          0311  5              BEGIN
;  245          0312  5              LOCAL X;
;  246          0313  5              X = .TDT[0,TDT_GOTO];
;  247          0314  5              IF .X EQL 0 THEN RETURN 0;
;  248          0315  5              TDT = TDT[.X, BASE_];       ! Goto new test item
;  249          0316  5              END
;  250          0317  4          ELSE
;  251          0318  4              TDT = TDT[1, BASE_];        ! Advance to next test
;  252          0319  4
;  253          0320  3          END;
;  254          0321  3
;  255          0322  3      RETURN 0;
;  256          0323  2      END;
;  257          0324  1      END;
```

```
                                                        .TITLE  SOR$SPEC_UTIL
                                                        .IDENT  \V04-000\

                                                        .EXTRN  SOR$$COMPARE

                                                        .PSECT  SOR$RO_CODE,NOWRT,  SHR,  PIC,2

                                                 001C 00000     .ENTRY  SOR$$TDT, Save R2,R3,R4                 ; 0156
                                              5E        10  C2 00002     SUBL2   #16, SP                         ;
                                              52    08  AC  D0 00005     MOVL    TDTPTR, TDT                     ; 0235
                                              53    04  AC  D0 00009     MOVL    INPREC, R3                      ; 0268
                                              04        62  E9 0000D 1$:  BLBC    (TDT), 2$                       ; 0245
                                              50        01  D0 00010     MOVL    #1, R0                          ;
                                                        04  00013     RET                                     ;
                                              50    01  A2  9A 00014 2$:  MOVZBL  1(TDT), FDT_IX                 ; 0254
                                           51 50        06  C5 00018     MULL3   #6, FDT_IX, R1                  ; 0255
                                              51    0110 CB  C0 0001C     ADDL2   272(CA), R1                     ;
                                              54        61  9A 00021     MOVZBL  (R1), TYPE                      ;
                                        08 AE     04    A1  3C 00024     MOVZWL  4(R1), FLD1                     ; 0256
                                        0C AE     02    A1  3C 00029     MOVZWL  2(R1), FLD1+4                   ; 0257
                                           51 08  AE    0C  AE C1 0002E     ADDL3   FLD1+4, FLD1, R1                ; 0258
                                        04 BC           51  D1 00034     CMPL    R1, @INPREC                     ;
                                                        11  1B 00038     BLEQU   3$                              ;
                                              0E        54  D1 0003A     CMPL    TYPE, #14                       ; 0264
                                                        4A  12 0003D     BNEQ    5$                              ;
                                     08 AE    04 BC     0C  AE C3 0003F     SUBL3   FLD1+4, @INPREC, FLD1           ; 0265
                                                        03  18 00046     BGEQ    3$                              ; 0266
                                                    08  AE  D4 00048     CLRL    FLD1                            ;
                                        0C AE     04    A3  C0 0004B 3$:  ADDL2   4(R3), FLD1+4                   ; 0268
                                              50    02  A2  9A 00050     MOVZBL  2(TDT), FDT_IX                  ; 0272
                                              50        06  C4 00054     MULL2   #6, R0                          ; 0276
                                           12             62  E1 00057     BBC     #4, (TDT), 4$                   ; 0273
                                              50    010C CB  C0 0005B     ADDL2   268(CA), R0                     ; 0276
                                              6E        60  9A 00060     MOVZBL  (R0), FLD2                      ;
                                     04 AE        02  A0    0C  CB C1 00063     ADDL3   268(CA), 2(R0), FLD2+4          ; 0277
                                                        2D  11 0006B     BRB     7$                              ; 0273
                                              50    0110 CB  C0 0006D 4$:  ADDL2   272(CA), R0                     ; 0281
                                              6E    04  A0  3C 00072     MOVZWL  4(R0), FLD2                     ;
                                        04 AE     02    A0  3C 00076     MOVZWL  2(R0), FLD2+4                   ; 0282
                                           51 6E     04    AE  C1 0007B     ADDL3   FLD2+4, FLD2, R1                ; 0283
                                        04 BC           51  D1 00080     CMPL    R1, @INPREC                     ;
                                                        0F  1B 00084     BLEQU   6$                              ;
                                              0E        60  91 00086     CMPB    (R0), #14                       ; 0289
                                                        51  12 00089 5$:  BNEQ    13$                             ;
                                     6E 04    BC     04  AE C3 0008B     SUBL3   FLD2+4, @INPREC, FLD2           ; 0290
                                                        02  18 00091     BGEQ    6$                              ; 0291
                                                    6E  D4 00093     CLRL    FLD2                            ;
                                        04 AE     04    A3  C0 00095 6$:  ADDL2   4(R3), FLD2+4                   ; 0293
                                                    5E  DD 0009A 7$:  PUSHL   SP                              ; 0300
                                                    0C  AE 9F 0009C     PUSHAB  FLD1                            ;
                                                        54  DD 0009F     PUSHL   TYPE                            ;
                                  00000000G     00        03  FB 000A1     CALLS   #3, COMP                        ;
                                                        50  D5 000A8     TSTL    CMP                             ; 0304
                                                        06  12 000AA     BNEQ    8$                              ;
                                           26             62  E1 000AC     BBC     #1, (TDT), 11$                  ;
                                                        18  11 000B0     BRB     10$                             ;
                                              01        50  D1 000B2 8$:  CMPL    CMP, #1                         ; 0305
```

```
                                      06  12 000B5        BNEQ    9$                     ;
        1B               62           03  E1 000B7        BBC     #3, (TDT), 11$         ;
                                      0D  11 000BB        BRB     10$                    ;
             FFFFFFFF    8F           50  D1 000BD 9$:    CMPL    CMP, #-1               ; 0306
                                      18  12 000C4        BNEQ    14$                    ;
        0C               62           02  E1 000C6        BBC     #2, (TDT), 11$         ;
                         50     03    A2  9A 000CA 10$:   MOVZBL  3(TDT), X              ; 0313
                                      0C  13 000CE        BEQL    13$                    ; 0314
                         52         6240  DE 000D0        MOVAL   (TDT)[X], TDT          ; 0315
                                      03  11 000D4        BRB     12$                    ; 0296
                         52           04  C0 000D6 11$:   ADDL2   #4, TDT                ; 0318
                                    FF31  31 000D9 12$:   BRW     1$                     ; 0237
                                      50  D4 000DC 13$:   CLRL    R0                     ; 0324
                                      04     000DE 14$:   RET                            ;

; Routine Size:  223 bytes,    Routine Base:  SOR$RO_CODE + 0000
```

```
;  259          0325  1 GLOBAL ROUTINE SOR$$RDT
;  260          0326  1         (
;  261          0327  1         INPREC: REF VECTOR[2],          ! Length/address of input record
;  262        L 0328  1 %IF %BLISS(BLISS16)
;  263        U 0329  1 %THEN
;  264        U 0330  1         RDT:    REF RDT_TAB[],         ! Record definition table
;  265        U 0331  1         COMP:   REF VECTOR[,BYTE],     ! Addr of routine to do simple compares
;  266        U 0332  1         TDT:    REF TDT_TAB[],         ! Test definition table
;  267        U 0333  1         FDT:    REF FDT_TAB[],         ! Field definition table
;  268        U 0334  1         CFT:    REF CFT_TAB[],         ! Constant definition table
;  269          0335  1 %FI
;  270          0336  1         RDTPTR: REF VECTOR[1]          ! Pointer to RDT entry (output)
;  271          0337  1         ):      CA_LINKAGE =
;  272          0338  1 !++
;  273          0339  1 !
;  274          0340  1 ! FUNCTIONAL DESCRIPTION:
;  275          0341  1 !
;  276          0342  1 !       This routine determines whether a record should be omitted or
;  277          0343  1 !       included.  If included, it returns the address of the RDT entry.
;  278          0344  1 !
;  279          0345  1 ! FORMAL PARAMETERS:
;  280          0346  1 !
;  281          0347  1 !       As described above.
;  282          0348  1 !
;  283          0349  1 !       Note that RDT, COMP, TDT, FDT, and CFT could be bound to the locations
;  284          0350  1 !       in the context area that hold the addresses.
;  285          0351  1 !
;  286          0352  1 ! IMPLICIT INPUTS:
;  287          0353  1 !
;  288          0354  1 !       NONE
;  289          0355  1 !
;  290          0356  1 ! IMPLICIT OUTPUTS:
;  291          0357  1 !
;  292          0358  1 !       NONE
;  293          0359  1 !
;  294          0360  1 ! ROUTINE VALUE:
;  295          0361  1 !
;  296          0362  1 !       0       indicates the record is to be omitted
;  297          0363  1 !       1       indicates the record is to be included
;  298          0364  1 !               RDTPTR is set to the address of the appropriate RDT table entry
;  299          0365  1 !
;  300          0366  1 ! SIDE EFFECTS:
;  301          0367  1 !
;  302          0368  1 !       NONE
;  303          0369  1 !
;  304          0370  1 !--
;  305          0371  2     BEGIN
;  306          0372  2     LOCAL
;  307          0373  2         RDT_PTR: REF RDT_TAB[];         ! Local pointer to record definitions
;  308          0374  2
;  309          0375  2     CA_AREA_(CA);
;  310          0376  2
;  311          0377  3     BEGIN
;  312        L 0378  3 %IF %BLISS(BLISS32)
;  313          0379  3 %THEN
;  314          0380  3     BIND
;  315          0381  3         RDT = CA[CA_RDT_ADR]: REF RDT_TAB[],    ! Record definition table
```

```
;   316          0382  3          TDT = CA[CA_TDT_ADR]: REF TDT_TAB[],     ! Test definition table
;   317          0383  3          FDT = CA[CA_FDT_ADR]: REF FDT_TAB[],     ! Field definition table
;   318          0384  3          CFT = CA[CA_CFT_ADR]: REF CFT_TAB[];     ! Constant definition table
;   319          0385  3      EXTERNAL ROUTINE
;   320          0386  3          SOR$$COMPARE:   CA_LINKAGE;
;   321          0387  3      BIND
;   322          0388  3          COMP = SOR$$COMPARE: VECTOR[,BYTE];      ! Addr of comparison routine
;   323          0389  3  %FI
;   324          0390  3      ! Get a local pointer to the record definition table
;   325          0391  3      !
;   326          0392  3      RDT_PTR = RDT[0,BASE_];
;   327          0393  3
;   328          0394  3      ! Advance RDT_PTR until we find a test that passes
;   329          0395  3      !
;   330          0396  3      WHILE_FAIL_('RDT');
;   331          0397  3
;   332          0398  3      ! Now determine whether we should omit or include this thing.
;   333          0399  3      !
;   334          0400  3      IF .RDT_PTR[0, RDT_INCLUDE]
;   335          0401  3      THEN
;   336          0402  4          BEGIN
;   337          0403  4          !
;   338          0404  4          ! Include the record, so store the address of the RDT table entry
;   339          0405  4          !
;   340          0406  4          RDTPTR[0] = RDT_PTR[0, BASE_];
;   341          0407  4          RETURN 1;
;   342          0408  4          END
;   343          0409  3      ELSE
;   344          0410  4          BEGIN
;   345          0411  4          !
;   346          0412  4          ! Omit the record
;   347          0413  4          !
;   348          0414  4          RETURN 0;
;   349          0415  3          END;
;   350          0416  3
;   351          0417  2      END;
;   352          0418  1      END;


                                                0004 00000          .ENTRY  SOR$$RDT, Save R2                                   ; 0325
                                      52  0104   CB   D0 00002      MOVL    260(CA), RDT_PTR                                    ; 0392
                                1D    62        01   E1 00007 1$:  BBC     #1, (RDT_PTR), 2$                                   ; 0396
                                      50    01  A2   9A 0000B      MOVZBL  1(RDT_PTR), R0
                                          0114 DB40  DF 0000F      PUSHAL  @276(CA)[R0]
                                            04  AC   DD 00014      PUSHL   INPREC
                                    FF05  CF        02  FB 00017  CALLS   #2, SOR$$TDT
                                          01        50  D1 0001C  CMPL    PASS, #1
                                                    14  1A 0001F  BGTRU   4$
                                                    05  13 00021  BEQL    2$
                                      52            06  C0 00023  ADDL2   #6, RDT_PTR
                                                    DF  11 00026  BRB     1$
                                      08            62  E9 00028 2$:  BLBC    (RDT_PTR), 3$                                   ; 0400
                                        08  BC        52  D0 0002B      MOVL    RDT_PTR, @RDTPTR                                ; 0406
                                      50            01  D0 0002F      MOVL    #1, R0                                          ; 0410
```

```
                04 00032              RET
                                                                                ;
                50  D4 00033 3$:       CLRL    R0                              ; 0414
                04 00035 4$:          RET                                     ; 0418

; Routine Size:  54 bytes,    Routine Base:  SOR$RO_CODE + 00DF
```

```
;  354          0419  1 GLOBAL ROUTINE SOR$$REFORM
;  355          0420  1         (
;  356          0421  1         INPREC: REF VECTOR[2],          ! Length/address of input record
;  357          0422  1         KFTPTR: REF KFT_TAB[],
;  358        L 0423  1 %IF %BLISS(BLISS16)
;  359        U 0424  1 %THEN
;  360        U 0425  1         RDT:    REF RDT_TAB[],         ! Record definition table
;  361        U 0426  1         COMP:   REF VECTOR[,BYTE],     ! Addr of routine to do simple compares
;  362        U 0427  1         TDT:    REF TDT_TAB[],         ! Test definition table
;  363        U 0428  1         FDT:    REF FDT_TAB[],         ! Field definition table
;  364        U 0429  1         CFT:    REF CFT_TAB[],         ! Constant definition table
;  365          0430  1 %FI
;  366          0431  1         RESULT: REF VECTOR[,BYTE],     ! Address of output area
;  367          0432  1         RECLEN: REF VECTOR[1,WORD]     ! Output format record length
;  368          0433  1         ):      CA_LINKAGE =
;  369          0434  1 !++
;  370          0435  1 !
;  371          0436  1 ! FUNCTIONAL DESCRIPTION:
;  372          0437  1 !
;  373          0438  1 !       This routine reformats a record into a result area.
;  374          0439  1 !
;  375          0440  1 ! FORMAL PARAMETERS:
;  376          0441  1 !
;  377          0442  1 !       As described above.
;  378          0443  1 !
;  379          0444  1 !       Note that RDT, COMP, TDT, FDT, and CFT could be bound to the locations
;  380          0445  1 !       in the context area that hold the addresses.
;  381          0446  1 !
;  382          0447  1 ! IMPLICIT INPUTS:
;  383          0448  1 !
;  384          0449  1 !       NONE
;  385          0450  1 !
;  386          0451  1 ! IMPLICIT OUTPUTS:
;  387          0452  1 !
;  388          0453  1 !       NONE
;  389          0454  1 !
;  390          0455  1 ! ROUTINE VALUE:
;  391          0456  1 !
;  392          0457  1 !       1       indicates everything went okay.
;  393          0458  1 !
;  394          0459  1 ! SIDE EFFECTS:
;  395          0460  1 !
;  396          0461  1 !       NONE
;  397          0462  1 !
;  398          0463  1 ! NOTES:
;  399          0464  1 !
;  400          0465  1 !       The protocol for using this routine is as follows:
;  401          0466  1 !
;  402          0467  1 !       Z = SOR$$RDT( INPREC, ..., RDTPTR )
;  403          0468  1 !       SELECTONE .Z OF
;  404          0469  1 !               SET
;  405          0470  1 !               [0]:    ...omit the record...;
;  406          0471  1 !               [1]:    BEGIN
;  407          0472  1 !                       KFT_IX = .RDTPTR[0, RDT_KFT_IDX];
;  408          0473  1 !                       Z = SOR$$REFORM( INPREC, KFT[.KFT_IX,BASE_], ... );
;  409          0474  1 !                       IF .Z NEQ 1 THEN ...error from comparison...;
;  410          0475  1 !                       END;
```

```
;  411          0476  1 !                [OTHERWISE]: ...error from comparison...;
;  412          0477  1 !                TES;
;  413          0478  1 !--
;  414          0479  2     BEGIN
;  415          0480  2     LOCAL
;  416          0481  2         KFT_PTR: REF KFT_TAB[],          ! Local pointer to KFT table
;  417          0482  2         FIRSTDATA:      WORD;            ! Offset to first data field
;  418          0483  2
;  419          0484  2     CA_AREA_(CA);
;  420          0485  2
;  421          0486  2     BEGIN
;  422        L 0487  2 %IF %BLISS(BLISS32)
;  423          0488  2 %THEN
;  424          0489  2     BIND
;  425          0490  2         RDT = CA[CA_RDT_ADR]: REF RDT_TAB[],    ! Record definition table
;  426          0491  2         TDT = CA[CA_TDT_ADR]: REF TDT_TAB[],    ! Test definition table
;  427          0492  2         FDT = CA[CA_FDT_ADR]: REF FDT_TAB[],    ! Field definition table
;  428          0493  2         CFT = CA[CA_CFT_ADR]: REF CFT_TAB[];    ! Constant definition table
;  429          0494  2     EXTERNAL ROUTINE
;  430          0495  2         SOR$$COMPARE:   CA_LINKAGE;
;  431          0496  2     BIND
;  432          0497  2         COMP = SOR$$COMPARE: VECTOR[,BYTE];     ! Addr of comparison routine
;  433          0498  2 %FI
;  434          0499  2
;  435          0500  2         !+
;  436          0501  2         ! The key/data field definition table consists of field definitions,
;  437          0502  2         ! which define the fields in a record; some of these may be conditional
;  438          0503  2         ! data fields.  As an example, the following specification file:
;  439          0504  2         !
;  440          0505  2         !   /DATA=FLD1
;  441          0506  2         !   /DATA=( IF COND1 THEN CONST1
;  442          0507  2         !           IF COND2 THEN CONST2
;  443          0508  2         !           ELSE CONST3 )
;  444          0509  2         !   /DATA=FLD3
;  445          0510  2         !   /DATA=FLD4
;  446          0511  2         !
;  447          0512  2         ! corresponds to the following field definition table entries:
;  448          0513  2         !
;  449          0514  2         ! CONTINUE  FLD1
;  450          0515  2         ! CONTINUE  CONST1  COND    COND1
;  451          0516  2         ! CONTINUE  CONST2  COND    COND2
;  452          0517  2         ! CONTINUE  CONST3
;  453          0518  2         ! CONTINUE  FLD3
;  454          0519  2         !           FLD4
;  455          0520  2         !
;  456          0521  2         ! Note that the ELSE part of the conditional data definition does not
;  457          0522  2         ! have the COND flag set.
;  458          0523  2         !-
;  459          0524  2
;  460          0525  2
;  461          0526  2         ! Initialize the local pointer to the KFT table
;  462          0527  2         ! Initialize the output format record length
;  463          0528  2         !
;  464          0529  2         KFT_PTR = KFTPTR[0,BASE_];
;  465          0530  2         REC[EN[0] = 0;
;  466          0531  2         FIRSTDATA = -1;
;  467          0532  3
```

```
;  468          0533  3
;  469          0534  3      ! While there are more fields
;  470          0535  3      !
;  471          0536  3      WHILE 1 DO
;  472          0537  4          BEGIN
;  473          0538  4          LOCAL
;  474          0539  4              FLD: VECTOR[2],     ! Length/address of field or constant
;  475          0540  4              FDT_IX;             ! Index into FDT (or CFT) table
;  476          0541  4
;  477          0542  4          ! Advance KFT_PTR until we find a test that passes
;  478          0543  4          !
;  479          0544  4          WHILE_FAIL_('KFT');
;  480          0545  4
;  481          0546  4          ! Determine whether we should grab the field from the record
;  482          0547  4          ! or from the constant table.
;  483          0548  4          !
;  484          0549  4          FDT_IX = .KFT_PTR[0,KFT_FDT_IDX];
;  485          0550  4          IF .KFT_PTR[0,KFT_CONSTANT]
;  486          0551  4          THEN
;  487          0552  5              BEGIN
;  488          0553  5              FLD[0] = .CFT[.FDT_IX, CFT_CON_LEN];
;  489          0554  5              FLD[1] = .CFT[.FDT_IX, CFT_CON_ADR] + CFT[0, BASE_];
;  490          0555  5              END
;  491          0556  4          ELSE
;  492          0557  5              BEGIN
;  493          0558  5              FLD[0] = KFT_UNITS_(KFT_PTR);       ! Get size in bytes
;  494          0559  5              FLD[1] = .FDT[.FDT_IX, FDT_FLD_POS];
;  495          0560  5              IF .FLD[0] + .FLD[1] GTRU .INPREC[0]
;  496          0561  5              THEN
;  497          0562  6                  BEGIN
;  498          0563  6                  !
;  499          0564  6                  ! If this is not a text field, it's an error
;  500          0565  6                  !
;  501          0566  6                  IF .FDT[.FDT_IX, FDT_TYPE] NEQ DT_T THEN RETURN 0;
;  502          0567  6                  FLD[0] = .INPREC[0] - .FLD[1];
;  503          0568  6                  IF .FLD[0] LSS 0 THEN FLD[0] = 0;
;  504          0569  5                  END;
;  505          0570  5              FLD[1] = .FLD[1] + .INPREC[1];
;  506          0571  4              END;
;  507          0572  4
;  508        L 0573  4          %IF MAX(TYP_K_RECORD,TYP_K_TAG) GEQ MIN(TYP_K_INDEX,TYP_K_ADDRESS)
;  509        U 0574  4          %THEN
;  510          0575  4              %ERROR('The following test won''t work') %FI
;  511          0576  4
;  512          0577  4          ! Copy the field to its place in the internal format record
;  513          0578  4          !
;  514          0579  4          IF .KFT_PTR[0,KFT_BUILD]
;  515          0580  4          THEN
;  516          0581  5              BEGIN
;  517          0582  5              CH$COPY(.FLD[0], .FLD[1], .CA[CA_PAD],
;  518          0583  5                  .KFT_PTR[0,KFT_NDE_SIZ], RESULT[.KFT_PTR[0,KFT_NDE_POS]]);
;  519          0584  5              IF .KFT_PTR[0,KFT_DATA]
;  520          0585  5                  OR .CA[CA_PROCESS] GEQ MIN(TYP_K_INDEX,TYP_K_ADDRESS)
;  521          0586  5              THEN
;  522          0587  6                  BEGIN
;  523          0588  6                  RECLEN[0] = MAXU(.RECLEN[0], .KFT_PTR[0,KFT_NDE_POS]+.FLD[0]);
;  524          0589  6                  FIRSTDATA = MINU(.FIRSTDATA, .KFT_PTR[0,KFT_NDE_POS]);
```

```
;  525          0590  5                END;
;  526          0591  4            END;
;  527          0592  4
;  528          0593  4        ! If we were in a conditional part of the record definition,
;  529          0594  4        ! advance KFT_PTR to the end of the conditional entries.
;  530          0595  4        !
;  531          0596  4        WHILE .KFT_PTR[0,KFT_CONDX] DO KFT_PTR = KFT_PTR[1,BASE_];
;  532          0597  4
;  533          0598  4        ! See whether this record definition is continued
;  534          0599  4        !
;  535          0600  4        IF NOT .KFT_PTR[0,KFT_CONTINUE] THEN EXITLOOP;
;  536          0601  4
;  537          0602  4        ! Advance KFT_PTR to the next entry
;  538          0603  4        !
;  539          0604  4        KFT_PTR = KFT_PTR[1,BASE_];
;  540          0605  4
;  541          0606  3        END;
;  542          0607  3
;  543          0608  3    RECLEN[0] = .RECLEN[0] - .FIRSTDATA;
;  544          0609  3    IF .CA[CA_PROCESS] GEQ MIN(TYP_K_INDEX,TYP_K_ADDRESS)
;  545          0610  3    THEN
;  546          0611  3        RECLEN[0] = .RECLEN[0] + 6;     ! Add 6 bytes for the RFA
;  547          0612  3
;  548          0613  3    RETURN 1;
;  549          0614  2    END;
;  550          0615  1    END;


                                07FC 00000          .ENTRY  SOR$$REFORM, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 0419
                                                            R10
                            5E      08  C2 00002            SUBL2   #8, SP                              ;
                            59  0110    CB  9E 00005        MOVAB   272(CA), R9                         ; 0492
                            58  010C    CB  9E 0000A        MOVAB   268(CA), R8                         ; 0493
                            56    08    AC  D0 0000F        MOVL    KFTPTR, KFT_PTR                     ; 0529
                            57    10    AC  D0 00013        MOVL    RECLEN, R7                          ; 0530
                                        67  B4 00017        CLRW    (R7)                                ;
                            5A          01  AE 00019        MNEGW   #1, FIRSTDATA                       ; 0531
                    1C      03  A6      03  E1 0001C 1$:    BBC     #3, 3(KFT_PTR), 3$                  ; 0544
                            50      05  A6  9A 00021        MOVZBL  5(KFT_PTR), R0                      ;
                                0114 DB40   DF 00025        PUSHAL  @276(CA)[R0]                        ;
                                  04    AC  DD 0002A        PUSHL   INPREC                              ;
                          FEB9  CF          02  FB 0002D    CALLS   #2, SOR$$TDT                        ;
                            01              50  D1 00032    CMPL    PASS, #1                            ;
                                            01  1B 00035    BLEQU   2$                                  ;
                                                04 00037    RET                                         ;
                                            03  13 00038 2$: BEQL    3$                                  ;
                                        00C7    31 0003A    BRW     15$                                 ;
                            50      04  A6  9A 0003D 3$:    MOVZBL  4(KFT_PTR), FDT_IX                  ; 0549
                    51      50          06  C5 00041        MULL3   #6, FDT_IX, R1                      ; 0553
                    0F      03  A6      01  E1 00045        BBC     #1, 3(KFT_PTR), 4$                  ; 0550
                    50          51      68  C1 0004A        ADDL3   (R8), R1, R0                        ; 0553
                                6E      60  9A 0004E        MOVZBL  (R0), FLD                           ;
                04  AE      02  A0      68  C1 00051        ADDL3   (R8), 2(R0), FLD+4                  ; 0554
                                        57  11 00057        BRB     9$                                  ; 0550
```

```
                        50      04   A6  9A 00059 4$:    MOVZBL  4(KFT_PTR), FDT_IX                      ; 0558
                        50           06  C4 0005D        MULL2   #6, R0
                07      03   A6      01  E1 00060        BBC     #1, 3(KFT_PTR), 5$                      ;
                        50        00 B840  9A 00065        MOVZBL  @0(R8)[R0], R0                          ;
                                     17  11 0006A        BRB     7$                                      ;
                        50           69  C0 0006C 5$:    ADDL2   (R9), R0                                ;
                        15           60  91 0006F        CMPB    (R0), #21                               ;
                                     0B  12 00072        BNEQ    6$                                      ;
                        50      04   A0  3C 00074        MOVZWL  4(R0), R0                               ;
                        50           02  C6 00078        DIVL2   #2, R0                                  ;
                                     50  D6 0007B        INCL    R0                                      ;
                                     04  11 0007D        BRB     7$                                      ;
                        50      04   A0  3C 0007F 6$:    MOVZWL  4(R0), R0                               ;
                        6E           50  D0 00083 7$:    MOVL    R0, FLD                                 ;
                50      69           51  C1 00086        ADDL3   R1, (R9), R0                            ; 0559
                    04  AE       02   A0  3C 0008A        MOVZWL  2(R0), FLD+4                            ;
                52      6E       04  AE  C1 0008F        ADDL3   FLD+4, FLD, R2                          ; 0560
                        51       04  AC  D0 00094        MOVL    INPREC, R1                              ;
                        61           52  D1 00098        CMPL    R2, (R1)                                ;
                                     0E  1B 0009B        BLEQU   8$                                      ;
                        0E           60  91 0009D        CMPB    (R0), #14                               ; 0566
                                     78  12 000A0        BNEQ    18$                                     ;
                6E      61       04  AE  C3 000A2        SUBL3   FLD+4, (R1), FLD                        ; 0567
                                     02  18 000A7        BGEQ    8$                                      ; 0568
                                     6E  D4 000A9        CLRL    FLD                                     ;
                    04  AE       04   A1  C0 000AB 8$:    ADDL2   4(R1), FLD+4                            ; 0570
                41      03   A6      04  E1 000B0 9$:    BBC     #4, 3(KFT_PTR), 13$                     ; 0579
                        51     0101  CB  9A 000B5        MOVZBL  257(CA), R1                             ; 0582
                        50           66  3C 000BA        MOVZWL  (KFT_PTR), R0                           ; 0583
                        50       0C  AC  C0 000BD        ADDL2   RESULT, R0                              ;
    06  A6              51       04  BE      6E  2C 000C1        MOVC5   FLD, @FLD+4, R1, 6(KFT_PTR), (R0)       ;
                                         60      000C8                                                   ;
                06      03   A6          06  E0 000C9        BBS     #6, 3(KFT_PTR), 10$                     ; 0584
                        03       58  AB  91 000CE        CMPB    88(CA), #3                              ; 0585
                                     22  1F 000D2        BLSSU   13$                                     ;
                        51           66  3C 000D4 10$:    MOVZWL  (KFT_PTR), R1                           ; 0588
                        51           6E  C0 000D7        ADDL2   FLD, R1                                 ;
                        50           67  3C 000DA        MOVZWL  (R7), R0                                ;
                        51           50  D1 000DD        CMPL    R0, R1                                  ;
                                     03  1E 000E0        BGEQU   11$                                     ;
                        50           51  D0 000E2        MOVL    R1, R0                                  ;
                        67           50  B0 000E5 11$:    MOVW    R0, (R7)                                ;
                        50           5A  3C 000E8        MOVZWL  FIRSTDATA, R0                           ; 0589
                        50           66  B1 000EB        CMPW    (KFT_PTR), R0                           ;
                                     03  1E 000EE        BGEQU   12$                                     ;
                        50           66  3C 000F0        MOVZWL  (KFT_PTR), R0                           ;
                        5A           50  B0 000F3 12$:    MOVW    R0, FIRSTDATA                           ;
                05      03   A6      03  E1 000F6 13$:    BBC     #3, 3(KFT_PTR), 14$                     ; 0596
                        56           08  C0 000FB        ADDL2   #8, KFT_PTR                             ;
                                     F6  11 000FE        BRB     13$                                     ;
                        06       03  A6  E9 00100 14$:    BLBC    3(KFT_PTR), 16$                         ; 0600
                        56           08  C0 00104 15$:    ADDL2   #8, KFT_PTR                             ; 0604
                                 FF12    31 00107        BRW     1$                                      ; 0536
                        67           5A  A2 0010A 16$:    SUBW2   FIRSTDATA, (R7)                         ; 0608
                        03       58  AB  91 0010D        CMPB    88(CA), #3                              ; 0609
                                     03  1F 00111        BLSSU   17$                                     ;
                        67           06  A0 00113        ADDW2   #6, (R7)                                ; 0611
```

```
                    50          01  D0 00116 17$:   MOVL    #1, R0                          ; 0613
                                    04 00119         RET                                     ;
                                50  D4 0011A 18$:   CLRL    R0                              ; 0615
                                    04 0011C         RET                                     ;

; Routine Size:  285 bytes,    Routine Base:  SOR$RO_CODE + 0115
```

```
;  552          0616  1 END
;  553          0617  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  SOR$RO_CODE                      562  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                                 Library Statistics
;
;                                          --------- Symbols ---------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped       Time
;
;  _$255$DUA28:[SORT32.SRC]SORLIB.L32;1      409       104        25          34         00:00.1
;  _$255$DUA28:[SORT32.SRC]SRTSPC.L32;1      120        42        35          12         00:00.1



;                                      COMMAND QUALIFIERS

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

; Size:         562 code + 0 data bytes
; Run Time:       00:14.4
; Elapsed Time:   00:50.1
; Lines/CPU Min:   2570
; Lexemes/CPU-Min: 27333
; Memory Used:  144 pages
; Compilation Complete
```

SRTSPCBLI
LIS

SORSPEC
LIS

SORTRNVEC
LIS

SORSORT
LIS

SORSPCUTI
LIS

SRTSPC
LIS