H 5

```
   SSSSSSSSSSSS     000000000     RRRRRRRRRRRR     TTTTTTTTTTTTTTT     333333333        222222222
   SSSSSSSSSSSS     000000000     RRRRRRRRRRRR     TTTTTTTTTTTTTTT     333333333        222222222
   SSSSSSSSSSSS     000000000     RRRRRRRRRRRR     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         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  RRRRRRRRRRRR           TTT                 333              222
   SSSSSSSSS     000         000  RRRRRRRRRRRR           TTT                 333              222
   SSSSSSSSS     000         000  RRRRRRRRRRRR           TTT                 333              222
            SSS  000         000  RRR   RRR              TTT                    333        222
            SSS  00C         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
SSSSSSSSSSSS        000000000     RRR         RRR        TTT           333333333     222222222222222
SSSSSSSSSSSS        000000000     RRR         RRR        TTT           333333333     222222222222222
SSSSSSSSSSSS        000000000     RRR         RRR        TTT           333333333     222222222222222
```

_$2

Pse

SOR

SOR

SOR

SOR

_LI

```
**FILE**ID**SORSPCUTI                                          L  7
```

```
  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            CCCCCCCC  UUUUUUUUUU      TT        IIIIII      ....
SSSSSSSS      OOOOOO    RR      RR  SSSSSSSS    PP            CCCCCCCC  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
```

```
0 MODULE SOR$SPEC_UTIL (
0               IDENT = 'V04-000'        ! File: SORSPCUTI.B32 Edit: PDG3024
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 !++
1 !
1 ! FACILITY:     VAX-11 SORT/MERGE
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains support routines for specification file features.
1 !
1 ! ENVIRONMENT:  VAX/VMS user mode
1 !
1 ! AUTHOR: Peter D Gilbert, CREATION DATE: 25-Aug-1982
1 !
1 ! MODIFIED BY:
1 !
1 !       T03-015         Original
1 !       T03-016 Put linkages on SOR$$COMPARE.  Rework the way TDT entries are
1 !               processed.  Return 0 if no tie-breaking and the strings collate
1 !               as equal.  PDG 13-Dec-1982
1 !       T03-017 Some fixes in SOR$$COLL_CMP and GET_COLL.  PDG 28-Dec-1982
1 !       T03-018 Add checks for short records in SOR$$TDT and SOR$$REFORM.
1 !               PDG 3-Jan-1983
1 !       T03-019 Remove collating sequence stuff (from this module).
1 !               PDG 26-Jan-1983
1 !       T03-020 Change CH$COPY to use a pad character.  PDG 8-Feb-1983
1 !       T03-021 Use KFT_NDE_SIZ for length in internal node.  PDG 12-Feb-1983
1 !       T03-022 Check the KFT_BUILD flag in SOR$$REFORM.  PDG 10-May-1983
1 !       T03-023 Make CFT_CON_ADR relative.  PDG 25-Jan-1984
1 !       T03-024 WHILE_FAIL_ is now defined in SRTSPC.REQ.  PDG 1-Feb-1984
```

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

```
  1 !       Require files
  1 !
  1 %IF %BLISS(BLISS32) %THEN
  1 REQUIRE 'SRC$:COM';              ! Common definitions for VAX-11 SORT/MERGE
  1 %FI
  1 LIBRARY 'SRC$:SRTSPC';           ! Define symbols for spec file processing
  1
  1 !       Linkage declarations
  1 !
  1 LITERAL
  1     PT_REG = %BLISS16(4) %BLISS32(4),
  1     ST_REG = %BLISS16(3) %BLISS32(3),
  1     XX_REG = 2;                          ! Parameter register
  1
  1 !       Routine declarations
  1 !
  1 FORWARD ROUTINE
  1     SOR$$TDT:            CA_LINKAGE,     ! Evaluate a test
  1     SOR$$RDT:            CA_LINKAGE,     ! Determine record type
  1     SOR$$REFORM:         CA_LINKAGE;     ! Reformat a record
  1
  1 !       Macro declarations
  1 !
L 1 %IF NOT %DECLARED(%QUOTE BASE_)
U 1 %THEN
U 1 MACRO
U 1     BASE_ =     0, 0, 0, 0 %;
  1 %FI
```

```
    1 GLOBAL ROUTINE SOR$$TDT
    1         (
    1         INPREC: REF VECTOR[2],          ! Length/address of input record
    1         TDTPTR: REF TDT_TAB[]           ! Test definitions
L   1 %IF %BLISS(BLISS16)
U   1 %THEN
U   1         COMP:   REF VECTOR[,BYTE],      ! Addr of routine to do simple compares
U   1         FDT:    REF FDT_TAB[],          ! Field definition table
U   1         CFT:    REF CFT_TAB[]           ! Constant definition table
    1 %FI
    1         ):      CA_LINKAGE =
    1 !++
    1 !
    1 ! FUNCTIONAL DESCRIPTION:
    1 !
    1 !       This routine evaluates a test.
    1 !
    1 ! FORMAL PARAMETERS:
    1 !
    1 !       As described above.
    1 !
    1 !       Note that COMP, FDT, and CFT could be bound to the locations
    1 !       in the context area that hold the addresses.
    1 !
    1 ! IMPLICIT INPUTS.
    1 !
    1 !       NONE
    1 !
    1 ! IMPLICIT OUTPUTS:
    1 !
    1 !       NONE
    1 !
    1 ! ROUTINE VALUE:
    1 !
    1 !       0       indicates test failed
    1 !       1       indicates test passed
    1 !
    1 ! SIDE EFFECTS:
    1 !
    1 !       NONE
    1 !
    1 !--
    2     BEGIN
    2     LOCAL
    2         TDT:    REF TDT_TAB[],  ! Local pointer to test descriptions
    2         RES;                    ! Running total/result
    2
    2     CA_AREA_(CA);
    2
    3     BEGIN
L   3 %IF %BLISS(BLISS32)
    3 %THEN
    3     BIND
    3         FDT = CA[CA_FDT_ADR]: REF FDT_TAB[],    ! Field definition table
    3         CFT = CA[CA_CFT_ADR]: REF CFT_TAB[];    ! Constant definition table
    3     EXTERNAL ROUTINE
    3         SOR$$COMPARE:   CA_LINKAGE;
    BIND
        COMP = SOR$$COMPARE: VECTOR[,BYTE];     ! Addr of comparison routine
%FI

    !+
    ! The test definition table consists of simple comparisons, and a flag
    ! indicating whether the result of this comparison should be ANDed or
    ! ORed with the running total/result.  Another flag indicates whether
    ! this is the last simple comparison (there is always at least one).
    !
    ! For example, the following:
    !
    ! CONTINUE  CMP1 XXX
    ! CONTINUE  CMP2 OP2
    !           CMP3 OP3
    !
    ! corresponds to the condition: ((CMP1 OP2 CMP2) OP3 CMP3)
    !-


    ! Get the address of the first test entry
    !
    TDT = TDTPTR[0, BASE_];

    WHILE 1 DO
        BEGIN
        LOCAL
            FDT_IX,                 ! Index into FDT (or CFT) table
            TYPE,                   ! Data type of the operands
            FLD1: VECTOR[2],        ! Length/address of first operand
            FLD2: VECTOR[2];        ! Length/address of first operand

        IF .TDT[0,TDT_TRUE] THEN RETURN 1;

        !+
        ! Find the datatypes, lengths, and addresses of the fields/data
        ! to be compared.
        !-

        ! The first field is always a field (not a constant)
        !
        FDT_IX = .TDT[0,TDT_FLD_ONE];
        TYPE = .FDT[.FDT_IX, FDT_TYPE];
        FLD1[0] = .FDT[.FDT_IX, FDT_FLD_SIZ];
        FLD1[1] = .FDT[.FDT_IX, FDT_FLD_POS];
        IF .FLD1[0] + .FLD1[1] GTRU .INPREC[0]
        THEN
            BEGIN
            !
            ! If this is not a text field, it's an error
            !
            IF .TYPE NEQ DT_T THEN RETURN 0;
            FLD1[0] = .INPREC[0] - .FLD1[1];
            IF .FLD1[0] LSS 0 THEN FLD1[0] = 0;
            END;
        FLD1[1] = .FLD1[1] + .INPREC[1];

```

```
;  4          ! The second field may be a field or a constant
;  4          !
;  4          FDT_IX = .TDT[0,TDT_FLD_TWO];
;  4          IF .TDT[0,TDT_CONSTANT]
;  4          THEN
;  5              BEGIN
;  5              FLD2[0] = .CFT[.FDT_IX, CFT_CON_LEN];
;  5              FLD2[1] = .CFT[.FDT_IX, CFT_CON_ADR] + CFT[0, BASE_];
;  5              END
;  4          ELSE
;  5              BEGIN
;  5              FLD2[0] = .FDT[.FDT_IX, FDT_FLD_SIZ];
;  5              FLD2[1] = .FDT[.FDT_IX, FDT_FLD_POS];
;  5              IF .FLD2[0] + .FLD2[1] GTRU .INPREC[0]
;  5              THEN
;  6                  BEGIN
;  6                  !
;  6                  ! If this is not a text field, it's an error
;  6                  !
;  6                  IF .FDT[.FDT_IX, FDT_TYPE] NEQ DT_T THEN RETURN 0;
;  6                  FLD2[0] = .INPREC[0] - .FLD2[1];
;  6                  IF .FLD2[0] LSS 0 THEN FLD2[0] = 0;
;  5                  END;
;  5              FLD2[1] = .FLD2[1] + .INPREC[1];
;  4              END;
;  4
;  4          IF
;  5              BEGIN
;  5              LOCAL CMP;
;  5
;  5              CMP = CA_LINKAGE(COMP[0], .TYPE, FLD1[0], FLD2[0]);
;  5
;  5              SELECTONE .CMP OF
;  5                  SET
;  5                  [0]:     .TDT[0,TDT_EQL];
;  5                  [+1]:    .TDT[0,TDT_GTR];
;  5                  [-1]:    .TDT[0,TDT_LSS];
;  5                  [OTHERWISE]:     RETURN .CMP;     ! Unexpected
;  5                  TES
;  5              END
;  4          THEN
;  5              BEGIN
;  5              LOCAL X;
;  5              X = .TDT[0,TDT_GOTO];
;  5              IF .X EQL 0 THEN RETURN 0;
;  5              TDT = TDT[.X, BASE_];          ! Goto new test item
;  5              END
;  4          ELSE
;  4              TDT = TDT[1, BASE_];           ! Advance to next test
;  4
;  3          END;
;  3
;  3      RETURN 0;
;  2      END;
;  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$:      ADDL2   4(R3), FLD1+4                                  ; 0268
                          50      02     A2   9A 00050          MOVZBL  2(TDT), FDT_IX                                 ; 0272
                          50             06   C4 00054          MULL2   #6, R0                                         ; 0276
          12              62             04   E1 00057          BBC     #4, (TDT), 4$                                  ; 0273
                          50    010C     CB   C0 0005B          ADDL2   268(CA), R0                                    ; 0276
                          6E             60   9A 00060          MOVZBL  (R0), FLD2                                     ;
    04    AE     02       A0    010C     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             01   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
```

SOR
V04

```
;  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
```

```
  1 GLOBAL ROUTINE SOR$$REFORM
  1         (
  1         INPREC: REF VECTOR[2],          ! Length/address of input record
  1         KFTPTR: REF KFT_TAB[],
L 1 %IF %BLISS(BLISS16)
U 1 %THEN
U 1         RDT:    REF RDT_TAB[],          ! Record definition table
U 1         COMP:   REF VECTOR[,BYTE],      ! Addr of routine to do simple compares
U 1         TDT:    REF TDT_TAB[],          ! Test definition table
U 1         FDT:    REF FDT_TAB[],          ! Field definition table
U 1         CFT:    REF CFT_TAB[],          ! Constant definition table
  1 %FI
  1         RESULT: REF VECTOR[,BYTE],      ! Address of output area
  1         RECLEN: REF VECTOR[1,WORD]      ! Output format record length
  1         ):      CA_LINKAGE =
  1 !++
  1 !
  1 ! FUNCTIONAL DESCRIPTION:
  1 !
  1 !       This routine reformats a record into a result area.
  1 !
  1 ! FORMAL PARAMETERS:
  1 !
  1 !       As described above.
  1 !
  1 !       Note that RDT, COMP, TDT, FDT, and CFT could be bound to the locations
  1 !       in the context area that hold the addresses.
  1 !
  1 ! IMPLICIT INPUTS:
  1 !
  1 !       NONE
  1 !
  1 ! IMPLICIT OUTPUTS:
  1 !
  1 !       NONE
  1 !
  1 ! ROUTINE VALUE:
  1 !
  1 !       1       indicates everything went okay.
  1 !
  1 ! SIDE EFFECTS:
  1 !
  1 !       NONE
  1 !
  1 ! NOTES:
  1 !
  1 !       The protocol for using this routine is as follows:
  1 !
  1 !       Z = SOR$$RDT( INPREC, ..., RDTPTR )
  1 !       SELECTONE .Z OF
  1 !               SET
  1 !               [0]:    ...omit the record...;
  1 !               [1]:    BEGIN
  1 !                       KFT_IX = .RDTPTR[0, RDT_KFT_IDX];
  1 !                       Z = SOR$$REFORM( INPREC, KFT[.KFT_IX,BASE_], ... );
  1 !                       IF .Z NEQ 1 THEN ...error from comparison...;
  1 !                       END;
  1 !               [OTHERWISE]: ...error from comparison...;
  1 !               TES;
  1 !--
  2      BEGIN
  2      LOCAL
  2          KFT_PTR: REF KFT_TAB[],          ! Local pointer to KFT table
  2          FIRSTDATA:      WORD;            ! Offset to first data field
  2
  2      CA_AREA_(CA);
  2
  3      BEGIN
L 3 %IF %BLISS(BLISS32)
  3 %THEN
  3      BIND
  3          RDT = CA[CA_RDT_ADR]: REF RDT_TAB[],    ! Record definition table
  3          TDT = CA[CA_TDT_ADR]: REF TDT_TAB[],    ! Test definition table
  3          FDT = CA[CA_FDT_ADR]: REF FDT_TAB[],    ! Field definition table
  3          CFT = CA[CA_CFT_ADR]: REF CFT_TAB[];    ! Constant definition table
  3      EXTERNAL ROUTINE
  3          SOR$$COMPARE:   CA_LINKAGE;
  3      BIND
  3          COMP = SOR$$COMPARE: VECTOR[,BYTE];     ! Addr of comparison routine
  3 %FI
  3
  3      !+
  3      ! The key/data field definition table consists of field definitions,
  3      ! which define the fields in a record; some of these may be conditional
  3      ! data fields.  As an example, the following specification file:
  3      !
  3      !   /DATA=FLD1
  3      !   /DATA=( IF COND1 THEN CONST1
  3      !           IF COND2 THEN CONST2
  3      !           ELSE CONST3 )
  3      !   /DATA=FLD3
  3      !   /DATA=FLD4
  3      !
  3      ! corresponds to the following field definition table entries:
  3      !
  3      ! CONTINUE  FLD1
  3      ! CONTINUE  CONST1  COND    COND1
  3      ! CONTINUE  CONST2  COND    COND2
  3      ! CONTINUE  CONST3
  3      ! CONTINUE  FLD3
  3      !           FLD4
  3      !
  3      ! Note that the ELSE part of the conditional data definition does not
  3      ! have the COND flag set.
  3      !-
  3
  3
  3      ! Initialize the local pointer to the KFT table
  3      ! Initialize the output format record length
  3      !
  3      KFT_PTR = KFTPTR[0,BASE_];
  3      RECLEN[0] = 0;
  3      FIRSTDATA = -1;
  3
  3
  3     ! While there are more fields
  3     !
  3     WHILE 1 DO
  4         BEGIN
  4         LOCAL
  4             FLD: VECTOR[2],         ! Length/address of field or constant
  4             FDT_IX;                 ! Index into FDT (or CFT) table
  4
  4         ! Advance KFT_PTR until we find a test that passes
  4         !
  4         WHILE_FAIL_('KFT');
  4
  4         ! Determine whether we should grab the field from the record
  4         ! or from the constant table.
  4         !
  4         FDT_IX = .KFT_PTR[0,KFT_FDT_IDX];
  4         IF .KFT_PTR[0,KFT_CONSTANT]
  4         THEN
  5             BEGIN
  5             FLD[0] = .CFT[.FDT_IX, CFT_CON_LEN];
  5             FLD[1] = .CFT[.FDT_IX, CFT_CON_ADR] + CFT[0, BASE_];
  5             END
  4         ELSE
  5             BEGIN
  5             FLD[0] = KFT_UNITS_(KFT_PTR);       ! Get size in bytes
  5             FLD[1] = .FDT[.FDT_IX, FDT_FLD_POS];
  5             IF .FLD[0] + .FLD[1] GTRU .INPREC[0]
  5             THEN
  6                 BEGIN
  6                 !
  6                 ! If this is not a text field, it's an error
  6                 !
  6                 IF .FDT[.FDT_IX, FDT_TYPE] NEQ DT_T THEN RETURN 0;
  6                 FLD[0] = .INPREC[0] - .FLD[1];
  6                 IF .FLD[0] LSS 0 THEN FLD[0] = 0;
  5                 END;
  5             FLD[1] = .FLD[1] + .INPREC[1];
  4             END;
  4
L 4         %IF MAX(TYP_K_RECORD,TYP_K_TAG) GEQ MIN(TYP_K_INDEX,TYP_K_ADDRESS)
U 4         %THEN
  4             %ERROR('The following test won''t work') %FI
  4
  4         ! Copy the field to its place in the internal format record
  4         !
  4         IF .KFT_PTR[0,KFT_BUILD]
  4         THEN
  5             BEGIN
  5             CH$COPY(.FLD[0], .FLD[1], .CA[CA_PAD],
  5                 .KFT_PTR[0,KFT_NDE_SIZ], RESULT[.KFT_PTR[0,KFT_NDE_POS]]);
  5             IF .KFT_PTR[0,KFT_DATA]
  5                 OR .CA[CA_PROCESS] GEQ MIN(TYP_K_INDEX,TYP_K_ADDRESS)
  5             THEN
  6                 BEGIN
  6                 RECLEN[0] = MAXU(.RECLEN[0], .KFT_PTR[0,KFT_NDE_POS]+.FLD[0]);
  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
```

0366 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SRTSPCBLI
LIS

SORSPEC
LIS

SORTRNVEC
LIS

SORSORT
LIS

SORSPCUTI
LIS

SRTSPC
LIS