```
   SSSSSSSSSSS     000000000     RRRRRRRRRRR     TTTTTTTTTTTTTTT     333333333       222222222
   SSSSSSSSSSS     000000000     RRRRRRRRRRRR    TTTTTTTTTTTTTTT     333333333       222222222
   SSSSSSSSSSS     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  RRRRRRRRRRR            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  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
            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
```

```
  SSSSSSSS    000000    RRRRRRRR      CCCCCCCC    000000    LL          LL            AAAAAA    TTTTTTTTTT
  SSSSSSSS    000000    RRRRRRRR      CCCCCCCC    000000    LL          LL            AAAAAA    TTTTTTTTTT
SS          00      00  RR      RR  CC          00      00  LL          LL          AA      AA      TT
SS          00      00  RR      RR  CC          00      00  LL          LL          AA      AA      TT
SS          00      00  RR      RR  CC          00      00  LL          LL          AA      AA      TT
SS          00      00  RR      RR  CC          00      00  LL          LL          AA      AA      TT
  SSSSSS    00      00  RRRRRRRR    CC          00      00  LL          LL          AA      AA      TT
  SSSSSS    00      00  RRRRRRRR    CC          00      00  LL          LL          AA      AA      TT
        SS  00      00  RR  RR      CC          00      00  LL          LL          AAAAAAAAAA      TT
        SS  00      00  RR  RR      CC          00      00  LL          LL          AAAAAAAAAA      TT
        SS  00      00  RR    RR    CC          00      00  LL          LL          AA      AA      TT          ....
        SS  00      00  RR    RR    CC          00      00  LL          LL          AA      AA      TT          ....
SSSSSSSS      000000    RR      RR    CCCCCCCC    000000    LLLLLLLLLL  LLLLLLLLLL  AA      AA      TT          ....
SSSSSSSS      000000    RR      RR    CCCCCCCC    000000    LLLLLLLLLL  LLLLLLLLLL  AA      AA      TT          ....

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
```

```
0000     1          .TITLE  SOR$COLLATE             Compare under collating sequence
0000     2          .IDENT  'V04-000'               ; File: SORCOLLAT.MAR Edit: PDG022
0000     3          .SBTTL  Copyright Notice
0000     4 ;
0000     5 ;**********************************************************************
0000     6 ;*                                                                    *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                           *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.            *
0000     9 ;*  ALL RIGHTS RESERVED.                                              *
0000    10 ;*                                                                    *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
0000    16 ;*  TRANSFERRED.                                                      *
0000    17 ;*                                                                    *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
0000    20 ;*  CORPORATION.                                                      *
0000    21 ;*                                                                    *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.           *
0000    24 ;*                                                                    *
0000    25 ;*                                                                    *
0000    26 ;**********************************************************************
0000    27 ;
```

```
0000     29          .SBTTL  Program description
0000     30 ;++
0000     31 ;
0000     32 ; FACILITY:
0000     33 ;
0000     34 ;       Support for compare under influence of collating sequences.
0000     35 ;
0000     36 ; ABSTRACT:
0000     37 ;
0000     38 ;       NONE
0000     39 ;
0000     40 ; ENVIRONMENT:
0000     41 ;
0000     42 ;       Native mode, User mode, AST reentrant
0000     43 ;
0000     44 ; AUTHOR:
0000     45 ;
0000     46 ;       Peter D Gilbert, January 1983
0000     47 ;
0000     48 ; MODIFIED BY:
0000     49 ;
0000     50 ;       T03-015         Original
0000     51 ;       T03-016 Support ignored pad characters.  PDG 28-Jan-1983
0000     52 ;       T03-017 Fix errors with PUSHR and POPR order.  PDG 1-Feb-1983
0000     53 ;       T03-018 Add XORB comments.  PDG 7-Mar-1983
0000     54 ;       T03-019 Reverse sense of the tie-breaking CMPC.  PDG 22-Mar-1983
0000     55 ;       T03-020 Make tie-break CMPC reversable at run-time.  PDG 12-Apr-1983
0000     56 ;       T03-021 Note that R5 is preserved.  PDG 11-Jul-1983
0000     57 ;       T03-022 Remove extra store of stable point.  Add "short-cut" entry
0000     58 ;               points.  PDG 17-Oct-1983
0000     59 ;--
```

```
                        .DSABL  GLOBAL          ; Externals must be explicitly declared
                        $PSWDEF
                        .PSECT  SOR$RO_CODE     NOVEC,NOWRT,RD,EXE,SHR,LCL,REL,CON,PIC,LONG
                ;+
                ;
                ; These routinse do comparisons under the influence of a collating sequence.
                ;
                ; Inputs:
                ;   R0  Length remaining in string1
                ;   R1  Address of source1
                ;   R2  Length remaining in string2
                ;   R3  Address of source2
                ;   R5  Address of tables
                ;   0(SP)       Return address
                ;
                ; Outputs:
                ;   R0  -1, 0, +1
                ;   R1 thru R4  Garbage
                ;   R5  Preserved
                ;   R9  Preserved
                ;   Condition codes as a result of MOVL R0, R0
                ;   The return address is removed from the stack.
                ;
                ;--
```

```
                                                B 10
0000     87          ; Fetch a character.
0000     88          ;
0000     89          .MACRO  FETCHSTR1, ?LAB1, ?LAB2
0000     90          SOBGEQ  R0, LAB1                ; Any characters left?
0000     91          CLRL    R0                      ; Make this zero again
0000     92          MOVZBL  B^RES$PAD(R5), R4       ; Use the pad character
0000     93          BRB     LAB2
0000     94 LAB1:    MOVZBL  (R1)+, R4               ; Fetch the character
0000     95 LAB2:    .ENDM
0000     96          .MACRO  FETCHSTR2, ?LAB1, ?LAB2
0000     97          SOBGEQ  R2, LAB1                ; Any characters left?
0000     98          CLRL    R2                      ; Make this zero again
0000     99          MOVZBL  B^RES$PAD(R5), R9       ; Use the pad character
0000    100          BRB     LAB2
0000    101 LAB1:    MOVZBL  (R3)+, R9               ; Fetch the character
0000    102 LAB2:    .ENDM
0000    103
0000    104          ; Fetch a character and look up its value in the primary table.
0000    105          ; If we were padding, and the pad character is ignored, leave the
0000    106          ; collating value as zero, so that this string will compare less than
0000    107          ; any non-ignored characters in the other string.
0000    108          ; If both string are being padded with ignored pad characters, the code
0000    109          ; branches back to the CMPC instruction, which compares equal, and the
0000    110          ; tie-break information will then be used.
0000    111          ;
0000    112          ; When we have the collating value, either fall through, or branch to
0000    113          ; the first parameter.
0000    114          ; The second parameter labels the call to SPECSTR or SPECDBL, so that
0000    115          ; other code can branch to it.
0000    116          ;
0000    117          .MACRO  FETCHSTR1NP, RESLAB, ?LAB1
0000    118          SOBGEQ  R0, LAB1                ; Any characters left?
0000    119          CLRL    R0                      ; Make this zero again
0000    120          MOVZBL  B^RES$PAD(R5), R4       ; Use the pad character
0000    121          MOVZBL  B^RES$PTAB(R5)[R4], R4  ; Fetch the collating value
0000    122          BRB     RESLAB                  ; Use this value, regardless
0000    123 LAB1:    MOVZBL  (R1)+, R4               ; Fetch the character
0000    124          MOVZBL  B^RES$PTAB(R5)[R4], R4  ; Check that it's not special
0000    125          BNEQ    RESLAB
0000    126          .ENDM
0000    127          .MACRO  FETCHSTR2NP, RESLAB, ?LAB1
0000    128          SOBGEQ  R2, LAB1                ; Any characters left?
0000    129          CLRL    R2                      ; Make this zero again
0000    130          MOVZBL  B^RES$PAD(R5), R9       ; Use the pad character
0000    131          MOVZBL  B^RES$PTAB(R5)[R9], R9  ; Fetch the collating value
0000    132          BRB     RESLAB                  ; Use this value, regardless
0000    133 LAB1:    MOVZBL  (R3)+, R9               ; Fetch the character
0000    134          MOVZBL  B^RES$PTAB(R5)[R9], R9  ; Check that it's not special
0000    135          BNEQ    RESLAB
0000    136          .ENDM
0000    137
0000    138          .MACRO  SPECSTR1, ?LAB1, ?LAB2, ?LAB3
0000    139          MOVAB   W^RES$STAB(R5), R4
0000    140          BRB     LAB2
0000    141 LAB1:    ADDL2   #4, R4
0000    142 LAB2:    CMPB    (R4), -1(R1)
0000    143          BLSSU   LAB1
```

```
0000   144          BEQL    LAB3
0000   145          MOVAL   #0, R4
0000   146 LAB3:    MOVZWL  2(R4), R4
0000   147          .ENDM
0000   148          .MACRO  SPECSTR2, ?LAB1, ?LAB2, ?LAB3
0000   149          MOVAB   W^RES$STAB(R5), R9
0000   150          BRB     LAB2
0000   151 LAB1:    ADDL2   #4, R9
0000   152 LAB2:    CMPB    (R9), -1(R3)
0000   153          BLSSU   LAB1
0000   154          BEQL    LAB3
0000   155          MOVAL   #0, R9
0000   156 LAB3:    MOVZWL  2(R9), R9
0000   157          .ENDM
0000   158
0000   159          .MACRO  SPECDBL1, ?LAB1, ?LAB2, ?LAB3, ?LAB4
0000   160          ;
0000   161          ; This code assumes that the length of the original string is non-zero.
0000   162          ;
0000   163          MOVAB   W^RES$STAB(R5), R4      ; Get the table address
0000   164          CLRL    -(SP)                   ; Assume it is ignored
0000   165          BRB     LAB2
0000   166 LAB1:    ADDL2   #4, R4                  ; Advance past the collating value
0000   167 LAB2:    CMPB    (R4), -1(R1)            ; Does this first character match?
0000   168          BLSSU   LAB1
0000   169          BGTRU   LAB4
0000   170          ADDL2   #2, R4
0000   171          MOVW    (R4), (SP)              ; It matched.  Save the collating value
0000   172          TSTW    R0
0000   173          BEQL    LAB4                    ; Any more chars in the string?
0000   174 LAB3:    ADDL2   #2, R4
0000   175          CMPW    (R4)+, -1(R1)           ; Do the two characters match?
0000   176          BLSSU   LAB3
0000   177          BGTRU   LAB4
0000   178          TSTW    2(R4)
0000   179          BEQL    LAB4                    ; Branch if we found the trailing stuff
0000   180          MOVW    (R4), (SP)              ; Copy the collating value
0000   181          DECL    R0                      ; Advance the string
0000   182          INCL    R1
0000   183 LAB4:    MOVL    (SP)+, R4               ; Put the collating value in R4
0000   184          .ENDM
0000   185          .MACRO  SPECDBL2, ?LAB1, ?LAB2, ?LAB3, ?LAB4
0000   186          MOVAB   W^RES$STAB(R5), R9
0000   187          CLRL    -(SP)
0000   188          BRB     LAB2
0000   189 LAB1:    ADDL2   #4, R9
0000   190 LAB2:    CMPB    (R9), -1(R3)
0000   191          BLSSU   LAB1
0000   192          BGTRU   LAB4
0000   193          ADDL2   #2, R9
0000   194          MOVW    (R9), (SP)
0000   195          TSTW    R2
0000   196          BEQL    LAB4
0000   197 LAB3:    ADDL2   #2, R9
0000   198          CMPW    (R9)+, -1(R3)
0000   199          BLSSU   LAB3
0000   200          BGTRU   LAB4
```

```
0000    201          TSTW    2(R9)
0000    202          BEQL    LAB4
0000    203          MOVW    (R9), (SP)
0000    204          DECL    R2
0000    205          INCL    R3
0000    206 LAB4:    MOVL    (SP)+, R9
0000    207          .ENDM
0000    208
0000    209          ; Define offsets
0000    210          ;
0000    211          .EXTRN  RES$PTAB                ; Address of the primary table
0000    212          .EXTRN  RES$UPPER               ; Address of the upper table
0000    213          .EXTRN  RES$TB                  ; Tie-break PSW
0000    214          .EXTRN  RES$REVERSE             ; Reverse sense of tie-break CMPC
0000    215          .EXTRN  RES$PAD                 ; Pad character
0000    216          .EXTRN  RES$STAB                ; Address of secondary table
```

```
                                                E 10

                                0000   218 ;
                                0000   219 ; SOR$$COLLATE_0
                                0000   220 ;
                                0000   221 ;   This routine assumes/supports:
                                0000   222 ;       No ignored characters
                                0000   223 ;       No double characters
                                0000   224 ;       No double collating values
                                0000   225 ;       Upcase table
                                0000   226 ;
                                0000   227 ; Registers:
                                0000   228 ;       R0-R4   Nopreserve
                                0000   229 ;       R5      Preserve
                                0000   230 ;       R6-R8   Not used
                                0000   231 ;       R9-R10  Preserve
                                0000   232 ;       R11     Preserve
                                0000   233 ;
                                0000   234 SOR$$COLLATE_0_A::
                     0200 8F    BB  0000   235         PUSHR   #^M<R9>
                          7E    DC  0004   236         MOVPSL  -(SP)
                          10    12  0006   237         BNEQ    JOIN_0
                          58    11  0008   238         BRB     USETB
                                    000A   239 SOR$$COLLATE_0::
                          7E    59  D0  000A   240         MOVL    R9, -(SP)                       ; Save R9
    63    52    00'A5    61    50  2D  000D   241         CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                          7E    DC  0014   242         MOVPSL  -(SP)                           ; Save the tie-break info
                          4A    13  0016   243         BEQL    USETB                           ; An optimization
                     6E   00'A5 8C  0018   244 JOIN_0: XORB2   B^RES$REVERSE(R5), (SP) ; Reverse sense of this compare
                     6E   00'A5 88  001C   245         BISB2   B^RES$TB(R5), (SP)      ; Tie-break adjustment
                                0020   246 0$:     ;
                                0020   247         ; Fetch the characters from the strings.
                                0020   248         ;
                                0020   249         FETCHSTR1
                                002E   250         FETCHSTR2
                                003C   251         ;
                                003C   252         ; We know they differ.
                                003C   253         ; Are they different cases of the same char?
                                003C   254         ;
                   10 6E    01  E0  003C   255         BBS     #PSW$V_V, (SP), 16$
        0000'C549   0000'C544   91  0040   256         CMPB    W^RES$UPPER(R5)[R4], W^RES$UPPER(R5)[R9]
                          05    13  0049   257         BEQL    16$                             ; Yes, the binary compare is okay
                          6E    DC  004B   258         MOVPSL  (SP)                            ; No, use this compare
                     6E   02    C8  004D   259         BISL2   #PSW$M_V, (SP)                  ; Don't do this again
                                0050   260 16$:    ;
                                0050   261         ; Compare the collating values of the characters.
                                0050   262         ;
             00'A549    00'A544   91  0050   263         CMPB    B^RES$PTAB(R5)[R4], B^RES$PTAB(R5)[R9]
                          18    12  0057   264         BNEQ    POPNE
    63    52    00'A5    61    50  2D  0059   265         CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                          BE    12  0060   266         BNEQ    0$                              ; If equal, then R0 = 0
                     0201 8F    BA  0062   267 USETB:   POPR    #^M<R0,R9>                      ; Pop PSL, Restore R9
                   11 50    02  E0  0066   268 USETB1:  BBS     #PSW$V_Z, R0, EQ                ; Zero?
                        0A  50  E9  006A   269         BLBC    R0, GT                          ; Like BGTRU, since PSW$V_C = 0
                     50   01    CE  006D   270 LT:      MNEGL   #1, R0                          ; Return -1
                          05    05  0070   271         RSB
                     0201 8F    BA  0071   272 POPNE:   POPR    #^M<R0,R9>                      ; Pop PSL, restore R9
                          F6    1B  0075   273 NE:      BLEQU   LT
                     50   01    D0  0077   274 GT:      MOVL    #1, R0                          ; Return +1
```

```
                05  007A   275             RSB
50              D4  007B   276 EQ:         CLRL    R0
                05  007D   277             RSB
```

```
                                007E    279 ;
                                007E    280 ; SOR$$COLLATE_1
                                007E    281 ;
                                007E    282 ;   This routine assumes/supports:
                                007E    283 ;        Ignored characters
                                007E    284 ;        No double characters
                                007E    285 ;        Double collating values
                                007E    286 ;        Upcase table
                                007E    287 ;        Ignored pad characters
                                007E    288 ;
                                007E    289 ; Registers:
                                007E    290 ;        R0-R4   Nopreserve
                                007E    291 ;        R5      Preserve
                                007E    292 ;        R6-R8   Not used
                                007E    293 ;        R9-R10  Preserve
                                007E    294 ;        R11     Preserve
                                007E    295 ;
                                007E    296         .ENABL  LSB
                                007E    297 SOR$$COLLATE_1_A::
                    0200 8F     BB  007E    298         PUSHR   #^M<R9>
                         7E     DC  0082    299         MOVPSL  -(SP)
                         10     12  0084    300         BNEQ    JOIN_1
                         DA     11  0086    301         BRB     USETB
                                    0088    302 SOR$$COLLATE_1::
                         7E  59 D0  0088    303         MOVL    R9, -(SP)                       ; Save R9
    63   52   00'A5    61  50 2D  008B    304         CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                         7E     DC  0092    305         MOVPSL  -(SP)                           ; Save the tie-break info
                         CC     13  0094    306         BEQL    USETB                           ; An optimization
                  6E  00'A5     8C  0096    307 JOIN_1: XORB2   B^RES$REVERSE(R5), (SP)         ; Reverse sense of this compare
                  6E  00'A5     88  009A    308         BISB2   B^RES$TB(R5), (SP)              ; Tie-break adjustment
                                    009E    309 0$:     ;
                                    009E    310         ; Fetch the characters from strings
                                    009E    311         ;
                                    009E    312         FETCHSTR1
                                    00AC    313         FETCHSTR2
                                    00BA    314         ;
                                    00BA    315         ; We know they differ.
                                    00BA    316         ; Are they different cases of the same char?
                                    00BA    317         ;
                 10 6E    01     E0  00BA    318         BBS     #PSW$V_V, (SP), 16$
      0000'C549   0000'C544      91  00BE    319         CMPB    W^RES$UPPER(R5)[R4], W^RES$UPPER(R5)[R9]
                         05     13  00C7    320         BEQL    16$                             ; Yes, the binary compare is okay
                         6E     DC  00C9    321         MOVPSL  (SP)                            ; No, use this compare
                      6E 02     C8  00CB    322         BISL2   #PSW$M_V, (SP)                  ; Don't do this again
                                    00CE    323 16$:    ;
                                    00CE    324         ; Fetch the collating values of the characters.
                                    00CE    325         ;
                   54   00'A544     9A  00CE    326         MOVZBL  B^RES$PTAB(R5)[R4], R4
                               18   13  00D3    327         BEQL    101$
                   59   00'A549     9A  00D5    328         MOVZBL  B^RES$PTAB(R5)[R9], R9
                               5A   13  00DA    329         BEQL    202$
                      59       54   91  00DC    330         CMPB    R4, R9
                               90   12  00DF    331 150$:   BNEQ    POPNE
    63   52   00'A5    61       50   2D  00E1    332 100$:   CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                               B4   12  00E8    333         BNEQ    0$                          ; If equal, then R0 = 0
                            FF75    31  00EA    334         BRW     USETB
                   59   00'A549     9A  00ED    335 101$:   MOVZBL  B^RES$PTAB(R5)[R9], R9
```

```
                   21   11  00F2   336           BRB     112$
   59    59    F8  8F   78  00F4   337 402$:     ASHL    #-8, R9, R9
                   E6   13  00F9   338           BEQL    100$
                            00FB   339 111$:     FETCHSTR1NP       113$
                            0115   340 112$:     SPECSTR1
                   C7   13  0132   341           BEQL    111$
                   59   95  0134   342 113$:     TSTB    R9
                   2D   13  0136   343 202$:     BEQL    201$
             59    54   91  0138   344 301$:     CMPB    R4, R9
                   A2   12  013B   345           BNEQ    150$
   54    54    F8  8F   78  013D   346           ASHL    #-8, R4, R4
                   B0   13  0142   347           BEQL    402$
   59    59    F8  8F   78  0144   348           ASHL    #-8, R9, R9
                   ED   12  0149   349           BNEQ    301$
                            014B   350 211$:     FETCHSTR2NP       301$
                            0165   351 201$:     SPECSTR2
                   C7   13  0182   352           BEQL    211$
                   B2   11  0184   353           BRB     301$
                            0186   354           .DSABL  LSB
```

```
                                        0186   356 ;
                                        0186   357 ; SOR$$COLLATE_2
                                        0186   358 ;
                                        0186   359 ;   This routine assumes/supports:
                                        0186   360 ;        Ignored characters
                                        0186   361 ;        Double characters
                                        0186   362 ;        Double collating values
                                        0186   363 ;        Upcase table
                                        0186   364 ;        Ignored pad characters
                                        0186   365 ;
                                        0186   366 ; Registers:
                                        0186   367 ;        R0-R4   Nopreserve
                                        0186   368 ;        R5      Preserve
                                        0186   369 ;        R6-R8   Not used
                                        0186   370 ;        R9-R10  Preserve
                                        0186   371 ;        R11     Preserve
                                        0186   372 ;
                                        0186   373 ; Restrictions:
                                        0186   374 ;        The following are not handled correctly:
                                        0186   375 ;            Double characters that use the pad character.
                                        0186   376 ;                (when this happens at the end of a string)
                                        0186   377 ;
                                        0186   378          .ENABL  LSB
                                        0186   379 SOR$$COLLATE_2::
                0202 8F     BB          0186   380          PUSHR   #^M<R1,R9>                  ; Save R9, save current stable point
    63  52  00'A5  61  50   2D          018A   381          CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                        7E  DC          0191   382          MOVPSL  -(SP)                       ; Save the tie-break info
                        7A  13          0193   383          BEQL    89$                         ; An optimization
                6E  00'A5   8C          0195   384          XORB2   B^RES$REVERSE(R5), (SP)     ; Reverse sense of this compare
                6E  00'A5   88          0199   385          BISB2   B^RES$TB(R5), (SP)          ; Tie-break adjustment
                            019D   386 0$:      ;
                            019D   387          ; Backup to some stable point
                            019D   388          ;
                59  51      D0      019D   389          MOVL    R1, R9
            04  AE  51      D1      01A0   390 2$:      CMPL    R1, 4(SP)
                    0B      13      01A4   391          BEQL    5$
                54  71      9A      01A6   392          MOVZBL  -(R1), R4
                00'A544     95      01A9   393          TSTB    B^RES$PTAB(R5)[R4]          ; Might this be start of a double char?
                    F1      13      01AD   394          BEQL    2$
                    51      D6      01AF   395          INCL    R1
                59  51      C2      01B1   396 5$:      SUBL2   R1, R9
                    09      13      01B4   397          BEQL    8$                          , ONLY an optimization
                50  59      C0      01B6   398          ADDL2   R9, R0
                52  59      C0      01B9   399          ADDL2   R9, R2
                53  59      C2      01BC   400          SUBL2   R9, R3
                            01BF   401          ;
                            01BF   402          ; Fetch the characters from strings
                            01BF   403          ;
                            01BF   404 8$:      FETCHSTR1
                            01CD   405          FETCHSTR2
                            01DB   406          ;
                            01DB   407          ; They may differ.
                            01DB   408          ; Are they different cases of the same char?
                            01DB   409          ;
            10  6E      01  E0      01DB   410          BBS     #PSW$V_V, (SP), 16$
0000'C549   0000'C544       91      01DF   411          CMPB    W^RES$UPPER(R5)[R4], W^RES$UPPER(R5)[R9]
                        05  13      01E8   412          BEQL    16$                         ; Yes, the binary compare is okay
```

```
                                 6E   DC  01EA   413          MOVPSL  (SP)                        ; No, use this compare
                            6E   02   C8  01EC   414          BISL2   #PSW$M_V, (SP)              ; Don't do this again
                                      01EF   415 16$:       ;
                                      01EF   416            ; Fetch the collating values of the characters.
                                      01EF   417            ;
                       54   00'A544   9A  01EF   418          MOVZBL  B^RES$PTAB(R5)[R4], R4
                                 2A   13  01F4   419          BEQL    101$
                       59   00'A549   9A  01F6   420          MOVZBL  B^RES$PTAB(R5)[R9], R9
                                 20   13  01FB   421          BEQL    202$
                            59   54   91  01FD   422          CMPB    R4, R9
                                 14   12  0200   423          BNEQ    150$
                         04 AE   51   D0  0202   424 100$:    MOVL    R1, 4(SP)                   ; Establish a new stable point
                                      0206   425            ;
                                      0206   426            ; If characters remain in both strings, bypass the CMPC instruction
                                      0206   427            ; if the first characters differ.  This is ONLY an optimization.
                                      0206   428            ;
                                      0206   429            ; This optimization was removed, to save code space.
                                      0206   430            ;
                                      0206   431            ;NOP
                                      0206   432            ;
                                      0206   433            ; Compare the remaining portions of the strings
                                      0206   434            ;
 63   52   00'A5   61   50   2D  0206   435          CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                                 8E   12  020D   436          BNEQ    0$                          ; If equal, then R0 = 0
                            0203 8F   BA  020F   437 89$:     POPR    #^M<R0,R1,R9>               ; Pop PSL & stable point, restore R9
                               FE50   31  0213   438          BRW     USETB1
                            0203 8F   BA  0216   439 150$:    POPR    #^M<R0,R1,R9>               ; Pop PSL & stable point, restore R9
                               FE58   31  021A   440          BRW     NE
                                0093  31  021D   441 202$:    BRW     201$
                       59   00'A549   9A  0220   442 101$:    MOVZBL  B^RES$PTAB(R5)[R9], R9
                                      0225   443            ;
                                      0225   444            ; R4 is zero; look for a double character.
                                      0225   445            ; R9 contains the value from PTAB.
                                      0225   446            ;
                                 21   11  0225   447          BRB     112$
 59   59   F8 8F   78  0227   448 402$:    ASHL    #-8, R9, R9
                                 D4   13  022C   449          BEQL    100$
                                      022E   450            ;
                                      022E   451            ; Fetch a character, and get it's collating value.
                                      022E   452            ;
                                      022E   453 111$:    FETCHSTR1NP     113$
                                      0248   454 112$:    SPECDBL1
                                 AC   13  0280   455          BEQL    111$
                                 59   95  0282   456 113$:    TSTB    R9
                                 2D   13  0284   457          BEQL    201$
                            59   54   91  0286   458 301$:    CMPB    R4, R9
                                 8B   12  0289   459          BNEQ    150$
 54   54   FE 8F   78  028B   460          ASHL    #-8, R4, R4
                                 95   13  0290   461          BEQL    402$
 59   59   F8 8F   78  0292   462          ASHL    #-8, R9, R9
                                 ED   12  0297   463          BNEQ    301$
                                      0299   464 211$:    FETCHSTR2NP     301$
                                      02B3   465 201$:    SPECDBL2
                                 AC   13  02EB   466          BEQL    211$
                                 97   11  02ED   467          BRB     301$
                                      02EF   468          .DSABL  LSB
                                      02EF   469
```

```
                02EF    470             .END
```

```
EQ                              0000007B R      02
GT                              00000077 R      02
JOIN_0                          00000018 R      02
JOIN_1                          00000096 R      02
LT                              0000006D R      02
NE                              00000075 R      02
POPNE                           00000071 R      02
PSW$M_V                       = 00000002
PSW$V_V                       = 00000001
PSW$V_Z                       = 00000002
RES$PAD                         ********    X   02
RES$PTAB                        ********    X   02
RES$REVERSE                     ********    X   02
RES$STAB                        ********    X   02
RES$TB                          ********    X   02
RES$UPPER                       ********    X   02
SOR$$COLLATE_0                  0000000A RG     02
SOR$$COLLATE_0_A                00000000 RG     02
SOR$$COLLATE_1                  00000088 RG     02
SOR$$COLLATE_1_A                0000007E RG     02
SOR$$COLLATE_2                  00000186 RG     02
USETB                           00000062 R      02
USETB1                          00000066 R      02
```

Psect synopsis

```
PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                           00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
SOR$RO_CODE                     000002EF  (  751.)  02 (  2.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD  NOWRT NOVEC LONG
```

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.04	00:00:01.69
Command processing	132	00:00:00.46	00:00:04.51
Pass 1	138	00:00:02.15	00:00:08.60
Symbol table sort	0	00:00:00.09	00:00:00.09
Pass 2	92	00:00:00.93	00:00:03.99
Symbol table output	4	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	402	00:00:03.72	00:00:18.93

The working set limit was 1200 pages.
11496 bytes (23 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 37 non-local and 62 local symbols.
470 source lines were read in Pass 1, producing 12 object records in Pass 2.
17 pages of virtual memory were used to define 15 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	4

76 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

```
MACRO/DISABLE=TRACE/LIS=LIS$:SORCOLLAT/OBJ=OBJ$:SORCOLLAT MSRC$:SORCOLLAT/UPDATE=(ENH$:SORCOLLAT)
```

SORCOMMAN
LIS

LIBFIXUPD
LIS

REQSYM
R32

CRETRANS
LIS

SFKEYWRD
LIS

SORCOLLAT
LIS

OPCODES
LIS

SORARCHAI
LIS

SORCOLUTI
LIS