D 8

```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**STRCOMPAR                                        E  2
```

```
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR      CCCCCCCC    000000    MM      MM  PPPPPPPP      AAAAAA    RRRRRRRR
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR      CCCCCCCC    000000    MM      MM  PPPPPPPP      AAAAAA    RRRRRRRR
SS              TT      RR      RR  CC          00      00  MMMM  MMMM  PP      PP  AA      AA  RR      RR
SS              TT      RR      RR  CC          00      00  MMMM  MMMM  PP      PP  AA      AA  RR      RR
SS              TT      RR      RR  CC          00      00  MM  MM  MM  PP      PP  AA      AA  RR      RR
SS              TT      RR      RR  CC          00      00  MM  MM  MM  PP      PP  AA      AA  RR      RR
  SSSSSS        TT      RRRRRRRR    CC          00      00  MM      MM  PPPPPPPP    AA      AA  RRRRRRRR
  SSSSSS        TT      RRRRRRRR    CC          00      00  MM      MM  PPPPPPPP    AA      AA  RRRRRRRR
        SS      TT      RR  RR      CC          00      00  MM      MM  PP          AAAAAAAAAA  RR  RR
        SS      TT      RR  RR      CC          00      00  MM      MM  PP          AAAAAAAAAA  RR  RR
        SS      TT      RR    RR    CC          00      00  MM      MM  PP          AA      AA  RR    RR        ....
        SS      TT      RR    RR    CC          00      00  MM      MM  PP          AA      AA  RR    RR        ....
SSSSSSSS        TT      RR      RR    CCCCCCCC    000000    MM      MM  PP          AA      AA  RR      RR      ....
SSSSSSSS        TT      RR      RR    CCCCCCCC    000000    MM      MM  PP          AA      AA  RR      RR      ....

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 STR$COMPARE (     ! Compare 2 strings with blank fill
0
0                   IDENT = '1-006' ! File: STRCOMPAR.B32  Edit: RKR1006
0
0                      ) =
1 BEGIN
1
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 ! FACILITY: String support library
1 !
1 ! ABSTRACT: This module takes 2 input strings of any supported class
1 !       and dtype and returns a +1 if string1 > string2,
1 !       a 0 if string1 = string2, or a -1 if string1 < string2.
1 !
1 ! ENVIRONMENT: User mode, AST level or not or mixed
1 !
1 ! AUTHOR: R. Will, CREATION DATE: 26-Mar-79
1 !
1 ! MODIFIED BY:
1 !
1 ! R. Will, 26-Mar-79 : VERSION 01
1 ! 1-001 - Original
1 ! 1-002 - String cleanup, change name to STR$, remove = length.
1 !         RW  5-Nov-79
1 ! 1-003 - Use CH$COMPARE instead of CH$EQL.  CH$EQL returns 1 if
1 !         strings match, 0 otherwise which is not what this routine is
1 !         supposed to do! SBL 1-Oct-1980
1 ! 1-004 - Extend routine to recognize additional classes of descriptors
1 !         by using $STR$GET_LEN_ADDR to extract length and address
1 !         of first byte of data from descriptors.  Remove string
1 !         interlocking code.  RKR 13-APR-81
1 ! 1-005 - Speed up code.  RKR 7-OCT-1981.
```

```
;    58          0058  1 ! 1-006 - Use SWITCHES ZIP to avoid some cross-jumping that would
;    59          0059  1 !          normally occur in this module.  RKR 18-NOV-1981.
;    60          0060  1 !--
;    61          0061  1 !<BLF/PAGE>
;    63        0062  1 !
;    64        0063  1 ! SWITCHES:
;    65        0064  1 !
;    66        0065  1
;    67        0066  1 SWITCHES ADDRESSING_MODE
;    68        0067  1                 (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    69        0068  1
;    70        0069  1 SWITCHES ZIP;
;    71        0070  1 !
;    72        0071  1 ! LINKAGES
;    73        0072  1 !
;    74        0073  1 REQUIRE 'RTLIN:STRLNK';          ! Linkage to LIB$ANALYZE_DESC_R3
;    75        0258  1
;    76        0259  1 !
;    77        0260  1 ! TABLE OF CONTENTS:
;    78        0261  1 !
;    79        0262  1
;    80        0263  1 FORWARD ROUTINE
;    81        0264  1     STR$COMPARE;   ! compare 2 strings, same contents with blank fill
;    82        0265  1
;    83        0266  1 !
;    84        0267  1 ! INCLUDE FILES:
;    85        0268  1 !
;    86        0269  1
;    87        0270  1 REQUIRE 'RTLIN:RTLPSECT';        ! Declare PSECTS code
;    88        0365  1 REQUIRE 'RTLIN:STRMACROS';       ! use string macros to write code
;    89        1281  1 LIBRARY 'RTLSTARLE';             ! STARLET library for macros and symbols
;    90        1282  1
;    91        1283  1 !
;    92        1284  1 ! MACROS: NONE
;    93        1285  1 !
;    94        1286  1
;    95        1287  1 !
;    96        1288  1 ! EQUATED SYMBOLS: NONE
;    97        1289  1 !
;    98        1290  1
;    99        1291  1 !
;   100        1292  1 ! PSECT DECLARATIONS
;   101        1293  1 !
;   102        1294  1
;   103        1295  1 DECLARE_PSECTS (STR);
;   104        1296  1
;   105        1297  1 !
;   106        1298  1 ! OWN STORAGE: NONE
;   107        1299  1 !
;   108        1300  1
;   109        1301  1 !
;   110        1302  1 ! EXTERNAL REFERENCES
;   111        1303  1 !
;   112        1304  1 !       NONE
;  114        1305  1 GLOBAL ROUTINE STR$COMPARE (    ! compare with blank fill
;  115        1306  1
;  116        1307  1                             STRING1, ! pointer to 1st string descriptor
;  117        1308  1                             STRING2  ! pointer to 2nd string descriptor
;  118        1309  1
;  119        1310  1                            ) : =
;  120        1311  1
;  121        1312  1 !++
;  122        1313  1 ! FUNCTIONAL DESCRIPTION:
;  123        1314  1 !
;  124        1315  1 !       This routine takes two source strings of any supported
;  125        1316  1 !       dtype and class, compares their contents with blank fill
;  126        1317  1 !       for the shorter string and returns
;  127        1318  1 !       a -1 if string1 < string2, 0 if both are the same with
;  128        1319  1 !       blank fill or a 1 if string1 > string2.
;  129        1320  1 !
;  130        1321  1 ! FORMAL PARAMETERS:
;  131        1322  1 !
;  132        1323  1 !       STRING1.rt.dx           pointer to 1st string descriptor
;  133        1324  1 !       STRING2.rt.dx           pointer to 2nd string descriptor
;  134        1325  1 !
;  135        1326  1 ! IMPLICIT INPUTS:
;  136        1327  1 !
;  137        1328  1 !       NONE
;  138        1329  1 !
;  139        1330  1 ! IMPLICIT OUTPUTS:
;  140        1331  1 !
;  141        1332  1 !       NONE
;  142        1333  1 !
;  143        1334  1 ! ROUTINE VALUE:
;  144        1335  1 !
;  145        1336  1 !       MATCH.wl.v              -1 if string1 < string2
;  146        1337  1 !                               0 if both are the same with blank fill
;  147        1338  1 !                               1 if string1 > string2
;  148        1339  1 !
;  149        1340  1 ! SIDE EFFECTS:
;  150        1341  1 !
;  151        1342  1 !       May signal STR$_ILLSTRCLA on bad string class
;  152        1343  1 !
;  153        1344  1 !--
;  154        1345  1
;  155        1346  2     BEGIN
;  156        1347  2
;  157        1348  2     MAP STRING1 : REF $STR$DESCRIPTOR;
;  158        1349  2     MAP STRING2 : REF $STR$DESCRIPTOR;
;  159        1350  2
;  160        1351  3     IF (.STRING1 [DSC$B_CLASS] LEQU DSC$K_CLASS_D  AND
;  161        1352  3         .STRING2 [DSC$B_CLASS] LEQU DSC$K_CLASS_D )
;  162        1353  2     THEN
;  163        1354  3         BEGIN
;  164        1355  4         RETURN ( CH$COMPARE ( .STRING1 [DSC$W_LENGTH],
;  165        1356  4                               .STRING1 [DSC$A_POINTER],
;  166        1357  4                               .STRING2 [DSC$W_LENGTH],
;  167        1358  4                               .STRING2 [DSC$A_POINTER],
;  168        1359  3                               STR$K_FILL_CHAR ) ) ;
;  169        1360  3         END
;  170        1361  3
```

```
2     ELSE        ! do it the hard way
2
3         BEGIN
3         LOCAL
3             S1_LENGTH,   ! Length of first string
3             S1_ADDR,     ! Address of 1st data byte of first string
3             S2_LENGTH,   ! Length of second string
3             S2_ADDR;     ! Address of 1st data byte of second string
3
3 !+
3 ! Compute the lengths and address of first bytes involved
3 !-
3
3         $STR$GET_LEN_ADDR (STRING1, S1_LENGTH, S1_ADDR) ;
3
3         $STR$GET_LEN_ADDR (STRING2, S2_LENGTH, S2_ADDR) ;
3
3 !+
3 ! Return the value from CH$COMPARE
3 !-
3
4         RETURN (CH$COMPARE (.S1_LENGTH, .S1_ADDR,        ! contents = ?
4                              .S2_LENGTH, .S2_ADDR,
3                              STR$K_FILL_CHAR ));
3
2         END ;
1     END;                                          ! End of STR$COMPARE


                                                      .TITLE  STR$COMPARE
                                                      .IDENT  \1-006\

                                                      .EXTRN  STR$ANALYZE_SDESC_R1

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

                                007C 00000            .ENTRY  STR$COMPARE, Save R2,R3,R4,R5,R6          ; 1305
                 55       04 AC  7D 00002             MOVQ    STRING1, R5                               ; 1351
                             51  D4 00006             CLRL    R1                                        ;
                 02       03 A5  91 00008             CMPB    3(R5), #2                                 ;
                             19  1A 0000C             BGTRU   1$                                        ;
                             51  D6 0000E             INCL    R1                                        ;
                 50       08 AC  D0 00010             MOVL    STRING2, R0                               ; 1352
                 02       03 A0  91 00014             CMPB    3(R0), #2                                 ;
                             0D  1A 00018             BGTRU   1$                                        ;
                 54          01  D0 0001A             MOVL    #1, R4                                    ; 1355
   66        20      04   B5 65  2D 0001D             CMPC5   (R5), @4(R5), #32, (R6), @4(R6)           ;
                          04 B6     00023                                                               ;
                             3C  11 00025             BRB     6$                                        ;
                 09          51  E9 00027 1$:         BLBC    R1, 2$                                    ; 1375
                 53          65  3C 0002A             MOVZWL  (R5), S1_LENGTH                           ;
                 52       04 A5  D0 0002D             MOVL    4(R5), ST_ADDR                            ;
                             0F  11 00031             BRB     3$                                        ;
                 50          55  D0 00033 2$:         MOVL    R5, R0                                    ;
                   00000000G 00  16 00036             JSB     STR$ANALYZE_SDESC_R1                      ;
                 53          50  D0 0003C             MOVL    R0, R3                                    ;
                 52          51  D0 0003F             MOVL    R1, R2                                    ;
```

```
                              02        03  A6  91 00042 3$:    CMPB    3(R6), #2                                  ; 1377
                                            09  1A 00046        BGTRU   4$                                         ;
                              50            66  3C 00048        MOVZWL  (R6), S2_LENGTH                            ;
                              51        04  A6  D0 0004B        MOVL    4(R6), S2_ADDR                             ;
                                            09  11 0004F        BRB     5$                                         ;
                              50            56  D0 00051 4$:    MOVL    R6, R0                                     ;
                                  00000000G 00  16 00054        JSB     STR$ANALYZE_SDESC_R1                       ;
                              54            01  D0 0005A 5$:    MOVL    #1, R4                                     ; 1383
        50            20      62            53  2D 0005D        CMPC5   S1_LENGTH, (S1_ADDR), #32, S2_LENGTH, -    ;
                                            61     00062                (S2_ADDR)                                  ;
                                            03  1A 00063 6$:    BGTRU   7$                                         ;
                              54            01  D9 00065        SBWC    #1, R4                                     ;
                              50            54  D0 00068 7$:    MOVL    R4, R0                                     ; 1364
                                                04 0006B        RET                                                ; 1388

; Routine Size:  108 bytes,    Routine Base:  _STR$CODE + 0000
;  199          1389  1 END                           !End of module
;  200          1390  0 ELUDOM




;                             PSECT SUMMARY
;
;     Name                 Bytes                     Attributes
;
;  _STR$CODE                    108  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                        Library Statistics
;
;                                        -------- Symbols --------       Pages       Processing
;     File                               Total    Loaded   Percent       Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1       9776         4         0        581         00:00.8




;                              COMMAND QUALIFIERS

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

; Size:          108 code + 0 data bytes
; Run Time:         00:05.1
; Elapsed Time:     00:17.4
; Lines/CPU Min:    16257
; Lexemes/CPU-Min: 39368
; Memory Used:  82 pages
; Compilation Complete
```

0214 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY