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**STRPOSEXT                                        D 1









  SSSSSSSS  TTTTTTTTTT  RRRRRRRR    PPPPPPPP      OOOOOO      SSSSSSSS  EEEEEEEEEE  XX      XX  TTTTTTTTTT
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR    PPPPPPPP      OOOOOO      SSSSSSSS  EEEEEEEEEE  XX      XX  TTTTTTTTTT
SS              TT      RR      RR  PP      PP  OO      OO  SS          EE          XX      XX      TT
SS              TT      RR      RR  PP      PP  OO      OO  SS          EE          XX      XX      TT
SS              TT      RR      RR  PP      PP  OO      OO  SS          EE            XX  XX        TT
SS              TT      RR      RR  PP      PP  OO      OO  SS          EE            XX  XX        TT
  SSSSSS        TT      RRRRRRRR    PPPPPPPP    OO      OO    SSSSSS    EEEEEEEE        XX          TT
  SSSSSS        TT      RRRRRRRR    PPPPPPPP    OO      OO    SSSSSS    EEEEEEEE        XX          TT
        SS      TT      RR  RR      PP          OO      OO          SS  EE            XX  XX        TT
        SS      TT      RR  RR      PP          OO      OO          SS  EE            XX  XX        TT
        SS      TT      RR    RR    PP          OO      OO          SS  EE          XX      XX      TT           ....
        SS      TT      RR    RR    PP          OO      OO          SS  EE          XX      XX      TT           ....
SSSSSSSS        TT      RR      RR  PP            OOOOOO    SSSSSSSS    EEEEEEEEEE  XX      XX      TT           ....
SSSSSSSS        TT      RR      RR  PP            OOOOOO    SSSSSSSS    EEEEEEEEEE  XX      XX      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
```

```
;     1           0001  0 MODULE STR$POS_EXTR (            ! Extract a substring by position
;     2           0002  0
;     3           0003  0            IDENT = '1-012' ! File: STRPOSEXT.B32   Edit:   RKR1012
;     4           0004  0
;     5           0005  0                ) =
;     6           0006  1 BEGIN
;     7           0007  1 !
;     8           0008  1 !****************************************************************************
;     9           0009  1 !*                                                                          *
;    10           0010  1 !*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                *
;    11           0011  1 !*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    12           0012  1 !*   ALL RIGHTS RESERVED.                                                   *
;    13           0013  1 !*                                                                          *
;    14           0014  1 !*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    15           0015  1 !*   ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16           0016  1 !*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;    17           0017  1 !*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    18           0018  1 !*   OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    19           0019  1 !*   TRANSFERRED.                                                           *
;    20           0020  1 !*                                                                          *
;    21           0021  1 !*   THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;    22           0022  1 !*   AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;    23           0023  1 !*   CORPORATION.                                                           *
;    24           0024  1 !*                                                                          *
;    25           0025  1 !*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    26           0026  1 !*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    27           0027  1 !*                                                                          *
;    28           0028  1 !*                                                                          *
;    29           0029  1 !****************************************************************************
;    30           0030  1 !
;    31           0031  1
;    32           0032  1 !++
;    33           0033  1 ! FACILITY: String support library
;    34           0034  1 !
;    35           0035  1 ! ABSTRACT:
;    36           0036  1 !
;    37           0037  1 !       This module extracts a substring according to the BASIC-PLUS-2
;    38           0038  1 !       syntax.  It finds the substring of a main string starting at the
;    39           0039  1 !       character position specified by the second input parameter and
;    40           0040  1 !       continues for the number of characters specified by the third
;    41           0041  1 !       input parameter.  This substring is copied to the destination
;    42           0042  1 !       string.
;    43           0043  1 !
;    44           0044  1 ! ENVIRONMENT: User mode, AST level or not or mixed
;    45           0045  1 !
;    46           0046  1 ! AUTHOR: R. Will, CREATION DATE: 21-Feb-79
;    47           0047  1 !
;    48           0048  1 ! MODIFIED BY:
;    49           0049  1 !
;    50           0050  1 !  R. Will, 28-Feb-79: VERSION 01
;    51           0051  1 ! 1-001 - Original
;    52           0052  1 ! 1-002 - Change linkage and call to COPY routine.  15-Mar-79  RW
;    53           0053  1 ! 1-003 - Make the string linkages start with STR$.  JBS 04-JUN-1979
;    54           0054  1 ! 1-004 - Change call to STR$$COPY.  JBS 16-JUL-1979
;    55           0055  1 ! 1-005 - Change name to STR$, string cleanup.  RW  1-Nov-79
;    56           0056  1 ! 1-006 - Correct psect name.  JBS 02-NOV-1979
;    57           0057  1 ! 1-007 - Change to new interlock macros.  JBS 06-NOV-1979
```

```
:  58        0058  1 ! 1-008 - Use string interlocks in CALL entry.  RW  15-Nov-79
:  59        0059  1 ! 1-009 - String speedup, remove edit 8.  RW  8-Jan-1980
:  60        0060  1 ! 1-010 - Enhance routine to recognize additional classes of descriptors
:  61        0061  1 !         by using $STR$GET_LEN_ADDR to extract length and adddress
:  62        0062  1 !         of 1st byte of source descriptor.  Remove string interlocking
:  63        0063  1 !         code.  RKR 21-APR-81.
:  64        0064  1 ! 1-011 - Speed up code.  RKR 7-OCT-1981.
:  65        0065  1 ! 1-012 - Use STR$COPY_R_R8 for copying operation.  Use
:  66        0066  1 !         $STR$SIGNAL_FATAL instead of $STR$CHECK_STATUS.
:  67        0067  1 !         RKR 18-NOV-1981.
:  68        0068  1 !--
:  69        0069  1
:  70        0070  1 !<BLF/PAGE>
```

```
1 !
1 ! SWITCHES:
1 !
1
1 SWITCHES ADDRESSING_MODE
1                 (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !
1 ! LINKAGES:
1 !
1
1 REQUIRE 'RTLIN:STRLNK';         ! Use require file with string linkage
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     STR$POS_EXTR,               ! Find a substr by position, CALL
1     STR$POS_EXTR_R8 : STR$JSB_POS_EXT;  ! find a substr by position, JSB
1
1 !
1 ! INCLUDE FILES:
1 !
1
1 REQUIRE 'RTLIN:RTLPSECT';       ! Declare PSECTs code
1
1 REQUIRE 'RTLIN:STRMACROS';      ! use string macros to code
1
1 LIBRARY 'RTLSTARLE';            ! STARLET library for macros and symbol
1
1 !
1 ! MACROS:
1 !
1 !      NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1 !      NONE
1 !
1 ! PSECT DECLARATIONS
1 !
1
1 DECLARE_PSECTS (STR);
1
1 !
1 ! OWN STORAGE:
1 !
1 !      NONE
1 !
1 ! EXTERNAL REFERENCES:
1 !
1
1 EXTERNAL ROUTINE
1     STR$COPY_R_R8 : STR$JSB_COPY_R ,           ! to do copying
1     LIB$STOP;                                  ! signal fatal errors
1
```

```
;  129          1321  1 EXTERNAL LITERAL
;  130          1322  1     STR$_NORMAL,                ; successful completion
;  131          1323  1     STR$_ILLSTRSPE,             ! illegal string specification
;  132          1324  1     STR$_ILLSTRPOS;             ! illegal string position
```

```
1 GLOBAL ROUTINE STR$POS_EXTR (            ! substring by position
1
1       DEST_DESC,              ! Pointer to destination descriptor
1       SRC_DESC,               ! Pointer to source descriptor
1       START_POS,              ! First character to be included
1       END_POS                 ! Last character position to include
1
1                             ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !     This routine extracts the characters starting at the
1 !     character position in the source string specified by the 3rd
1 !     input parameter and continuing to the character position in the
1 !     source string specified by the 4th input parameter, and copies
1 !     that substring to the destination string (by JSB to
1 !     STR$COPY_R_R8) according to the syntax of the class of the
1 !     destination string.
1 !     If the sta-ting position is < 1, 1 is used.  If the ending
1 !     position is > length of the source string, the length of the
1 !     source string is used.  If the starting position > the ending
1 !     position, a null string is returned.  The call entry point is
1 !     implemented by JSBing to the JSB entry point.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !     DEST_DESC.wt.dx         pointer to destination string descriptor
1 !     SRC_DESC.rt.dx          pointer to source string descriptor
1 !     START_POS.rl.r          character position in src to start
1 !                             substring
1 !     END_POS.rl.r            character position in src to end
1 !                             substring
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !     NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !     NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !     SS$_NORMAL        Success
1 !     STR$_ILLSTRPOS    Character position reference outside of string
1 !     STR$_ILLSTRSPE    End pos < start pos, or length too long
1 !     STR$_NEGSTRLEN    Negative length supplied, 0 used
1 !     STR$_TRU          Truncation occured in copying to destination
1 !                       (Warning)
1 !
1 ! SIDE EFFECTS:
1 !
1 !     May signal:
1 !         STR$_FATINTERR      Fatal internal error
1 !         STR$_ILLSTRCLA      Illegal (or unsupported) string class
1 !         STR$_INSVIRMEM      Insufficient virtual memory for
```

```
;  191          1382  1 !                                    reallocation of dynamic string
;  192          1383  1 !
;  193          1384  1 !--
;  194          1385  1
;  195          1386  2     BEGIN
;  196          1387  2
;  197          1388  2     MAP
;  198          1389  2         SRC_DESC  : REF $STR$DESCRIPTOR,
;  199          1390  2         DEST_DESC : REF $STR$DESCRIPTOR;
;  200          1391  2
;  201          1392  2     RETURN STR$POS_EXTR_R8 ( DEST_DESC [0,0,0,0],
;  202          1393  2                              SRC_DESC [0,0,0,0],
;  203          1394  2                              ..START_POS,
;  204          1395  2                              ..END_POS);
;  205          1396  1     END;                                           !End of STR$POS_EXTR


                                                                            .TITLE  STR$POS_EXTR
                                                                            .IDENT  \1-012\

                                                                            .EXTRN  STR$COPY_R_R8, LIB$STOP
                                                                            .EXTRN  STR$_NORMAL, STR$_ILLSTRSPE
                                                                            .EXTRN  STR$_ILLSTRPOS

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

                                              01FC 00000                    .ENTRY  STR$POS_EXTR, Save R2,R3,R4,R5,R6,R7,R8          ; 1325
                               53      10   BC   D0 00002                   MOVL    @END_POS, R3                                     ; 1393
                               52      0C   BC   D0 00006                   MOVL    @START_POS, R2                                   ;
                               50      04   AC   7D 0000A                   MOVQ    DEST_DESC, R0                                    ;
                                          0000V  30 0000E                   BSBW    STR$POS_EXTR_R8                                  ;
                                                 04 00011                   RET                                                      ; 1396

; Routine Size:  18 bytes,    Routine Base:  _STR$CODE + 0000
;  207       1397  1 GLOBAL ROUTINE STR$POS_EXTR_R8 (        ! extract substring by position
;  208       1398  1
;  209       1399  1         DEST_DESC,              ! Pointer to destination descriptor
;  210       1400  1         SRC_DESC,               ! Pointer to source descriptor
;  211       1401  1         START_POS,              ! First character to be included
;  212       1402  1         END_POS                 ! Last character position to include
;  213       1403  1
;  214       1404  1                             ) : STR$JSB_POS_EXT =
;  215       1405  1
;  216       1406  1 !++
;  217       1407  1 ! FUNCTIONAL DESCRIPTION:
;  218       1408  1 !
;  219       1409  1 !       This routine extracts the characters starting at the
;  220       1410  1 !       character position in the source string specified by the 3rd
;  221       1411  1 !       input parameter and continuing to the character position in the
;  222       1412  1 !       source string specified by the 4th input parameter, and copies
;  223       1413  1 !       that substring to the destination string (by JSB to
;  224       1414  1 !       STR$COPY_R_R8) according to the syntax of the class of the
;  225       1415  1 !       destination string.
;  226       1416  1 !       If the starting position is < 1, 1 is used.  If the ending
;  227       1417  1 !       position is > length of the source string, the length of the
;  228       1418  1 !       source string is used.  If the starting position > the ending
;  229       1419  1 !       position, a null string is returned.
;  230       1420  1 !
;  231       1421  1 ! FORMAL PARAMETERS:
;  232       1422  1 !
;  233       1423  1 !       DEST_DESC.wt.dx         pointer to destination string descriptor
;  234       1424  1 !       SRC_DESC.rt.dx          pointer to source string descriptor
;  235       1425  1 !       START_POS.rl.v          character position in src to start
;  236       1426  1 !                               substring
;  237       1427  1 !       END_POS.rl.v            character position in src to end
;  238       1428  1 !                               substring
;  239       1429  1 !
;  240       1430  1 ! IMPLICIT INPUTS:
;  241       1431  1 !
;  242       1432  1 !       NONE
;  243       1433  1 !
;  244       1434  1 ! IMPLICIT OUTPUTS:
;  245       1435  1 !
;  246       1436  1 !       NONE
;  247       1437  1 !
;  248       1438  1 ! COMPLETION CODES:
;  249       1439  1 !
;  250       1440  1 !       SS$_NORMAL      Success
;  251       1441  1 !       STR$_ILLSTRPOS  Character position reference outside of string
;  252       1442  1 !       STR$_ILLSTRSPE  End pos < start pos, or length too long
;  253       1443  1 !       STR$_NEGSTRLEN  Negative length supplied, 0 used
;  254       1444  1 !       STR$_TRU        Truncation occured in copying to destination
;  255       1445  1 !                       (Warning)
;  256       1446  1 !
;  257       1447  1 ! SIDE EFFECTS:
;  258       1448  1 !
;  259       1449  1 !       May signal:
;  260       1450  1 !           STR$_FATINTERR      Fatal internal error
;  261       1451  1 !           STR$_ILLSTRCLA      Illegal (or unsupported) string class
;  262       1452  1 !           STR$_INSVIRMEM      Insufficient virtual memory for
;  263       1453  1 !                               reallocation of dynamic string
;  264          1454  1 !
;  265          1455  1 !--
;  266          1456  1
;  267          1457  2     BEGIN
;  268          1458  2
;  269          1459  2     LOCAL
;  270          1460  2         IN_LEN,                         ! length of source string
;  271          1461  2         IN_ADDR,                        ! addr of 1st byte of source
;  272          1462  2         ACTUAL_START,                   ! MAX (1, .START_POS)
;  273          1463  2         LENGTH,                         ! length to copy
;  274          1464  2         COPY_STATUS,                    ! status returned by copy
;  275          1465  2         RETURN_STATUS;                  ! save status for this routine
;  276          1466  2
;  277          1467  2     MAP
;  278          1468  2         SRC_DESC  : REF $STR$DESCRIPTOR,
;  279          1469  2         DEST_DESC : REF $STR$DESCRIPTOR;
;  280          1470  2
;  281          1471  2     RETURN_STATUS = 1 ;         ! Assume success to follow
;  282          1472  2
;  283          1473  2 !+
;  284          1474  2 ! Extract length and address of 1st byte of source string.
;  285          1475  2 !-
;  286          1476  2     $STR$GET_LEN_ADDR ( SRC_DESC, IN_LEN, IN_ADDR ) ;
;  287          1477  2
;  288          1478  2     ACTUAL_START = .START_POS;
;  289          1479  2
;  290          1480  2 !+
;  291          1481  2 ! length is >=0 and the smaller of the
;  292          1482  2 ! (input end_pos and src len) - start pos (must be >0) + 1
;  293          1483  2 !-
;  294          1484  2     LENGTH = 1 +                        ! length = end - start + 1
;  295          1485  3         BEGIN
;  296          1486  3         IF .IN_LEN LSS .END_POS
;  297          1487  3         THEN
;  298          1488  4             BEGIN
;  299          1489  4             RETURN_STATUS = STR$_ILLSTRPOS;     ! END_POS is too large
;  300          1490  4
;  301          1491  4             .IN_LEN                             ! use srclen, remember
;  302          1492  4                                                 ! error
;  303          1493  4             END
;  304          1494  3         ELSE
;  305          1495  3             .END_POS                            ! use specified
;  306          1496  3         END
;  307          1497  3
;  308          1498  2         -
;  309          1499  2
;  310          1500  3         BEGIN
;  311          1501  3         IF .START_POS LSS 1
;  312          1502  3         THEN
;  313          1503  4             BEGIN
;  314          1504  4             RETURN_STATUS = STR$_ILLSTRPOS;     ! START_POS is too small
;  315          1505  4             ACTUAL_START = 1;                   ! use 1, remember error
;  316          1506  4             1
;  317          1507  4             END
;  318          1508  3         ELSE
;  319          1509  3             .START_POS                          ! use specified
;  320          1510  2         END;
;  321          1511  2
;  322          1512  2       IF .LENGTH LSS 0 THEN
;  323          1513  3           BEGIN
;  324          1514  3           LENGTH = 0;
;  325          1515  3           RETURN_STATUS = STR$_ILLSTRSPE;        ! START_POS > END_POS
;  326          1516  2           END;                                   ! null string, remember
;  327          1517  2                                                  ! error
;  328          1518  2
;  329          1519  2   !+
;  330          1520  2   ! Copy to destination descriptor for length as computed above.
;  331          1521  2   ! Input pointer is  the sum - 1 of the source pointer and input
;  332          1522  2   ! start position ( which must be > 0).
;  333          1523  2   !-
;  334          1524  2
;  335          1525  2       COPY_STATUS =
;  336          1526  2                 STR$COPY_R_R8 ( .DEST_DESC, .LENGTH,
;  337          1527  2                             CH$PLUS (.IN_ADDR, .ACTUAL_START - 1) );
;  338          1528  2
;  339          1529  2       IF .COPY_STATUS NEQ SS$_NORMAL
;  340          1530  2       THEN RETURN_STATUS = .COPY_STATUS;             ! copy truncated,
;  341          1531  2                                                      ! return truncate
;  342          1532  2                                                      ! instead of previous
;  343          1533  2                                                      ! status
;  344          1534  2
;  345          1535  2       $STR$SIGNAL_FATAL (RETURN_STATUS);             ! signal fatal errors
;  346          1536  2       RETURN .RETURN_STATUS;
;  347          1537  1       END;                                           !End of STR$POS_EXTR_R8


                                                      .EXTRN  STR$ANALYZE_SDESC_R1

                        56          50  D0 00000 STR$POS_EXTR_R8::
                                                      MOVL    R0, R6                                         ; 1397
                                    01  DD 00003      PUSHL   #1                                             ; 1471
                        02       03 A1  91 00005      CMPB    3(SRC_DESC), #2                                ; 1476
                                    09  1A 00009      BGTRU   1$                                             ;
                        50          61  3C 0000B      MOVZWL  (SRC_DESC), IN_LEN                             ;
                        55       04 A1  D0 0000E      MOVL    4(SRC_DESC), IN_ADDR                           ;
                                    0C  11 00012      BRB     2$                                             ;
                        50          51  D0 00014 1$:  MOVL    SRC_DESC, R0                                   ;
                           00000000G 00 16 00017      JSB     STR$ANALYZE_SDESC_R1                           ;
                        55          51  D0 0001D      MOVL    R1, R5                                         ;
                        54          52  D0 00020 2$:  MOVL    START_POS, ACTUAL_START                        ; 1478
                        53          50  D1 00023      CMPL    IN_LEN, END_POS                                ; 1486
                                    09  18 00026      BGEQ    3$                                             ;
                        6E 00000000G 8F D0 00028      MOVL    #STR$_ILLSTRPOS, RETURN_STATUS                 ; 1489
                                    03  11 0002F      BRB     4$                                             ; 1491
                        50          53  D0 00031 3$:  MOVL    END_POS, R0                                    ; 1495
                                    52  D5 00034 4$:  TSTL    START_POS                                      ; 1501
                                    0D  14 00036      BGTR    5$                                             ;
                        6E 00000000G 8F D0 00038      MOVL    #STR$_ILLSTRPOS, RETURN_STATL                  ; 1504
                        54          01  D0 0003F      MOVL    #1, ACTUAL_START                               ; 1505
                        52          01  D0 00042      MOVL    #1, R2                                         ; 1503
                        50          52  C2 00045 5$:  SUBL2   R2, R0                                         ; 1500
                        51       01 A0  9E 00048      MOVAB   1(R0), LENGTH                                  ; 1498
                                    09  18 0004C      BGEQ    6$                                             ; 1512
```

```
                                               51  D4 0004E           CLRL    LENGTH                                        ; 1514
                             6E 00000000G  8F  D0 00050           MOVL    #STR$_ILLSTRSPE, RETURN_STATUS                ; 1515
                             52        FF A445 9E 00057 6$:       MOVAB   -1(ACTUAL_START)[IN_ADDR], R2                 ; 1527
                             50            56  D0 0005C           MOVL    DEST_DESC, R0                                 ;
                                00000000G  00  16 0005F           JSB     STR$COPY_R_R8                                 ;
                             01            50  D1 00065           CMPL    COPY_STATUS, #1                               ; 1529
                                           03  13 00068           BEQL    7$                                            ;
                             6E            50  D0 0006A           MOVL    COPY_STATUS, RETURN_STATUS                    ; 1530
                             10            6E  E8 0006D 7$:       BLBS    RETURN_STATUS, 8$                             ; 1535
          04            6E   03            00  ED 00070           CMPZV   #0, #3, RETURN_STATUS, #4                     ;
                                           09  12 00075           BNEQ    8$                                            ;
                                           6E  DD 00077           PUSHL   RETURN_STATUS                                 ;
                   00000000G 00            01  FB 00079           CALLS   #1, LIB$STOP                                  ;
                             50            8E  D0 00080 8$:       MOVL    RETURN_STATUS, R0                             ; 1536
                                               05 00083           RSB                                                   ; 1537

; Routine Size:  132 bytes,    Routine Base:  _STR$CODE + 0012


;  348         1538  1
;  349         1539  1 END                                                  !End of module
;  350         1540  1
;  351         1541  0 ELUDOM
```

```
;                                          PSECT SUMMARY
;
;      Name                        Bytes                           Attributes
;
;   _STR$CODE                          150  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        7         0         581         00:00.7




;                                          COMMAND QUALIFIERS
;
;      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:STRPOSEXT/OBJ=OBJ$:STRPOSEXT MSRC$:STRPOSEXT/UPDATE=(ENH$:STRPOSEXT
;      )
;
; Size:          150 code + 0 data bytes
; Run Time:         00:06.1
; Elapsed Time:     00:31.3
; Lines/CPU Min:    15257
; Lexemes/CPU-Min:  32683
; Memory Used:  87 pages
; Compilation Complete
```

0215 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY