```
LLL                 IIIIIIIII    BBBBBBBBBBBB   RRRRRRRRRRR   TTTTTTTTTTTTTTT   LLL
LLL                 IIIIIIIII    BBBBBBBBBBBB   RRRRRRRRRRR   TTTTTTTTTTTTTTT   LLL
LLL                 IIIIIIIII    BBBBBBBBBBBB   RRRRRRRRRRR   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       BBBBBBBBBBBB   RRRRRRRRRRR         TTT         LLL
LLL                    III       BBBBBBBBBBBB   RRRRRRRRRRR         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
```

STRLENEXT

LIS

```
   1    0001  O MODULE STR$LEN_EXTR (            ! Extract a substr by length
   2    0002  0
   3    0003  0              IDENT = '1-012' ! File: STRLENEXT.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, 21-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 - Change string linkages to 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 - String cleanup, change name to STR$.  RW  1-Nov-79
  56    0056  1 !  1-006 - Make the module name agree with the entry point name, and
  57    0057  1 !              correct the PSECT name.  JBS 02-NOV-1979
```

```
:    58        0058   1 ! 1-007 - Use new interlock macros.  JBS 06-NOV-1979
:    59        0059   1 ! 1-008 - Interlock only from CALL entry.  RW  15-Nov-79
:    60        0060   1 ! 1-009 - String speedup, remove edit 8.  RW  8-Jan-1980
:    61        0061   1 ! 1-010 - E ance to recognize additional classes of descriptors by
:    62        0062   1 !          using $STR$GET_LEN_ADDR to extract length and address of
:    63        0063   1 !          1st data byte from a descriptor.  Remove string interlock
:    64        0064   1 !          code.  RKR 21-APR-81
:    65        0065   1 ! 1-011 - Speed up code.  RKR 7-OCT-1981.
:    66        0066   1 ! 1-012 - Use STR$COPY_R_R8 for copy operation.  Use $STR$SIGNAL_FATAL
:    67        0067   1 !          instead of $STR$CHECK_STATUS.  RKR 18-NOV-1981.
:    68        0068   1 !--
:    69        0069   1
:    70        0070   1 !<BLF/PAGE>
```

```
 72    0071   1 !
 73    0072   1 ! SWITCHES:
 74    0073   1 !
 75    0074   1
 76    0075   1 SWITCHES ADDRESSING_MODE
 77    0076   1             (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
 78    0077   1
 79    0078   1 !
 80    0079   1 ! LINKAGES:
 81    0080   1 !
 82    0081   1
 83    0082   1 REQUIRE 'RTLIN:STRLNK';              ! Use require file with string linkage
 84    0267   1
 85    0268   1 !
 86    0269   1 ! TABLE OF CONTENTS:
 87    0270   1 !
 88    0271   1
 89    0272   1 FORWARD ROUTINE
 90    0273   1     STR$LEN_EXTR,                       ! Extract a substring by len.CALL
 91    0274   1     STR$LEN_EXTR_R8 : STR$JSB_LEN_EXT; ! Extract a substr by len,   JSB
 92    0275   1
 93    0276   1 !
 94    0277   1 ! INCLUDE FILES:
 95    0278   1 !
 96    0279   1
 97    0280   1 REQUIRE 'RTLIN:RTLPSECT';          ! Declare PSECTS code
 98    0375   1
 99    0376   1 REQUIRE 'RTLIN:STRMACROS';         ! use string macros to code
100    1292   1
101    1293   1 LIBRARY 'RTLSTARLE';               ! STARLET library for macros and symbol
102    1294   1
103    1295   1 !
104    1296   1 ! MACROS:
105    1297   1 !
106    1298   1       NONE
107    1299   1 !
108    1300   1 ! EQUATED SYMBOLS:
109    1301   1 !
110    1302   1       NONE
111    1303   1 !
112    1304   1 ! PSECT DECLARATIONS
113    1305   1 !
114    1306   1
115    1307   1 DECLARE_PSECTS (STR);
116    1308   1
117    1309   1 !
118    1310   1 ! OWN STORAGE:
119    1311   1 !
120    1312   1       NONE
121    1313   1 !
122    1314   1 ! EXTERNAL REFERENCES:
123    1315   1 !
124    1316   1
125    1317   1 EXTERNAL ROUTINE
126    1318   1     LIB$STOP,                                ! signal fatal errors
127    1319   1     STR$COPY_R_R8 : STR$JSB_COPY_R ;         ! Routine to do the copy
128    1320   1
```

```
 :  129    1321  1 EXTERNAL LITERAL                          ! successful completion
 :  130    1322  1     STR$_NORMAL,                          ! successful completion
 :  131    1323  1     STR$_NEGSTRLEN,        ! negative string length
 :  132    1324  1     STR$_ILLSTRPOS;       ! illegal string position
 :  133    1325  1     STR$_ILLSTRSPE;       ! illegal string specification
```

STR$LEN_EXTR
1-012

L 12
16-Sep-1984 01:41:37   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:08   [LIBRTL.SRC]STRLENEXT.B32;1

Page 5
(3)

```
  135   1326  1  GLOBAL ROUTINE STR$LEN_EXTR (              ! extract a substring by len
  136   1327  1
  137   1328  1         DEST_DESC,              ! Pointer to destination descriptor
  138   1329  1         SRC_DESC,              ! Pointer to source descriptor
  139   1330  1         CHAR_POS,              ! First character to be included
  140   1331  1         SUB_LENGTH             ! Length of the substring
  141   1332  1
  142   1333  1                        ) : =
  143   1334  1
  144   1335  1  !++
  145   1336  1  ! FUNCTIONAL DESCRIPTION:
  146   1337  1  !
  147   1338  1  !        This routine extracts the characters starting at the
  148   1339  1  !        character position in the source string specified by the input
  149   1340  1  !        parameter and continuing for the number of characters specified
  150   1341  1  !        by the input, and copies that substring
  151   1342  1  !        to the destination string (by JSB to STR$COPY_R_R8) according
  152   1343  1  !        to the syntax of the class of the destination string.
  153   1344  1  !        If the input character position is < 1, 1 is used.  If
  154   1345  1  !        the input character position is > the length
  155   1346  1  !        of the source string, then the destination string becomes a null
  156   1347  1  !        string.  If the substring length is < 1 a null string is
  157   1348  1  !        returned.
  158   1349  1  !        If character position + substring length - 1 > source string
  159   1350  1  !        length then the source string length is used. The CALL entry
  160   1351  1  !        point is implemented by calling the JSB entry point.
  161   1352  1  !
  162   1353  1  ! FORMAL PARAMETERS:
  163   1354  1  !
  164   1355  1  !        DEST_DESC.wt.dx         pointer to destination string descriptor
  165   1356  1  !        SRC_DESC.rt.dx          pointer to source string descriptor
  166   1357  1  !        CHAR_POS.rl.r           character position in src to start
  167   1358  1  !                                substring
  168   1359  1  !        SUB_LENGTH.rl.r         length of substring
  169   1360  1  !
  170   1361  1  ! IMPLICIT INPUTS:
  171   1362  1  !
  172   1363  1  !        NONE
  173   1364  1  !
  174   1365  1  ! IMPLICIT OUTPUTS:
  175   1366  1  !
  176   1367  1  !        NONE
  177   1368  1  !
  178   1369  1  ! COMPLETION CODES:
  179   1370  1  !
  180   1371  1  !        SS$_NORMAL       Success
  181   1372  1  !        STR$_ILLSTRPOS   Character position reference outside of string
  182   1373  1  !        STR$_ILLSTRSPE   End pos < start pos, or length too long
  183   1374  1  !        STR$_NEGSTRLEN   Negative length supplied, 0 used
  184   1375  1  !        STR$_TRU         Truncation occured in copying to destination
  185   1376  1  !                         (Warning)
  186   1377  1  !
  187   1378  1  ! SIDE EFFECTS:
  188   1379  1  !
  189   1380  1  !        May signal:
  190   1381  1  !            STR$_FATINTERR      Fatal internal error
  191   1382  1  !            STR$_ILLSTRCLA      Illegal (or unsupported) string class
```

```
:   192         1383  1 !         STR$_INSVIRMEM      Insufficient virtual memory for
:   193         1384  1 !                             reallocation of dynamic string
:   194         1385  1 !
:   195         1386  1 !--
:   196         1387  1
:   197         1388  2     BEGIN
:   198         1389  2
:   199         1390  2     MAP
:   200         1391  2         SRC_DESC  : REF $STR$DESCRIPTOR,
:   201         1392  2         DEST_DESC : REF $STR$DESCRIPTOR;
:   202         1393  2
:   203         1394  2     RETURN STR$LEN_EXTR_R8 ( DEST_DESC [0,0,0,0],
:   204         1395  2                              SRC_DESC [0,0,0,0],
:   205         1396  2                              ..CHAR_POS,
:   206         1397  2                              ..SUB_LENGTH);
:   207         1398  1     END;                                !End of STR$LEN_EXTR


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

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

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

                              01FC 00000          .ENTRY   STR$LEN_EXTR, Save R2,R3,R4,R5,R6,R7,R8    ; 1326
               53     10  BC  D0 00002            MOVL     @SUB_LENGTH, R3                            ; 1395
               52     0C  BC  D0 00006            MOVL     @CHAR_POS, R2                              :
               50     04  AC  7D 0000A            MOVQ     DEST_DESC, R0                              :
                        0000V 30 0000E            BSBW     STR$LEN_EXTR_R8                            :
                           04 00011               RET                                                ; 1398

; Routine Size:  18 bytes,    Routine Base:  _STR$CODE + 0000
```

STR$LEN_EXTR
1-012

N 12
16-Sep-1984 01:41:37   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:08   [LIBRTL.SRC]STRLENEXT.B32;1

Page 7
(4)

```
:  209          1399   1 GLOBAL ROUTINE STR$LEN_EXTR_R8 (           ! extract a substring by len
:  210          1400   1
:  211          1401   1          DEST_DESC,               ! Pointer to destination descriptor
:  212          1402   1          SRC_DESC,                ! Pointer to source descriptor
:  213          1403   1          CHAR_POS,                ! First character to be included
:  214          1404   1          SUB_LENGTH               ! Length of the substring
:  215          1405   1
:  216          1406   1                                   ) : STR$JSB_LEN_EXT =
:  217          1407   1
:  218          1408   1 !++
:  219          1409   1 ! FUNCTIONAL DESCRIPTION:
:  220          1410   1 !
:  221          1411   1 !        This routine extracts the characters starting at the
:  222          1412   1 !        character position in the source string specified by the input
:  223          1413   1 !        parameter and continuing for the number of characters specified
:  224          1414   1 !        by the input, and copies that substring
:  225          1415   1 !        to the destination string (by JSB to STR$$COPY_R_R8) according
:  226          1416   1 !        to the syntax of the class of the destination string.
:  227          1417   1 !        If the input character position is < 1, 1 is used.  If
:  228          1418   1 !        the input character position is > the length
:  229          1419   1 !        of the source string, then the destination string becomes a null
:  230          1420   1 !        string.  If the substring length is < 1 a null string is
:  231          1421   1 !        returned.
:  232          1422   1 !        If character position + substring length - 1 > source string
:  233          1423   1 !        length then the source string length is used.
:  234          1424   1 !
:  235          1425   1 ! FORMAL PARAMETERS:
:  236          1426   1 !
:  237          1427   1 !        DEST_DESC.wt.dx          pointer to destination string descriptor
:  238          1428   1 !        SRC_DESC.rt.dx           pointer to source string descriptor
:  239          1429   1 !        CHAR_POS.rl.v            character position in src to start
:  240          1430   1 !                                 substring
:  241          1431   1 !        SUB_LENGTH.rl.v          length of substring
:  242          1432   1 !
:  243          1433   1 ! IMPLICIT INPUTS:
:  244          1434   1 !
:  245          1435   1 !        NONE
:  246          1436   1 !
:  247          1437   1 ! IMPLICIT OUTPUTS:
:  248          1438   1 !
:  249          1439   1 !        NONE
:  250          1440   1 !
:  251          1441   1 ! COMPLETION CODES:
:  252          1442   1 !
:  253          1443   1 !        SS$_NORMAL       Success
:  254          1444   1 !        STR$_ILLSTRPOS   Character position reference outside of string
:  255          1445   1 !        STR$_ILLSTRSPE   End pos < start pos, or length too long
:  256          1446   1 !        STR$_NEGSTRLEN   Negative length supplied, 0 used
:  257          1447   1 !        STR$_TRU         Truncation occured in copying to destination
:  258          1448   1 !                         (Warning)
:  259          1449   1 !
:  260          1450   1 ! SIDE EFFECTS:
:  261          1451   1 !
:  262          1452   1 !        May signal:
:  263          1453   1 !            STR$_FATINTERR       Fatal internal error
:  264          1454   1 !            STR$_ILLSTRCLA       Illegal (or unsupported) string class
:  265          1455   1 !            STR$_INSVIRMEM       Insufficient virtual memory for
```

```
 266      1456   1  !                                   reallocation of dynamic string
 267      1457   1  !
 268      1458   1  !--
 269      1459   1
 270      1460   2      BEGIN
 271      1461   2
 272      1462   2      LOCAL
 273      1463   2          IN_LEN,                    ! Length of source string
 274      1464   2          IN_ADDR,                   ! addr of 1st byte of source string
 275      1465   2          START_POS,                 ! CHAR_POS corrected for errors
 276      1466   2          LENGTH,                    ! SUB_LENGTH corrected for errors
 277      1467   2          COPY_STATUS,               ! status returned by copy
 278      1468   2          RETURN_STATUS;             ! keep the routine status
 279      1469   2
 280      1470   2      MAP
 281      1471   2          SRC_DESC  : REF $STR$DESCRIPTOR,
 282      1472   2          DEST_DESC : REF $STR$DESCRIPTOR;
 283      1473   2
 284      1474   2      RETURN_STATUS = 1 ;            ! Assume success to follow
 285      1475   2
 286      1476   2  !+
 287      1477   2  ! Extract length and address of 1st byte of source string.  Signal fatal
 288      1478   2  ! errors encountered.
 289      1479   2  !-
 290      1480   2      $STR$GET_LEN_ADDR ( SRC_DESC, IN_LEN, IN_ADDR ) ;
 291      1481   2
 292      1482   2      START_POS =
 293      1483   3          BEGIN
 294      1484   3          IF .IN_LEN LSS .CHAR_POS - 1
 295      1485   3          THEN
 296      1486   4              BEGIN
 297      1487   4              RETURN_STATUS = STR$_ILLSTRPOS; ! input position is illegal
 298      1488   4
 299      1489   4              .IN_LEN                    ! use srclen and remember error
 300      1490   4              END
 301      1491   3          ELSE
 302      1492   3              .CHAR_POS - 1              ! use original position
 303      1493   2          END;
 304      1494   2
 305      1495   2      IF .START_POS LSS 0 THEN
 306      1496   3          BEGIN
 307      1497   3          START_POS = 0;                ! input position is negative
 308      1498   3          RETURN_STATUS = STR$_ILLSTRPOS; ! use 0 and remember error
 309      1499   2          END;
 310      1500   2
 311      1501   2      LENGTH =
 312      1502   3          BEGIN
 313      1503   3          IF (.IN_LEN - .START_POS) LSS .SUB_LENGTH
 314      1504   3          THEN
 315      1505   4              BEGIN
 316      1506   4              RETURN_STATUS = STR$_ILLSTRSPE;! input length is too large
 317      1507   4
 318      1508   4              .IN_LEN - .START_POS          ! compute length, give error
 319      1509   4              END
 320      1510   4
 321      1511   3          ELSE
 322      1512   3              .SUB_LENGTH                   ! use orignal length
```

STR$LEN_EXTR
1-012

C 13
16-Sep-1984 01:41:37   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:08   [LIBRTL.SRC]STRLENEXT.B32;1

Page 9
     (4)

S
1

```
 323   1513  2          END;
 324   1514  2
 325   1515  2      IF .LENGTH LSS 0 THEN
 326   1516  2          BEGIN
 327   1517  3          LENGTH = 0;                      ! input length is negative
 328   1518  3          RETURN_STATUS = STR$_NEGSTRLEN;  ! use 0, remember error
 329   1519  2          END;
 330   1520  2
 331   1521  2  !+
 332   1522  2  ! Copy to descriptor of the destination for length as computed above.
 333   1523  2  ! Pointer is the sum - 1 of the source pointer and input character
 334   1524  2  ! position (which must be  > 0 and <= source length).
 335   1525  2  !-
 336   1526  2      COPY_STATUS =
 337   1527  2              STR$COPY_R_R8 ( .DEST_DESC, .LENGTH,
 338   1528  2                             CH$PLUS (.IN_ADDR, .START_POS) ) ;
 339   1529  2
 340   1530  2      IF .COPY_STATUS NEQ SS$_NORMAL
 341   1531  2      THEN RETURN_STATUS = .COPY_STATUS;   ! copy truncated, return
 342   1532  2                                          ! truncate 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$LEN_EXTR_R8
```

```
                                        .EXTRN   STR$ANALYZE_SDESC_R1

           55        50  D0 00000 STR$LEN_EXTR_R8::
                                        MOVL    R0, R5                              ; 1399
                     01  DD 00003       PUSHL   #1                                  ; 1474
           02    03  A1  91 00005       CMPB    3(SRC_DESC), #2                     ; 1480
                     09  1A 00009       BGTRU   1$
           50        61  3C 0000B       MOVZWL  (SRC_DESC), IN_LEN
           54    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
           54        51  D0 0001D       MOVL    R1, R4
                     52  D7 00020 2$:   DECL    R2                                  ; 1484
           52        50  D1 00022       CMPL    IN_LEN, R2
                     0A  18 00025       BGEQ    3$
           6E 00000000G  8F D0 00027    MOVL    #STR$_ILLSTRPOS, RETURN_STATUS      ; 1487
           52        50  D0 0002E       MOVL    IN_LEN, START_POS                   ; 1489
                     52  D5 00031 3$:   TSTL    START_POS                          ; 1495
                     09  18 00033       BGEQ    4$
                     52  D4 00035       CLRL    START_POS                          ; 1497
           6E 00000000G  8F D0 00037    MOVL    #STR$_ILLSTRPOS, RETURN_STATUS      ; 1498
           50        52  C2 0003E 4$:   SUBL2   START_POS, R0                      ; 1503
           53        50  D1 00041       CMPL    R0, SUB_LENGTH
                     0C  18 00044       BGEQ    5$
           6E 00000000G  8F D0 00046    MOVL    #STR$_ILLSTRSPE, RETURN_STATUS      ; 1506
           51        50  D0 0004D       MOVL    R0, LENGTH                          ; 1508
                     03  11 00050       BRB     6$
           51        53  D0 00052 5$:   MOVL    SUB_LENGTH, LENGTH                  ; 1512
```

STR$LEN_EXTR
1-012
D 13
16-Sep-1984 01:41:37   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:40:08   [LIBRTL.SRC]STRLENEXT.B32;1
Page 10
(4)
S
1

```
                              09  18 00055 6$:    BGEQ    7$                               ; 1515
                              51  D4 00057         CLRL    LENGTH                           ; 1517
             6E 00000000G     8F  D0 00059         MOVL    #STR$_NEGSTRLEN, RETURN_STATUS   ; 1518
                              52  C0 00060 7$:      ADDL2   IN_ADDR, R2                       ; 1528
                              50  D0 00063         MOVL    DEST_DESC, R0
                 00000000G    00  16 00066         JSB     STR$COPY_R_R8
                          01  50  D1 0006C         CMPL    COPY_STATUS, #1                  ; 1530
                          03  13 0006F         BEQL    8$
                              6E  D0 00071         MOVL    COPY_STATUS, RETURN_STATUS      ; 1531
                          10  6E  E8 00074 8$:     BLBS    RETURN_STATUS, 9$               ; 1535
        04          6E      03  00  ED 00077         CMPZV   #0, #3, RETURN_STATUS, #4
                          09  12 0007C         BNEQ    9$
                              6E  DD 0007E         PUSHL   RETURN_STATUS
             00000000G     00  01  FB 00080         CALLS   #1, LIB$STOP
                              50  8E  D0 00087 9$:  MOVL    RETURN_STATUS, R0              ; 1536
                                  05 0008A         RSB                                     ; 1537
```

; Routine Size: 139 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   157 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

;
;
;   Library Statistics
;
;   -------- Symbols --------   Pages   Processing
;   File   Total   Loaded   Percent   Mapped   Time
;
; _255DUA28:[SYSLIB]STARLET.L32;1   9776   7   0   581   00:00.7

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

```
; Size:            157 code + 0 data bytes
; Run Time:            00:06.0
; Elapsed Time:        00:29.3
; Lines/CPU Min:    15461
; Lexemes/CPU-Min: 33461
; Memory Used:   91 pages
; Compilation Complete
```

STRDUPLCH
LIS

STRCOMPAR
LIS

STRFINDSB
LIS

STRMATCH
LIS

STRMSG
LIS

STRLENEXT
LIS

STRCOPY
LIS

STRFINDFI
LIS

STRLEFT
LIS

STRMULTI
LIS

STRCOMEQL
LIS

STRCONCAT
LIS

STRMOVQ
LIS

STRGETFRE
LIS