```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTTT  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
LLLLLLLLLLLLLLLL    IIIIIIIII     BBBBBBBBBBBB     RRR         RRR         TTT         LLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLL    IIIIIIIII     BBBBBBBBBBBB     RRR         RRR         TTT         LLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLL    IIIIIIIII     BBBBBBBBBBBB     RRR         RRR         TTT         LLLLLLLLLLLLLLLL
```

```
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR      GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  FFFFFFFFFF  RRRRRRRR    EEEEEEEEEE
  SSSSSSSS  TTTTTTTTTT  RRRRRRRR      GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  FFFFFFFFFF  RRRRRRRR    EEEEEEEEEE
SS              TT      RR      RR  GG          EE              TT      FF          RR      RR  EE
SS              TT      RR      RR  GG          EE              TT      FF          RR      RR  EE
SS              TT      RR      RR  GG          EE              TT      FF          RR      RR  EE
SS              TT      RR      RR  GG          EE              TT      FF          RR      RR  EE
  SSSSSS        TT      RRRRRRRR    GG          EEEEEEEE        TT      FFFFFFFF    RRRRRRRR    EEEEEEEE
  SSSSSS        TT      RRRRRRRR    GG          EEEEEEEE        TT      FFFFFFFF    RRRRRRRR    EEEEEEEE
        SS      TT      RR  RR      GG  GGGGGG  EE              TT      FF          RR  RR      EE
        SS      TT      RR  RR      GG  GGGGGG  EE              TT      FF          RR  RR      EE
        SS      TT      RR    RR    GG      GG  EE              TT      FF          RR    RR    EE              ....
        SS      TT      RR    RR    GG      GG  EE              TT      FF          RR    RR    EE              ....
SSSSSSSS        TT      RR      RR    GGGGGG    EEEEEEEEEE      TT      FF          RR      RR  EEEEEEEEEE      ....
SSSSSSSS        TT      RR      RR    GGGGGG    EEEEEEEEEE      TT      FF          RR      RR  EEEEEEEEEE      ....

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
```

```
0001  0 MODULE STR$GET_FREE (                          ! Allocate and deallocate dynamic strings
0002  0                 IDENT = '1-012'                ! File: STRGETFRE.B32 Edit: LEB1012
0003  0                 ) =
0004  1 BEGIN
0005  1 !
0006  1 !****************************************************************************
0007  1 !*                                                                          *
0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0010  1 !*  ALL RIGHTS RESERVED.                                                    *
0011  1 !*                                                                          *
0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0017  1 !*  TRANSFERRED.                                                            *
0018  1 !*                                                                          *
0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0021  1 !*  CORPORATION.                                                            *
0022  1 !*                                                                          *
0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0025  1 !*                                                                          *
0026  1 !*                                                                          *
0027  1 !****************************************************************************
0028  1 !
0029  1
0030  1 !++
0031  1 ! FACILITY:  String handling : allocation.
0032  1 !
0033  1 ! ABSTRACT:
0034  1 !
0035  1 !       This module contains routines to get and free strings.  Within
0036  1 !       the STR$ facility these operations are performed in-line;
0037  1 !       these routines are for outsiders who wish to allocate and
0038  1 !       deallocate strings.
0039  1 !
0040  1 ! ENVIRONMENT:  VAX-11 User Mode
0041  1 !
0042  1 ! AUTHOR: John Sauter, CREATION DATE: 14-MAR-1979
0043  1 !
0044  1 ! MODIFIED BY:
0045  1 !
0046  1 ! 1-001 - Original.  JBS 15-MAR-1979
0047  1 ! 1-002 - Use the new STR error messages.  Conversion of input scalars
0048  1 !           to by-reference will wait until later.  JBS 16-MAY-1979
0049  1 ! 1-003 - Convert the one input scalar to by-reference.  JBS 18-MAY-1979
0050  1 ! 1-004 - Add JSB entry points.  JBS 22-MAY-1979
0051  1 ! 1-005 - Make the JSB entry points end in _R4.  JBS 22-MAY-1979
0052  1 ! 1-006 - Use STRLNK.REQ.  JBS 04-JUN-1979
0053  1 ! 1-007 - Make some minor edits based on the code review.  JBS 12-JUN-1979
0054  1 ! 1-008 - Correct some misspelled comments.  JBS 22-JUN-1979
0055  1 ! 1-009 - Improve some comments.  JBS 30-JUL-1979
0056  1 ! 1-010 - String speedup, status from alloc and dealoc macros.  RW  11-Jan-1980
0057  1 ! 1-011 - Correct a typo in STR$GET1_DX_R4 (an extra dot).  JBS 04-MAR-1980
```

```
;   58          0058  1 ! 1-012 - Change EXTERNAL ROUTINE declaration of LIB$STOP so that it matches
;   59          0059  1 !           the way it is declared in macro $STR$SIGNAL_FATAL (without the
;   60          0060  1 !           NOVALUE attribute).  This is fallout from a new BLISS compiler.
;   61          0061  1 !           LEB 11-May-1983
;   62          0062  1 !--
;   63          0063  1
;   64          0064  1 !<BLF/PAGE>
```

```
!
! SWITCHES:
!

SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);

!
! LINKAGES:
!

REQUIRE 'RTLIN:STRLNK';                                 ! JSB linkages

!
! TABLE OF CONTENTS:
!

FORWARD ROUTINE
    STR$GET1_DX,                                        ! Allocate a string
    STR$GET1_DX_R4 : STR$JSB_GETFRE,                    ! (JSB entry point)
    STR$FREE1_DX,                                       ! Deallocate a string
    STR$FREE1_DX_R4 : STR$JSB_GETFRE;                   ! (JSB entry point)

!
! INCLUDE FILES:
!

REQUIRE 'RTLIN:RTLPSECT';                               ! Macros for defining psects

REQUIRE 'RTLIN:STRMACROS';                              ! String macros

LIBRARY 'RTLSTARLE';                                    ! System symbols

!
! MACROS:
!
!       NONE
!
! EQUATED SYMBOLS:
!
!       NONE
!
! PSECTS:
!

DECLARE_PSECTS (STR);                                   ! Declare psects for STR$ facility

!
! OWN STORAGE:
!
!       NONE
!
! EXTERNAL REFERENCES:
!

EXTERNAL ROUTINE
    LIB$STOP ;                                          ! Signal a fatal error

!+
! The following are the error codes used by this module.
```

```
;  123          1315  1 !-
;  124          1316  1
;  125          1317  1 EXTERNAL LITERAL
;  126          1318  1     STR$_NORMAL : UNSIGNED (6),                    ! Success
;  127          1319  1     STR$_ILLSTRCLA;                                ! Illegal string class
;  128          1320  1
```

```
1 GLOBAL ROUTINE STR$GET1_DX (                          ! Allocate a string
1         LEN,                                          ! Number of bytes to allocate
1         DESCRIP                                       ! Descriptor to allocate into
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Allocate a string.  LEN bytes are allocated to DESCRIP, which
1 !       must be a dynamic descriptor.  If the descriptor already
1 !       has enough storage allocated to it, nothing is done.  However,
1 !       if the descriptor does not reference a "short string", it cannot
1 !       be shortened: it will be deallocated and reallocated instead.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       LEN.rw.r        Number of bytes to allocate in the string.
1 !       DESCRIP.mq.r    The descriptor which will be re-allocated.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !       Always SS$_NORMAL
1 !
1 ! SIDE EFFECTS:
1 !
1 !       May deallocate the descriptor's storage and allocate new
1 !       storage for it.
1 !       Signals ILLEGAL STRING CLASS if the descriptor is not dynamic.
1 !       Signals STR$_INSVIRMEM if can't get dynamic string space
1 !       Signals STR$_FATINTERR if can't free space allocated to string
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         DESCRIP : REF $STR$DESCRIPTOR;
2
2 !+
2 ! Verify that the descriptor is a dynamic string.
2 !-
2
2     CASE .$STR$CLASS (DESCRIP) FROM DSC$K_CLASS_Z TO DSC$K_CLASS_D OF
2         SET
2
2         [DSC$K_CLASS_D] :
3             BEGIN
3
3             LOCAL
```

```
;  187          1378  3                RETURN_STATUS;                    ! remember status
;  188          1379  3
;  189          1380  3  !+
;  190          1381  3  ! Prevent AST's to change string while we are looking at it.  If string is
;  191          1382  3  ! already being written, signal an error
;  192          1383  3  !-
;  193          1384  3            $STR$INTERLOCK (1);
;  194          1385  3
;  195          1386  4            IF (RETURN_STATUS = $STR$INTERLOCK_WRITE (.DESCRIP, 0))
;  196          1387  3            THEN
;  197          1388  3  !+
;  198          1389  3  ! This is a dynamic descriptor.  If it has enough storage already,
;  199          1390  3  ! we need not re-allocate it.
;  200          1391  3  !-
;  201          1392  3
;  202          1393  4                IF ($STR$NEED_ALLOC ((..LEN AND %X'FFFF'), $STR$DYN_AL_LEN (DESCRIP)))
;  203          1394  3                THEN
;  204          1395  4                    BEGIN
;  205          1396  4  !+
;  206          1397  4  ! We must deallocate and reallocate the string.  If the deallocate
;  207          1398  4  ! succeeds then do the reallocate.
;  208          1399  4  !-
;  209          1400  4
;  210          1401  5                    IF (RETURN_STATUS = $STR$DEALLOCATE (DESCRIP))
;  211          1402  4                    THEN
;  212          1403  4                        RETURN_STATUS = $STR$ALLOCATE ((..LEN AND %X'FFFF'), DESCRIP);
;  213          1404  4
;  214          1405  4                    END
;  215          1406  3                ELSE
;  216          1407  3                    $STR$LENGTH (DESCRIP) = (..LEN AND %X'FFFF');
;  217          1408  3
;  218          1409  3            $STR$INTERLOCK_CLEAR (.DESCRIP, 0);
;  219          1410  3            $STR$SIGNAL_FATAL (RETURN_STATUS);
;  220          1411  2            END;
;  221          1412  2
;  222          1413  2        [DSC$K_CLASS_S, INRANGE, OUTRANGE] :
;  223          1414  2            LIB$STOP (STR$_ILLSTRCLA);
;  224          1415  2        TES;
;  225          1416  2
;  226          1417  2    RETURN (SS$_NORMAL);
;  227          1418  1    END;                                          ! end of STR$GET1_DX


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

                                                                        .EXTRN  LIB$STOP, STR$_NORMAL
                                                                        .EXTRN  STR$_ILLSTRCLA, STR$$Q_SHORT_Q
                                                                        .EXTRN  LIB$FREE_VM, STR$_FATINTERR
                                                                        .EXTRN  STR$$INIT, STR$$V_INIT
                                                                        .EXTRN  STR$$ALOC_SHORT
                                                                        .EXTRN  LIB$GET_VM, STR$_INSVIRMEM

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

                                          007C 00000                    .ENTRY  STR$GET1_DX, Save R2,R3,R4,R5,R6        ; 1321
                          56 00000000G  00   9E 00002                   MOVAB   STR$$Q_SHORT_Q, R6                      ;
```

```
                5E          04  C2 00009          SUBL2   #4, SP                                  ;
                52       08 AC  D0 0000C          MOVL    DESCRIP, R2                             ; 1371
        02      00       03 A2  8F 00010          CASEB   3(R2), #0, #2                           ;
      000F            0006     0006  00015 1$:     .WORD   2$-1$,-                                 ;
                                                           2$-1$,-                                 ;
                                                           3$-1$                                   ;
            00000000G   8F      DD 0001B 2$:       PUSHL   #STR$_ILLSTRCLA                         ; 1414
                        015A    31 00021           BRW     30$                                     ;
                55          01  D0 00024 3$:       MOVL    #1, RETURN_STATUS                       ; 1386
                53       04 A2  D0 00027           MOVL    4(R2), R3                               ; 1393
                            54  D4 0002B           CLRL    R4                                      ;
                            53  D5 0002D           TSTL    R3                                      ;
                            06  12 0002F           BNEQ    4$                                      ;
                            54  D6 00031           INCL    R4                                      ;
                            50  D4 00033           CLRL    R0                                      ;
                            13  11 00035           BRB     6$                                      ;
      00F0      8F            62  B1 00037 4$:     CMPW    (R2), #240                              ;
                            05  1B 0003C           BLEQU   5$                                      ;
                50          62  3C 0003E           MOVZWL  (R2), R0                                ;
                            07  11 00041           BRB     6$                                      ;
                50          53  D0 00043 5$:       MOVL    R3, STRING_BLOCK                        ;
                50       FE A0  3C 00046           MOVZWL  -2(STRING_BLOCK), R0                    ;
    000000F0    8F            50  D1 0004A 6$:     CMPL    R0, #240                                ;
                            25  1F 00051           BLSSU   10$                                     ;
                51       04 BC  3C 00053           MOVZWL  @LEN, R1                                ;
                04          54  E9 00057           BLBC    R4, 7$                                  ;
                            50  D4 0005A           CLRL    R0                                      ;
                            13  11 0005C           BRB     9$                                      ;
      00F0      8F            62  B1 0005E 7$:     CMPW    (R2), #240                              ;
                            05  1B 00063           BLEQU   8$                                      ;
                50          62  3C 00065           MOVZWL  (R2), R0                                ;
                            07  11 00068           BRB     9$                                      ;
                50          53  D0 0006A 8$:       MOVL    R3, STRING_BLOCK                        ;
                50       FE A0  3C 0006D           MOVZWL  -2(STRING_BLOCK), R0                    ;
                50          51  D1 00071 9$:       CMPL    R1, R0                                  ;
                            25  13 00074           BEQL    14$                                     ;
                            26  11 00076           BRB     15$                                     ;
                51       04 BC  3C 00078 10$:      MOVZWL  @LEN, R1                                ;
                04          54  E9 0007C           BLBC    R4, 11$                                 ;
                            50  D4 0007F           CLRL    R0                                      ;
                            13  11 00081           BRB     13$                                     ;
      00F0      8F            62  B1 00083 11$:     CMPW    (R2), #240                              ;
                            05  1B 00088           BLEQU   12$                                     ;
                50          62  3C 0008A           MOVZWL  (R2), R0                                ;
                            07  11 0008D           BRB     13$                                     ;
                50          53  D0 0008F 12$:      MOVL    R3, STRING_BLOCK                        ;
                50       FE A0  3C 00092           MOVZWL  -2(STRING_BLOCK), R0                    ;
                50          51  D1 00096 13$:      CMPL    R1, R0                                  ;
                            03  1A 00099           BGTRU   15$                                     ;
                         00D1   31 0009B 14$:      BRW     28$                                     ;
                50        00G  9A 0009E 15$:       MOVZBL  S^STR$_NORMAL, RETURN_STATUS            ; 1401
                            53  D5 000A1           TSTL    R3                                      ;
                            37  13 000A3           BEQL    17$                                     ;
      00F0      8F            62  B1 000A5           CMPW    (R2), #240                              ;
                            15  1A 000AA           BGTRU   16$                                     ;
                51          53  D0 000AC           MOVL    R3, STRING_BLOCK                        ;
                51       FE A1  3C 000AF           MOVZWL  -2(STRING_BLOCK), ALLOC_LENGTH          ;
```

```
                                51  D7 000B3          DECL    R1
                51              07  8A 000B5          BICB2   #7, R1
                51              56  C0 000B8          ADDL2   R6, R1
            00  B1              63  0E 000BB          INSQUE  (R3), @0(INSQUE_ADDR)
                                1B  11 000BF          BRB     17$
                        04      A2  9F 000C1 16$:     PUSHAB  4(R2)
            04  AE              62  3C 000C4          MOVZWL  (R2), 4(SP)
                        04      AE  9F 000C8          PUSHAB  4(SP)
    00000000G   00              02  FB 000CB          CALLS   #2, LIB$FREE_VM
                07              50  E8 000D2          BLBS    RETURN_STATUS, 17$
                50  00000000G   8F  D0 000D5          MOVL    #STR$_FATINTERR, RETURN_STATUS
                55              50  D0 000DC 17$:     MOVL    RETURN_STATUS, RETURN_STATUS
                03              50  E8 000DF          BLBS    RETURN_STATUS, 18$
                            008D    31 000E2          BRW     29$
                07  00000000G   00  E8 000E5 18$:     BLBS    STR$$V_INIT, 19$                    1403
    00000000G   00              00  FB 000EC          CALLS   #0, STR$$INIT
                50              00G 9A 000F3 19$:     MOVZBL  S^STR$_NORMAL, RETURN_STATUS
        00F0    8F          04  BC  B1 000F6          CMPW    @LEN, #240
                                47  1A 000FC          BGTRU   25$
                            04  BC  B5 000FE          TSTW    @LEN
                                04  12 00101          BNEQ    20$
                                53  D4 00103          CLRL    TEMP
                                2D  11 00105          BRB     24$
                51          04  BC  3C 00107 20$:     MOVZWL  @LEN, R1
                                51  D7 0010B          DECL    R1
                51              07  8A 0010D          BICB2   #7, R1
    54          51              56  C1 00110          ADDL3   R6, R1, REMQUE_ADDR
                53          00  B4  0F 00114 21$:     REMQUE  @0(REMQUE_ADDR), TEMP
                                05  1D 00118          BVS     22$
                52              01  D0 0011A          MOVL    #1, ALLOC_DONE
                                0D  11 0011D          BRB     23$
                                52  D4 0011F 22$:     CLRL    ALLOC_DONE
                7E          04  BC  3C 00121          MOVZWL  @LEN, -(SP)
    00000000G   00              01  FB 00125          CALLS   #1, STR$$ALOC_SHORT
                05              52  E8 0012C 23$:     BLBS    ALLOC_DONE, 24$
                38              50  E9 0012F          BLBC    RETURN_STATUS, 27$
                                E0  11 00132          BRB     21$
                33              50  E9 00134 24$:     BLBC    RETURN_STATUS, 27$
                51          08  AC  D0 00137          MOVL    DESCRIP, R1
            04  A1              53  D0 0013B          MOVL    TEMP, 4(R1)
                61          04  BC  B0 0013F          MOVW    @LEN, (R1)
                                25  11 00143          BRB     27$
    7E      08  AC              04  C1 00145 25$:     ADDL3   #4, DESCRIP, -(SP)
            04  AE          04  BC  3C 0014A          MOVZWL  @LEN, 4(SP)
                            04  AE  9F 0014F          PUSHAB  4(SP)
    00000000G   00              02  FB 00152          CALLS   #2, LIB$GET_VM
                09              50  E8 00159          BLBS    RETURN_STATUS, 26$
                50  00000000G   8F  D0 0015C          MOVL    #STR$_INSVIRMEM, RETURN_STATUS
                                05  11 00163          BRB     27$
            08  BC          04  BC  B0 00165 26$:     MOVW    @LEN, @DESCRIP
                55              50  D0 0016A 27$:     MOVL    RETURN_STATUS, RETURN_STATUS
                                03  11 0016D          BRB     29$                                 1393
                62              51  B0 0016F 28$:     MOVW    R1, (R2)                            1407
                10              55  E8 00172 29$:     BLBS    RETURN_STATUS, 31$                  1410
04      55      03              00  ED 00175          CMPZV   #0, #3, RETURN_STATUS, #4
                                09  12 0017A          BNEQ    31$
                                55  DD 0017C          PUSHL   RETURN_STATUS
```

```
                    00000000G  00          01  FB 0017E 30$:    CALLS   #1, LIB$STOP                    ;
                               50          01  D0 00185 31$:    MOVL    #1, R0                          ; 1417
                                               04 00188         RET                                     ; 1418

; Routine Size:  393 bytes,    Routine Base:  _STR$CODE + 0000

;  228          1419  1
```

```
1 GLOBAL ROUTINE STR$GET1_DX_R4 (                       ! Allocate a string
1         LEN,                                          ! Number of bytes to allocate
1         DESCRIP                                       ! Descriptor to allocate into
1     ) : STR$JSB_GETFRE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Allocate a string.  LEN bytes are allocated to DESCRIP, which
1 !       had better be a dynamic descriptor.  If the descriptor already
1 !       has enough storage allocated to it, nothing is done.  However,
1 !       if the descriptor does not reference a "short string", it cannot
1 !       be shortened: it will be deallocated and reallocated instead.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       LEN.rw.v        Number of bytes to allocate in the string.
1 !       DESCRIP.mq.r    The descriptor which will be re-allocated.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !       Always SS$_NORMAL
1 !
1 ! SIDE EFFECTS:
1 !
1 !       May deallocate the descriptor's storage and allocate new
1 !       storage for it.
1 !       Signals ILLEGAL STRING CLASS if the descriptor is not dynamic.
1 !       Signals STR$_INSVIRMEM if can't get dynamic string space
1 !       Signals STR$_FATINTERR if can't free space allocated to string
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         DESCRIP : REF $STR$DESCRIPTOR;
2
2 !+
2 ! Verify that the descriptor is a dynamic string.
2 !-
2
2     CASE .$STR$CLASS (DESCRIP) FROM DSC$K_CLASS_Z TO DSC$K_CLASS_D OF
2         SET
2
2         [DSC$K_CLASS_D] :
3             BEGIN
3
3             LOCAL
```

```
;  287          1477  3                    RETURN_STATUS;                    ! remember status
;  288          1478  3
;  289          1479  3  !+
;  290          1480  3  ! Prevent AST's to change string while we are looking at it.  If string is
;  291          1481  3  ! already being written, signal an error
;  292          1482  3  !-
;  293          1483  3                $STR$INTERLOCK (1);
;  294          1484  3
;  295          1485  4                IF (RETURN_STATUS = $STR$INTERLOCK_WRITE (.DESCRIP, 0))
;  296          1486  3                THEN
;  297          1487  3  !+
;  298          1488  3  ! This is a dynamic descriptor.  If it has enough storage already,
;  299          1489  3  ! we need not re-allocate it.
;  300          1490  3  !-
;  301          1491  3
;  302          1492  4                    IF ($STR$NFED_ALLOC ((.LEN AND %X'FFFF'), $STR$DYN_AL_LEN (DESCRIP)))
;  303          1493  3                    THEN
;  304          1494  4                        BEGIN
;  305          1495  4  !+
;  306          1496  4  ! We must deallocate and reallocate the string.  If the deallocate
;  307          1497  4  ! succeeds then do the reallocate.
;  308          1498  4  !-
;  309          1499  4
;  310          1500  5                        IF (RETURN_STATUS = $STR$DEALLOCATE (DESCRIP))
;  311          1501  4                        THEN
;  312          1502  4                            RETURN_STATUS = $STR$ALLOCATE ((.LEN AND %X'FFFF'), DESCRIP);
;  313          1503  4
;  314          1504  4                        END
;  315          1505  3                    ELSE
;  316          1506  3                        $STR$LENGTH (DESCRIP) = (.LEN AND %X'FFFF');
;  317          1507  3
;  318          1508  3                $STR$INTERLOCK_CLEAR (.DESCRIP, 0);
;  319          1509  3                $STR$SIGNAL_FATAL (RETURN_STATUS);
;  320          1510  2                END;
;  321          1511  2
;  322          1512  2            [DSC$K_CLASS_S, INRANGE, OUTRANGE] :
;  323          1513  2                LIB$STOP (STR$_ILLSTRCLA);
;  324          1514  2            TES;
;  325          1515  2
;  326          1516  2        RETURN (SS$_NORMAL);
;  327          1517  1        END;                                          ! end of STR$GET1_DX_R4


                    5E          0C  C2 00000 STR$GET1_DX_R4::
                                                 SUBL2   #12, SP                                  ; 1420
                    52          51  D0 00003         MOVL    R1, R2                                  ;
                    54          50  D0 00006         MOVL    R0, R4                                  ;
          02        00    03    A2  8F 00009         CASEB   3(DESCRIP), #0, #2                      ; 1470
       000F       0006        0006     000DE 1$:      .WORD   2$-1$,-                                 ;
                                                               2$-1$,-                                 ;
                                                               3$-1$                                   ;
                 00000000G  8F  DD 00014 2$:      PUSHL   #STR$_ILLSTRCLA                         ; 1513
                          0159  31 0001A         BRW     30$                                     ;
             08   AE        01  D0 0001D 3$:      MOVL    #1, RETURN_STATUS                       ; 1485
```

```
                                53      04  A2  D0 00021          MOVL    4(DESCRIP), R3                          ; 1492
                                                50  D4 00025          CLRL    R0                                      ;
                                                53  D5 00027          TSTL    R3                                      ;
                                                06  12 00029          BNEQ    4$                                      ;
                                                50  D6 0002B          INCL    R0                                      ;
                                                51  D4 0002D          CLRL    R1                                      ;
                                                13  11 0002F          BRB     6$                                      ;
                        00F0    8F              62  B1 00031 4$:      CMPW    (DESCRIP), #240                         ;
                                                05  1B 00036          BLEQU   5$                                      ;
                                51              62  3C 00038          MOVZWL  (DESCRIP), R1                           ;
                                                07  11 0003B          BRB     6$                                      ;
                                51              53  D0 0003D 5$:      MOVL    R3, STRING_BLOCK                        ;
                                51      FE      A1  3C 00040          MOVZWL  -2(STRING_BLOCK), R1                    ;
                    000000F0    8F              51  D1 00044 6$:      CMPL    R1, #240                                ;
                                                23  1F 0004B          BLSSU   10$                                     ;
                                04              50  E9 0004D          BLBC    R0, 7$                                  ;
                                                51  D4 00050          CLRL    R1                                      ;
                                                13  11 00052          BRB     9$                                      ;
                        00F0    8F              62  B1 00054 7$:      CMPW    (DESCRIP), #240                         ;
                                                05  1B 00059          BLEQU   8$                                      ;
                                51              62  3C 0005B          MOVZWL  (DESCRIP), R1                           ;
                                                07  11 0005E          BRB     9$                                      ;
                                51              53  D0 00060 8$:      MOVL    R3, STRING_BLOCK                        ;
                                51      FE      A1  3C 00063          MOVZWL  -2(STRING_BLOCK), R1                    ;
        51              54      10              00  ED 00067 9$:      CMPZV   #0, #16, [EN, R1                        ;
                                                23  13 0006C          BEQL    14$                                     ;
                                                24  11 0006E          BRB     15$                                     ;
                                04              50  E9 00070 10$:     BLBC    R0, 11$                                 ;
                                                51  D4 00073          CLRL    R1                                      ;
                                                13  11 00075          BRB     13$                                     ;
                        00F0    8F              62  B1 00077 11$:     CMPW    (DESCRIP), #240                         ;
                                                05  1B 0007C          BLEQU   12$                                     ;
                                51              62  3C 0007E          MOVZWL  (DESCRIP), R1                           ;
                                                07  11 00081          BRB     13$                                     ;
                                51              53  D0 00083 12$:     MOVL    R3, STRING_BLOCK                        ;
                                51      FE      A1  3C 00086          MOVZWL  -2(STRING_BLOCK), R1                    ;
        51              54      10              00  ED 0008A 13$:     CMPZV   #0, #16, [EN, R1                        ;
                                                03  1A 0008F          BGTRU   15$                                     ;
                                                00D0 31 00091 14$:    BRW     28$                                     ;
                                50              00G 9A 00094 15$:     MOVZBL  S^STR$_NORMAL, RETURN_STATUS            ; 1500
                                                53  D5 00097          TSTL    R3                                      ;
                                                3C  13 00099          BEQL    17$                                     ;
                        00F0    8F              62  B1 0009B          CMPW    (DESCRIP), #240                         ;
                                                1A  1A 000A0          BGTRU   16$                                     ;
                                51              53  D0 000A2          MOVL    R3, STRING_BLOCK                        ;
                                51      FE      A1  3C 000A5          MOVZWL  -2(STRING_BLOCK), ALLOC_LENGTH          ;
                                                51  D7 000A9          DECL    R1                                      ;
                                51              07  8A 000AB          BICB2   #7, R1                                  ;
                                51 00000000G0041    9E 000AE          MOVAB   STR$$Q_SHORT_Q[R1], INSQUE_ADDR         ;
                        00      B1              63  0E 000B6          INSQUE  (R3), @0(INSQUE_ADDR)                   ;
                                                1B  11 000BA          BRB     17$                                     ;
                                        04      A2  9F 000BC 16$:     PUSHAB  4(DESCRIP)                              ;
                        08      AE              62  3C 000BF          MOVZWL  (DESCRIP), 8(SP)                        ;
                                        08      AE  9F 000C3          PUSHAB  8(SP)                                   ;
                    00000000G   00              02  FB 000C6          CALLS   #2, LIB$FREE_VM                         ;
                                07              50  E8 000CD          BLBS    RETURN_STATUS, 17$                      ;
                                50 00000000G    8F  D0 000D0          MOVL    #STR$_FATINTERR, RETURN_STATUS          ;
```

```
                08  AE            50  D0 000D7 17$:   MOVL    RETURN_STATUS, RETURN_STATUS                    ;
                    03            50  E8 000DB        BLBS    RETURN_STATUS, 18$                              ;
                              0086  31 000DE        BRW     29$                                             ;
                    07 00000000G  00  E8 000E1 18$:   BLBS    STR$$V_INIT, 19$                                ; 1502
            00000000G  00         00  FB 000E8        CALLS   #0, STR$$INIT                                   ;
                    50            00G 9A 000EF 19$:   MOVZBL  S^STR$_NORMAL, RETURN_STATUS                    ;
              00F0  8F            54  B1 000F2        CMPW    LEN, #240                                       ;
                                  45  1A 000F7        BGTRU   25$                                             ;
                                  54  B5 000F9        TSTW    LEN                                             ;
                                  04  12 000FB        BNEQ    20$                                             ;
                                  6E  D4 000FD        CLRL    TEMP                                            ;
                                  34  11 000FF        BRB     24$                                             ;
                    51            54  3C 00101 20$:   MOVZWL  LEN, R1                                         ;
                                  51  D7 00104        DECL    R1                                              ;
                    51            07  8A 00106        BICB2   #7, R1                                          ;
                04  AE 00000000G0041  9E 00109        MOVAB   STR$$Q_SHORT_Q[R1], REMQUE_ADDR                 ;
                    51        04  BE  9E 00112 21$:   MOVAB   @REMQUE_ADDR, R1                                ;
                    6E        00  B1  0F 00116        REMQUE  @0(R1), TEMP                                    ;
                                  05  1D 0011A        BVS     22$                                             ;
                    53            01  D0 0011C        MOVL    #1, ALLOC_DONE                                  ;
                                  0C  11 0011F        BRB     23$                                             ;
                                  53  D4 00121 22$:   CLRL    ALLOC_DONE                                      ;
                    7E            54  3C 00123        MOVZWL  LEN, -(SP)                                      ;
            00000000G  00         01  FB 00126        CALLS   #1, STR$$ALOC_SHORT                             ;
                    05            53  E8 0012D 23$:   BLBS    ALLOC_DONE, 24$                                 ;
                    2B            50  E9 00130        BLBC    RETURN_STATUS, 27$                              ;
                                  DD  11 00133        BRB     21$                                             ;
                    26            50  E9 00135 24$:   BLBC    RETURN_STATUS, 27$                              ;
                04  A2            6E  D0 00138        MOVL    TEMP, 4(DESCRIP)                                ;
                                  1D  11 0013C        BRB     26$                                             ;
                        04        A2  9F 0013E 25$:   PUSHAB  4(DESCRIP)                                      ;
                08  AE            54  3C 00141        MOVZWL  LEN, 8(SP)                                      ;
                        08        AE  9F 00145        PUSHAB  8(SP)                                           ;
            00000000G  00         02  FB 00148        CALLS   #2, LIB$GET_VM                                  ;
                    09            50  E8 0014F        BLBS    RETURN_STATUS, 26$                              ;
                    50 00000000G  8F  D0 00152        MOVL    #STR$_INSVIRMEM, RETURN_STATUS                  ;
                                  03  11 00159        BRB     27$                                             ;
                    62            54  B0 0015B 26$:   MOVW    LEN, (DESCRIP)                                  ;
                08  AE            50  D0 0015E 27$:   MOVL    RETURN_STATUS, RETURN_STATUS                    ;
                                  03  11 00162        BRB     29$                                             ; 1492
                    62            54  B0 00164 28$:   MOVW    LEN, (DESCRIP)                                  ; 1506
                    12        08  AE  E8 00167 29$:   BLBS    RETURN_STATUS, 31$                              ; 1509
        04      08  AE            03  00  ED 0016B        CMPZV   #0, #3, RETURN_STATUS, #4                   ;
                                      0A  12 00171        BNEQ    31$                                         ;
                                  08  AE  DD 00173        PUSHL   RETURN_STATUS                               ;
            00000000G  00         01  FB 00176 30$:   CALLS   #1, LIB$STOP                                    ;
                    50            01  D0 0017D 31$:   MOVL    #1, R0                                          ; 1516
                    5E            0C  C0 00180        ADDL2   #12, SP                                         ; 1517
                                      05 00183        RSB                                                     ;

; Routine Size:  388 bytes,    Routine Base:  _STR$CODE + 0189

;  328          1518  1
```

```
;  330          1519  1 GLOBAL ROUTINE STR$FREE1_DX (                    ! Deallocate a string
;  331          1520  1          DESCRIP                                 ! The descriptor to deallocate
;  332          1521  1      ) =
;  333          1522  1
;  334          1523  1 !++
;  335          1524  1 ! FUNCTIONAL DESCRIPTION:
;  336          1525  1 !
;  337          1526  1 !        Deallocate a string.  The string must by dynamic.  If the
;  338          1527  1 !        string is "short", it is put on a queue of strings of the
;  339          1528  1 !        proper length.  Otherwise it is returned to virtual memory.
;  340          1529  1 !
;  341          1530  1 ! FORMAL PARAMETERS:
;  342          1531  1 !
;  343          1532  1 !        DESCRIP.wq.r    The descriptor of the string to deallocate.
;  344          1533  1 !
;  345          1534  1 ! IMPLICIT INPUTS:
;  346          1535  1 !
;  347          1536  1 !        NONE
;  348          1537  1 !
;  349          1538  1 ! IMPLICIT OUTPUTS:
;  350          1539  1 !
;  351          1540  1 !        NONE
;  352          1541  1 !
;  353          1542  1 ! ROUTINE VALUE:
;  354          1543  1 ! COMPLETION CODES:
;  355          1544  1 !
;  356          1545  1 !        Always SS$_NORMAL
;  357          1546  1 !
;  358          1547  1 ! SIDE EFFECTS:
;  359          1548  1 !
;  360          1549  1 !        May deallocate virtual storage.
;  361          1550  1 !        May signal STR$_FATINTERR if can't deallocate string
;  362          1551  1 !        May signal STR$_ILLSTRCLA if not a dynamic string
;  363          1552  1 !
;  364          1553  1 !--
;  365          1554  1
;  366          1555  2     BEGIN
;  367          1556  2
;  368          1557  2     MAP
;  369          1558  2         DESCRIP : REF $STR$DESCRIPTOR;
;  370          1559  2
;  371          1560  2 !+
;  372          1561  2 ! Verify that this is a dynamic descriptor.
;  373          1562  2 !-
;  374          1563  2
;  375          1564  2     CASE .$STR$CLASS (DESCRIP) FROM DSC$K_CLASS_S TO DSC$K_CLASS_D OF
;  376          1565  2         SET
;  377          1566  2
;  378          1567  2         [DSC$K_CLASS_D] :
;  379          1568  3             BEGIN
;  380          1569  3
;  381          1570  3             LOCAL
;  382          1571  3                 RETURN_STATUS;                  ! remember status
;  383          1572  3
;  384          1573  3 !+
;  385          1574  3 ! Prevent AST's to change string while we are looking at it.  If string is
;  386          1575  3 ! already being written, signal an error
```

```
;  387          1576  3 !-
;  388          1577  3                $STR$INTERLOCK (1);
;  389          1578  3
;  390          1579  4                IF (RETURN_STATUS = $STR$INTERLOCK_WRITE (.DESCRIP, 0))
;  391          1580  3                THEN
;  392          1581  4                    BEGIN
;  393          1582  4 !+
;  394          1583  4 ! Deallocate the string data.
;  395          1584  4 !-
;  396          1585  4
;  397          1586  5                    IF (RETURN_STATUS = $STR$DEALLOCATE (DESCRIP))
;  398          1587  5 !+
;  399          1588  5 ! Make sure the pointer and length field are zero, so the user is less
;  400          1589  5 ! likely to mistakenly use an old address.  Also, if he calls
;  401          1590  5 ! STR$GET1_DX without reinitializing the descriptor it will not get
;  402          1591  5 ! confused.
;  403          1592  5 !-
;  404          1593  4                    THEN
;  405          1594  5                        BEGIN
;  406          1595  5                        DESCRIP [DSC$W_LENGTH] = 0;
;  407          1596  5                        DESCRIP [DSC$A_POINTER] = 0;
;  408          1597  4                        END;
;  409          1598  4
;  410          1599  3                    END;
;  411          1600  3
;  412          1601  3                $STR$INTERLOCK_CLEAR (.DESCRIP, 0);
;  413          1602  3                $STR$SIGNAL_FATAL (RETURN_STATUS);
;  414          1603  2                END;
;  415          1604  2
;  416          1605  2            [INRANGE, OUTRANGE] :
;  417          1606  2                LIB$STOP (STR$_ILLSTRCLA);
;  418          1607  2            TES;
;  419          1608  2
;  420          1609  2        RETURN (SS$_NORMAL);
;  421          1610  1        END;                                    ! end of STR$FREE1_DX


                                    001C 00000          .ENTRY  STR$FREE1_DX, Save R2,R3,R4                 ; 1519
                 5E                 04   C2 00002        SUBL2   #4, SP                                      ;
                 52          04     AC   D0 00005        MOVL    DESCRIP, R2                                 ; 1564
     01          01          03     A2   8F 00009        CASEB   3(R2), #1, #1                               ;
               000C               0004     0000E 1$:     .WORD   2$-1$,-                                     ;
                                                                  3$-1$                                     ;
                     00000000G     8F   DD 00012 2$:     PUSHL   #STR$_ILLSTRCLA                             ; 1606
                                   63   11 00018         BRB     7$                                          ;
                 54                01   D0 0001A 3$:     MOVL    #1, RETURN_STATUS                           ; 1579
                 50                00G  9A 0001D         MOVZBL  S^STR$_NORMAL, RETURN_STATUS                ; 1586
                 53          04     A2   D0 00020        MOVL    4(R2), R3                                   ;
                                    3C   13 00024        BEQL    5$                                          ;
           00F0  8F                 62   B1 00026        CMPW    (R2), #240                                  ;
                                    1A   1A 0002B        BGTRU   4$                                          ;
                 51                53   D0 0002D        MOVL    R3, STRING_BLOCK                            ;
                 51          FE     A1   3C 00030        MOVZWL  -2(STRING_BLOCK), ALLOC_LENGTH              ;
                                    51   D7 00034        DECL    R1                                          ;
```

```
                                51          07  8A 00036          BICB2   #7, R1                                          ;
                                51 00000000G0041  9E 00039          MOVAB   STR$$Q_SHORT_Q[R1], INSQUE_ADDR                 ;
                        00      B1          63  0E 00041          INSQUE  (R3), @0(INSQUE_ADDR)                           ;
                                            1B  11 00045          BRB     5$                                              ;
                                    04      A2  9F 00047 4$:      PUSHAB  4(R2)                                           ;
                        04      AE          62  3C 0004A          MOVZWL  (R2), 4(SP)                                     ;
                                    04      AE  9F 0004E          PUSHAB  4(SP)                                           ;
                 00000000G      00          02  FB 00051          CALLS   #2, LIB$FREE_VM                                 ;
                                07          50  E8 00058          BLBS    RETURN_STATUS, 5$                               ;
                                50 00000000G  8F  D0 0005B          MOVL    #STR$_FATINTERR, RETURN_STATUS                  ;
                                54          50  D0 00062 5$:      MOVL    RETURN_STATUS, RETURN_STATUS                    ;
                                09          50  E9 00065          BLBC    RETURN_STATUS, 6$                               ;
                                50      04  AC  D0 00068          MOVL    DESCRIP, R0                                     ; 1595
                                            60  B4 0006C          CLRW    (R0)                                            ;
                                    04      A0  D4 0006E          CLRL    4(R0)                                           ; 1596
                                10          54  E8 00071 6$:      BLBS    RETURN_STATUS, 8$                               ; 1602
        04          54          03          00  ED 00074          CMPZV   #0, #3, RETURN_STATUS, #4                       ;
                                            09  12 00079          BNEQ    8$                                              ;
                                            54  DD 0007B          PUSHL   RETURN_STATUS                                   ;
                 00000000G      00          01  FB 0007D 7$:      CALLS   #1, LIB$STOP                                    ;
                                50          01  D0 00084 8$:      MOVL    #1, R0                                          ; 1609
                                                04 00087          RET                                                     ; 1610

; Routine Size:  136 bytes,    Routine Base:  _STR$CODE + 030D

;  422          1611  1
```

```
1 GLOBAL ROUTINE STR$FREE1_DX_R4 (                      ! Deallocate a string
1         DESCRIP                                       ! The descriptor to deallocate
1     ) : STR$JSB_GETFRE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Deallocate a string.  The string must by dynamic.  If the
1 !       string is "short", it is put on a queue of strings of the
1 !       proper length.  Otherwise it is returned to virtual memory.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DESCRIP.wq.r    The descriptor of the string to deallocate.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !       Always SS$_NORMAL.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       May deallocate virtual storage.
1 !       May signal STR$_FATINTERR if can't deallocate string
1 !       May signal STR$_ILLSTRCLA if not a dynamic string
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         DESCRIP : REF $STR$DESCRIPTOR;
2
2 !+
2 ! Verify that this is a dynamic descriptor.
2 !-
2
2     CASE .$STR$CLASS (DESCRIP) FROM DSC$K_CLASS_S TO DSC$K_CLASS_D OF
2         SET
2
2         [DSC$K_CLASS_D] :
3             BEGIN
3
3             LOCAL
3                 RETURN_STATUS;                        ! remember status
3
3 !+
3 ! Prevent AST's to change string while we are looking at it.  If string is
3 ! already being written, signal an error
3 !-
```

```
;  481          1669  3              $STR$INTERLOCK (1);
;  482          1670  3
;  483          1671  4              IF (RETURN_STATUS = $STR$INTERLOCK_WRITE (.DESCRIP, 0))
;  484          1672  3              THEN
;  485          1673  4                  BEGIN
;  486          1674  4  !+
;  487          1675  4  ! Deallocate the string data.
;  488          1676  4  !-
;  489          1677  4
;  490          1678  5                  IF (RETURN_STATUS = $STR$DEALLOCATE (DESCRIP))
;  491          1679  5  !+
;  492          1680  5  ! Make sure the pointer and length field are zero, so the user is less
;  493          1681  5  ! likely to mistakenly use an old address.  Also, if he calls
;  494          1682  5  ! STR$GET1_DX without reinitializing the descriptor it will not get
;  495          1683  5  ! confused.
;  496          1684  5  !-
;  497          1685  4                  THEN
;  498          1686  5                      BEGIN
;  499          1687  5                      DESCRIP [DSC$W_LENGTH] = 0;
;  500          1688  5                      DESCRIP [DSC$A_POINTER] = 0;
;  501          1689  4                      END;
;  502          1690  4
;  503          1691  3                  END;
;  504          1692  3
;  505          1693  3              $STR$INTERLOCK_CLEAR (.DESCRIP, 0);
;  506          1694  3              $STR$SIGNAL_FATAL (RETURN_STATUS);
;  507          1695  2              END;
;  508          1696  2
;  509          1697  2          [INRANGE, OUTRANGE] :
;  510          1698  2              LIB$STOP (STR$_ILLSTRCLA);
;  511          1699  2          TES;
;  512          1700  2
;  513          1701  2      RETURN (SS$_NORMAL);
;  514          1702  1      END;                                    ! end of STR$FREE1_DX_R4


                    5E          04  C2 00000 STR$FREE1_DX_R4::
                                                 SUBL2   #4, SP                                  ; 1612
                    52          50  D0 00003         MOVL    R0, R2                                  ;
01                  01      03  A2  8F 00006         CASEB   3(DESCRIP), #1, #1                      ; 1656
                  000C        0004     0000B 1$:      .WORD   2$-1$,-                                 ;
                                                                3$-1$                                ;
                 00000000G    8F  DD 0000F 2$:       PUSHL   #STR$_ILLSTRCLA                         ; 1698
                              5F  11 00015           BRB     7$                                      ;
                    54        01  D0 00017 3$:       MOVL    #1, RETURN_STATUS                       ; 1671
                    50        00G 9A 0001A           MOVZBL  S^STR$_NORMAL, RETURN_STATUS            ; 1678
                    53      04  A2  D0 0001D         MOVL    4(DESCRIP), R3                          ;
                              3C  13 00021           BEQL    5$                                      ;
          00F0      8F        62  B1 00023           CMPW    (DESCRIP), #240                         ;
                              1A  1A 00028           BGTRU   4$                                      ;
                    51        53  D0 0002A           MOVL    R3, STRING_BLOCK                        ;
                    51      FE  A1  3C 0002D         MOVZWL  -2(STRING_BLOCK), ALLOC_LENGTH          ;
                              51  D7 00031           DECL    R1                                      ;
                    51        07  8A 00033           BICB2   #7, R1                                  ;
```

```
                                51 00000000G0041  9E 00036          MOVAB   STR$$Q_SHORT_Q[R1], INSQUE_ADDR          ;
                         00     B1             63  0E 0003E          INSQUE  (R3), @0(INSQUE_ADDR)                  ;
                                               1B  11 00042          BRB     5$                                     ;
                                        04     A2  9F 00044 4$:      PUSHAB  4(DESCRIP)                             ;
                         04     AE             62  3C 00047          MOVZWL  (DESCRIP), 4(SP)                       ;
                                        04     AE  9F 0004B          PUSHAB  4(SP)                                  ;
                   00000000G    00             02  FB 0004E          CALLS   #2, LIB$FREE_VM                        ;
                                07             50  E8 00055          BLBS    RETURN_STATUS, 5$                      ;
                                50 00000000G   8F  D0 00058          MOVL    #STR$_FATINTERR, RETURN_STATUS         ;
                                54             50  D0 0005F 5$:      MOVL    RETURN_STATUS, RETURN_STATUS           ;
                                05             50  E9 00062          BLBC    RETURN_STATUS, 6$                      ;
                                               62  B4 00065          CLRW    (DESCRIP)                              ; 1687
                                        04     A2  D4 00067          CLRL    4(DESCRIP)                             ; 1688
                                10             54  E8 0006A 6$:      BLBS    RETURN_STATUS, 8$                      ; 1694
     04             54          03             00  ED 0006D          CMPZV   #0, #3, RETURN_STATUS, #4              ;
                                               09  12 00072          BNEQ    8$                                     ;
                                               54  DD 00074          PUSHL   RETURN_STATUS                          ;
                   00000000G    00             01  FB 00076 7$:      CALLS   #1, LIB$STOP                           ;
                                50             01  D0 0007D 8$:      MOVL    #1, R0                                 ; 1701
                                5E             04  C0 00080          ADDL2   #4, SP                                 ; 1702
                                                   05 00083          RSB                                            ;

; Routine Size:  132 bytes,    Routine Base:  _STR$CODE + 0395

;  515          1703  1
;  516          1704  1 END                                          ! end of module STR$GET_FREE
;  517          1705  1
;  518          1706  0 ELUDOM


;                                  PSECT SUMMARY
;      Name                     Bytes                        Attributes
;  _STR$CODE                     1049  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        9         0        581        00:00.7
```

```
;                                       COMMAND QUALIFIERS

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

; Size:         1049 code + 0 data bytes
; Run Time:        00:15.0
; Elapsed Time:    00:59.8
; Lines/CPU Min:    6837
; Lexemes/CPU-Min: 36296
; Memory Used:  156 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

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