```
                                        M 1

BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**BASSTR                                          K 1

BBBBBBBB      AAAAAA      SSSSSSSS    SSSSSSSS  TTTTTTTTTT  RRRRRRRR
BBBBBBBB      AAAAAA      SSSSSSSS    SSSSSSSS  TTTTTTTTTT  RRRRRRRR
BB      BB  AA      AA  SS          SS              TT      RR      RR
BB      BB  AA      AA  SS          SS              TT      RR      RR
BB      BB  AA      AA  SS          SS              TT      RR      RR
BB      BB  AA      AA  SS          SS              TT      RR      RR
BBBBBBBB    AA      AA    SSSSSS      SSSSSS        TT      RRRRRRRR
BBBBBBBB    AA      AA    SSSSSS      SSSSSS        TT      RRRRRRRR
BB      BB  AAAAAAAAAA          SS          SS      TT      RR  RR
BB      BB  AAAAAAAAAA          SS          SS      TT      RR  RR
BB      BB  AA      AA          SS          SS      TT      RR    RR        ....
BB      BB  AA      AA          SS          SS      TT      RR    RR        ....
BBBBBBBB    AA      AA  SSSSSSSS    SSSSSSSS        TT      RR      RR      ....
BBBBBBBB    AA      AA  SSSSSSSS    SSSSSSSS        TT      RR      RR      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE BAS$STR (                       ! Routines to do BASIC STR$ function
0                 IDENT = '1-008'        ! module BASSTR.B32 Edit: PLL1008
0                 ) =
1 BEGIN
1 !
1 !*****************************************************************************
1 !*                                                                           *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
1 !*  ALL RIGHTS RESERVED.                                                     *
1 !*                                                                           *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
1 !*  TRANSFERRED.                                                             *
1 !*                                                                           *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
1 !*  CORPORATION.                                                             *
1 !*                                                                           *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
1 !*                                                                           *
1 !*                                                                           *
1 !*****************************************************************************
1 !
1
1
1 !++
1 ! FACILITY: BASIC Support Library
1 !
1 ! ABSTRACT:
1 !
1 !       This module has entry points long, floating, double,
1 !       g floating, and h floating.
1 !       The double routine checks for a BASIC frame and picks
1 !       up the scale factor.  Then all routines convert a number
1 !       to a numeric string as it would be formatted by the BASIC print
1 !       statement but without leading or trailing spaces (by a CALL to the
1 !       correct BAS$ conversion routine).
1 !
1 ! ENVIRONMENT: User mode, AST level or not or mixed
1 !
1 ! AUTHOR: R. Will, CREATION DATE: 8-Mar-79
1 !
1 ! MODIFIED BY:
1 !
1 ! R. Will, 8-Mar-79: VERSION 01
1 ! 01    - original
1 ! 1-002 - Prefix string linkages with STR$.  JBS 04-JUN-1979
1 ! 1-003 - Add BASLNK for scaling linkages.  RW  26-Jun-79
1 ! 1-004 - Change to use new conversion routines.  RW 7-Jul-79
1 ! 1-005 - Add longword entry point.  RW  10-Sept-79
1 ! 1-006 - String cleanup, don't use $STR$ macros.  RW  30-Oct-79
1 ! 1-007 - Add entry points for g & h floating.  PLL 3-Sep-81
```

```
:     58            0058  1  ! 1-008 - Add entry point for packed decimal.  PLL 19-Jan-82
:     59            0059  1  !--
:     60            0060  1  !<BLF/PAGE>
```

```
;    62         0061   1 !
;    63         0062   1 ! SWITCHES:
;    64         0063   1 !
;    65         0064   1
;    66         0065   1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    67         0066   1
;    68         0067   1 !
;    69         0068   1 ! LINKAGES: NONE
;    70         0069   1 !
;    71         0070   1
;    72         0071   1 !
;    73         0072   1 ! TABLE OF CONTENTS:
;    74         0073   1 !
;    75         0074   1
;    76         0075   1 FORWARD ROUTINE
;    77         0076   1     BAS$STR_L : NOVALUE,                       ! Find STR$ of a longword value
;    78         0077   1     BAS$STR_F : NOVALUE,                       ! Find STR$ of a floating value
;    79         0078   1     BAS$STR_D : NOVALUE,                       ! Find STR$ of a double value
;    80         0079   1     BAS$STR_G : NOVALUE,                       ! Find STR$ of a g float value
;    81         0080   1     BAS$STR_H : NOVALUE,                       ! Find STR$ of an h float value
;    82         0081   1     BAS$STR_P : NOVALUE;                       ! Find STR$ of a decimal value
;    83         0082   1
;    84         0083   1 !
;    85         0084   1 ! INCLUDE FILES:
;    86         0085   1 !
;    87         0086   1
;    88         0087   1 REQUIRE 'RTLIN:RTLPSECT';                      ! Declare PSECTs code
;    89         0182   1 REQUIRE 'RTLIN:BASLNK';                        ! Linkages for BASIC scaling
;    90         0259   1 REQUIRE 'RTLIN:BASFRAME';                      ! Define offsets in a BASIC frame
;    91         0462   1
;    92         0463   1 !
;    93         0464   1 ! MACROS: NONE
;    94         0465   1 !
;    95         0466   1
;    96         0467   1 !
;    97         0468   1 ! EQUATED SYMBOLS:
;    98         0469   1 !
;    99         0470   1
;   100         0471   1 LITERAL
;   101         0472   1     digits_in_long = 10,                       ! # of digits to display for longword
;   102         0473   1                                                !  note: float & double use the default
;   103         0474   1     strip_spaces = 1;                          ! flag for stripping spaces
;   104         0475   1
;   105         0476   1 !
;   106         0477   1 ! PSECT DECLARATIONS
;   107         0478   1 !
;   108         0479   1
;   109         0480   1 DECLARE_PSECTS (BAS);
;   110         0481   1
;   111         0482   1 !
;   112         0483   1 ! OWN STORAGE: NONE
;   113         0484   1 !
;   114         0485   1
;   115         0486   1 !
;   116         0487   1 ! EXTERNAL REFERENCES:
;   117         0488   1 !
;   118         0489   1
;  119           0490  1 EXTERNAL ROUTINE
;  120           0491  1     BAS$CVT_OUT_D_G,                              ! Convert dbl to BASIC string format
;  121           0492  1     BAS$CVT_OUT_G_G,                              ! Convert gfloat to BASIC string format
;  122           0493  1     BAS$CVT_OUT_H_G,                              ! Convert hfloat to BASIC string format
;  123           0494  1     BAS$CVT_OUT_P_G;                              ! Convert packed to BASIC string format
;  124           0495  1
;  125           0496  1 BUILTIN
;  126           0497  1     CVTLD;                                        ! Convert long to double to call CVT rtn
```

```
1 GLOBAL ROUTINE BAS$STR_L (                    ! convert integer to string
1                            STRING,            ! Address of destination descriptor
1                            VALUE) :           ! Find numeric value of this number
1                            NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine takes a longword integer and formats it as the BASIC PRINT
1 !       statement would without leading and trailing spaces
1 !       and gives that value to the destination string.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       STRING.wt.dx            pointer to input string descriptor
1 !       VALUE.rl.v              value of a longword number
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       This routine calls the conversion routine and so may signal any of its
1 !       errors or have any of its side effects.  In particular, the conversion
1 !       routine calls STR$ routines and so may allocate or deallocate dynamic
1 !       string space, and lock a string from being written for a period.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         STRING : REF BLOCK [8,BYTE];
2
2     LOCAL
2         STR_LENGTH : WORD,                            ! conversion rtn returns len
2         TEMP : VECTOR [2, LONG];                      ! need double to pass to cnv
2
2     CVTLD (VALUE, TEMP [0]);                          ! make value into double
2     BAS$CVT_OUT_D_G (TEMP [0],                        ! convert this value to string
2                 strip_spaces,                         ! set flag to strip spaces
2                 STR_LENGTH,                           ! return bytes needed for str
2                 STRING [0,0,0,0],                     ! descriptor of result string
2                 0,                                    ! no scale factor
2                 digits_in_long);                      ! # of significant digits
2
2     RETURN;
```

```
; 185          0555  1     END;                                        !End of BAS$STR_L


                                                              .TITLE  BAS$STR
                                                              .IDENT  \1-008\

                                                              .EXTRN  BAS$CVT_OUT_D_G
                                                              .EXTRN  BAS$CVT_OUT_G_G
                                                              .EXTRN  BAS$CVT_OUT_H_G
                                                              .EXTRN  BAS$CVT_OUT_P_G

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

                                      0000 00000              .ENTRY  BAS$STR_L, Save nothing                   ; 0498
                    5E             0C   C2 00002              SUBL2   #12, SP                                   ;
            04      AE       08    AC   6E 00005              CVTLD   VALUE, TEMP                               ; 0546
                                   0A   DD 0000A              PUSHL   #10                                       ; 0550
                                   7E   D4 0000C              CLRL    -(SP)                                     ;
                             04    AC   DD 0000E              PUSHL   STRING                                    ;
                             0C    AE   9F 00011              PUSHAB  STR_LENGTH                                ; 0547
                                   01   DD 00014              PUSHL   #1                                        ; 0550
                             18    AE   9F 00016              PUSHAB  TEMP                                      ; 0547
         00000000G  00             06   FB 00019              CALLS   #6, BAS$CVT_OUT_D_G                       ; 0550
                                        04 00020              RET                                               ; 0555

; Routine Size:  33 bytes,    Routine Base:  _BAS$CODE + 0000
```

```
1 GLOBAL ROUTINE BAS$STR_F (                          ! floating number to a string
1                           STRING,                   ! Address of destination descriptor
1                           VALUE) :                  ! Find numeric value of this string
1                           NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine takes a floating number and formats it as the BASIC PRINT
1 !       statement would without leading and trailing spaces
1 !       and gives that value to the destination string.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       STRING.wt.dx             pointer to input string descriptor
1 !       VALUE.rf.v               value of a floating number
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       This routine calls the conversion and so may signal any of its errors
1 !       or have any of its side effects.  In particular, the conversion routine
1 !       calls STR$ routines and so may allocate or deallocate dynamic string
1 !       space, or write lock a string for a time.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2       STRING : REF BLOCK [8,BYTE];
2
2     LOCAL
2       STR_LENGTH : WORD,                            ! conversion rtn returns len
2       TEMP : VECTOR [2, LONG];                      ! need double to pass to cnv
2
2     TEMP [0] = .VALUE;                              ! make value into double
2     TEMP [1] = 0;
2     BAS$CVT_OUT_D_G (TEMP [0],                      ! convert this value to string
2                   strip_spaces,                     ! set flag to strip spaces
2                   STR_LENGTH,                       ! return bytes needed for str
2                   STRING [0,0,0,0]);                ! descriptor of result string
2                                                     ! no scale to cvt
2                                                     ! default # of digits
2
```

```
;  244          0613  2     RETURN;
;  245          0614  1     END;                                         !End of BAS$STR_F



                                           0000 00000          .ENTRY   BAS$STR_F, Save nothing                               ; 0556
                          5E           0C   C2 00002           SUBL2    #12, SP                                                ;
                    04    AE     08    AC   D0 00005           MOVL     VALUE, TEMP                                            ; 0604
                                 08    AE   D4 0000A           CLRL     TEMP+4                                                 ; 0605
                                 04    AC   DD 0000D           PUSHL    STRING                                                 ; 0609
                                 04    AE   9F 00010           PUSHAB   STR_LENGTH                                             ; 0606
                                       01   DD 00013           PUSHL    #1                                                     ; 0609
                                 10    AE   9F 00015           PUSHAB   TEMP                                                   ; 0606
              00000000G   00           04   FB 00018           CALLS    #4, BAS$CVT_OUT_D_G                                    ; 0609
                                            04 0001F           RET                                                             ; 0614

; Routine Size:  32 bytes,    Routine Base:  _BAS$CODE + 0021
```

```
1 GLOBAL ROUTINE BAS$STR_D (                         ! convert double to string
1                           STRING,                 ! Address of destination descriptor
1                           VALUE1,                 ! 1st longword of double value to put in
1                           VALUE2) :               ! 2nd longwrd of double value for string
1                           NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine takes a double number and formats it as the BASIC PRINT
1 !       statement would, except without leading and trailing spaces,
1 !       and gives that value to the destination string.
1 !       Note that this routine violates the calling standard by accepting and
1 !       calling a routine with double floating passed by value.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       STRING.wt.dx            pointer to input string descriptor
1 !       VALUE.rd.v              value of a double number
1 !       (VALUE1 and VALUE2 used to pick up the 2 words of double value)
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       Scale factor from the BASIC frame
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       This routine calls the conversion routine and so may signal any of its
1 !       errors and have any of its side effects.  In particular, the conversion
1 !       routine calls STR$ routines and so may allocate or deallocate
1 !       dynamic string space, or write lock a string for a short time.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         STRING : REF BLOCK [8,BYTE];
2
2     LOCAL
2         STR_LENGTH : WORD;                          ! conversion rtn returns len
2
2     BAS$CVT_OUT_D_G (VALUE1,                        ! convert this value to string
2                     strip_spaces,                   ! set flag to strip spaces
2                     STR_LENGTH,                     ! return bytes needed for str
2                     STRING [0,0,0,0],               ! return string
2                     $BAS$SCALE);                    ! scale factor
2                                                     ! default # of digits
```

```
;  304           0672  2
;  305           0673  2      RETURN;
;  306           0674  1      END;                          !End of BAS$STR_D


                                                                  .EXTRN   BAS$$SCALE_L_R1

                                         0FFC 00000               .ENTRY   BAS$STR_D, Save R2,R3,R4,R5,R6,R7,R8,R9,-    ; 0615
                                                                           R10,R11                                       :
                           5E          04   C2 00002              SUBL2    #4, SP                                        :
                           51          5D   D0 00005              MOVL     FP, FMP                                       ; 0669
                           50      0C  A1   D0 00008              MOVL     12(FMP), R0                                   :
                            00000000G  00   16 0000C              JSB      BAS$$SCALE_L_R1                               :
                                       50   DD 00012              PUSHL    R0                                            :
                                   04  AC   DD 00014              PUSHL    STRING                                        :
                                   08  AE   9F 00017              PUSHAB   STR_LENGTH                                    ; 0666
                                       01   DD 0001A              PUSHL    #1                                            ; 0669
                                   08  AC   9F 0001C              PUSHAB   VALUE1                                        ; 0666
              00000000G  00            05   FB 0001F              CALLS    #5, BAS$CVT_OUT_D_G                           ; 0669
                                            04 00026              RET                                                    ; 0674

; Routine Size:  39 bytes,    Routine Base:  _BAS$CODE + 0041
```

```
GLOBAL ROUTINE BAS$STR_G (                           ! convert g float to string
                            STRING,                  ! Address of destination descriptor
                            VALUE1,                  ! 1st longword of g float value to put in
                            VALUE2) :                ! 2nd longwrd of g float value for string
                            NOVALUE =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine takes a g float number and formats it as the BASIC PRINT
!       statement would, except without leading and trailing spaces,
!       and gives that value to the destination string.
!       Note that this routine violates the calling standard by accepting and
!       calling a routine with g floating passed by value.
!
! FORMAL PARAMETERS:
!
!       STRING.wt.dx            pointer to input string descriptor
!       VALUE.rg.v              value of a g float number
!       (VALUE1 and VALUE2 used to pick up the 2 words of g float value)
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       NONE
!
! SIDE EFFECTS:
!
!       This routine calls the conversion routine and so may signal any of its
!       errors and have any of its side effects.  In particular, the conversion
!       routine calls STR$ routines and so may allocate or deallocate
!       dynamic string space, or write lock a string for a short time.
!
!--

    BEGIN

    MAP
        STRING : REF BLOCK [8,BYTE];

    LOCAL
        STR_LENGTH : WORD;                               ! conversion rtn returns len

    BAS$CVT_OUT_G_G (VALUE1,                             ! convert this value to string
                    strip_spaces,                        ! set flag to strip spaces
                    STR_LENGTH,                          ! return bytes needed for str
                    STRING [0,0,0,0]);                   ! return string
                                                         ! default # of digits

```

```
;  365         0732  2      RETURN;
;  366         0733  1      END;                             !End of BAS$STR_G



                                                  0000 00000          .ENTRY  BAS$STR_G, Save nothing              ; 0675
                             5E             04  C2 00002          SUBL2   #4, SP                                  ;
                                      04    AC  DD 00005          PUSHL   STRING                                  ; 0729
                                      04    AE  9F 00008          PUSHAB  STR_LENGTH                              ; 0726
                                            01  DD 0000B          PUSHL   #1                                      ; 0729
                                      08    AC  9F 0000D          PUSHAB  VALUE1                                  ; 0726
                   00000000G  00            04  FB 00010          CALLS   #4, BAS$CVT_OUT_G_G                     ; 0729
                                                04 00017          RET                                             ; 0733

; Routine Size:  24 bytes,    Routine Base:  _BAS$CODE + 0068
```

```
GLOBAL ROUTINE BAS$STR_H (                          ! convert g float to string
                           STRING,                  ! Address of destination descriptor
                           VALUE1,                  ! 1st longword of h float value to put in
                           VALUE2,                  ! 2nd longword of h float value for string
                           VALUE3,                  ! 3rd longword of h float value
                           VALUE4) :                ! 4th longword of h float value
                           NOVALUE =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine takes an h float number and formats it as the BASIC PRINT
!       statement would, except without leading and trailing spaces,
!       and gives that value to the destination string.
!       Note that this routine violates the calling standard by accepting and
!       calling a routine with double floating passed by value.
!
! FORMAL PARAMETERS:
!
!       STRING.wt.dx            pointer to input string descriptor
!       VALUE.rg.v              value of a double number
!       (VALUE1, VALUE2, VALUE3, & VALUE4 used to pick up the 4 words of h float value)
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       NONE
!
! SIDE EFFECTS:
!
!       This routine calls the conversion routine and so may signal any of its
!       errors and have any of its side effects.  In particular, the conversion
!       routine calls STR$ routines and so may allocate or deallocate
!       dynamic string space, or write lock a string for a short time.
!
!--

    BEGIN

    MAP
        STRING : REF BLOCK [8,BYTE];

    LOCAL
        STR_LENGTH : WORD;                          ! conversion rtn returns len

    BAS$CVT_OUT_H_G (VALUE1,                        ! convert this value to string
                     strip_spaces,                  ! set flag to strip spaces
                     STR_LENGTH,                    ! return bytes needed for str
                     STRING [0,0,0,0]);             ! return string
```

```
;  425          0791  2                                                  ! default # of digits
;  426          0792  2
;  427          0793  2       RETURN;
;  428          0794  1       END;                             !End of BAS$STR_H



                                                 0000 00000          .ENTRY  BAS$STR_H, Save nothing                  ; 0734
                          5E                  04   C2 00002          SUBL2   #4, SP                                   ;
                                       04     AC   DD 00005          PUSHL   STRING                                   ; 0790
                                       04     AE   9F 00008          PUSHAB  STR_LENGTH                               ; 0787
                                              01   DD 0000B          PUSHL   #1                                       ; 0790
                                       08     AC   9F 0000D          PUSHAB  VALUE1                                   ; 0787
                  00000000G   00              04   FB 00010          CALLS   #4, BAS$CVT_OUT_H_G                      ; 0790
                                                   04 00017          RET                                              ; 0794

; Routine Size:  24 bytes,    Routine Base:  _BAS$CODE + 0080
```

```
1 GLOBAL ROUTINE BAS$STR_P (                     ! packed number to a string
1                              STRING,           ! Address of destination descriptor
1                              VALUE) :          ! Find numeric value of this string
1                              NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine takes a packed decimal number and formats it as the BASIC
1 !       PRINT statement would without leading and trailing spaces
1 !       and gives that value to the destination string.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       STRING.wt.dx            pointer to input string descriptor
1 !       VALUE.rp.dsd            desc of packed decimal number
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       This routine calls a conversion routine and so may signal any of its errors
1 !       or have any of its side effects.  In particular, the conversion routine
1 !       calls STR$ routines and so may allocate or deallocate dynamic string
1 !       space, or write lock a string for a time.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2       STRING : REF BLOCK [8,BYTE],
2       VALUE : REF BLOCK [12,BYTE];
2
2     LOCAL
2       STR_LENGTH : WORD;                              ! conversion rtn returns len
2
2     BAS$CVT_OUT_P_G (.VALUE,                          ! convert this value to string
2                   strip_spaces,                       ! set flag to strip spaces
2                   STR_LENGTH,                         ! return bytes needed for str
2                   STRING [0,0,0,0]);                  ! descriptor of result string
2                                                       ! no scale to cvt
2                                                       ! default # of digits
2
2     RETURN;
1     END;                                              !End of BAS$STR_P
```

```
                                  0000 00000        .ENTRY  BAS$STR_P, Save nothing                    ; 0795
                    5E          04  C2 00002        SUBL2   #4, SP                                     ;
                          04    AC  DD 00005        PUSHL   STRING                                     ; 0846
                          04    AE  9F 00008        PUSHAB  STR_LENGTH                                 ; 0843
                                01  DD 0000B        PUSHL   #1                                         ; 0846
                          08    AC  DD 0000D        PUSHL   VALUE                                      ;
          00000000G 00          04  FB 00010        CALLS   #4, BAS$CVT_OUT_P_G                        ;
                                    04 00017        RET                                                ; 0851

; Routine Size:  24 bytes,    Routine Base:  _BAS$CODE + 0098
;  488          0852  1 END                                  !End of module
;  489          0853  0 ELUDOM



;                                 PSECT SUMMARY
;
;       Name                     Bytes                        Attributes
;
;  _BAS$CODE                         176  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)




;                                  COMMAND QUALIFIERS

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

; Size:          176 code + 0 data bytes
; Run Time:         00:06.6
; Elapsed Time:     00:15.4
; Lines/CPU Min:     7766
; Lexemes/CPU-Min: 20922
; Memory Used:  38 pages
; Compilation Complete
```

0032 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASUDFWF
LIS

BASSTRING
LIS

BASTERMIO
LIS

BASTRM
LIS

BASTAB
LIS

BASUDFRM
LIS

BASSYS
LIS

BASSTR
LIS

BASUDFRL
LIS