```
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     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    RRRRRRRRRRR       TTT        LLL
BBBBBBBBBBBB       AAA         AAA     SSSSSSSSS    RRRRRRRRRRR       TTT        LLL
BBBBBBBBBBBB       AAA         AAA     SSSSSSSSS    RRRRRRRRRRR       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  SSSSSSSSSSS  RRR         RRR    TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA         AAA  SSSSSSSSSSS  RRR         RRR    TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA         AAA  SSSSSSSSSSS  RRR         RRR    TTT        LLLLLLLLLLLLLLL
```

FILEID**BASPOWRR

```
BBBBBBBB      AAAAAA      SSSSSSSS  PPPPPPPP      000000    WW      WW  RRRRRRRR    RRRRRRRR
BBBBBBBB      AAAAAA      SSSSSSSS  PPPPPPPP      000000    WW      WW  RRRRRRRR    RRRRRRRR
BB      BB   AA      AA  SS        PP      PP  OO      OO  WW      WW  RR      RR  RR      RR
BB      BB   AA      AA  SS        PP      PP  OO      OO  WW      WW  RR      RR  RR      RR
BB      BB   AA      AA  SS        PP      PP  OO      OO  WW      WW  RR      RR  RR      RR
BB      BB   AA      AA  SS        PP      PP  OO      OO  WW      WW  RR      RR  RR      RR
BBBBBBBB     AA      AA    SSSSSS  PPPPPPPP    OO      OO  WW      WW  RRRRRRRR    RRRRRRRR
BBBBBBBB     AA      AA    SSSSSS  PPPPPPPP    OO      OO  WW      WW  RRRRRRRR    RRRRRRRR
BB      BB   AAAAAAAAAA          SS  PP        OO      OO  WW  WW  WW  RR  RR      RR  RR
BB      BB   AAAAAAAAAA          SS  PP        OO      OO  WW  WW  WW  RR  RR      RR  RR
BB      BB   AA      AA          SS  PP        OO      OO  WWWW  WWWW  RR    RR    RR    RR      ....
BB      BB   AA      AA          SS  PP        OO      OO  WWWW  WWWW  RR    RR    RR    RR      ....
BBBBBBBB     AA      AA  SSSSSSSS    PP          000000    WW      WW  RR      RR  RR      RR      ....
BBBBBBBB     AA      AA  SSSSSSSS    PP          000000    WW      WW  RR      RR  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
```

Table of contents

```
0000     1          .TITLE  BAS$POWRR               ; BASIC real ** real routine
0000     2          .IDENT  /1-005/         ; File: BASPOWRR.MAR Edit: RNH1005
0000     3
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; FACILITY: Basic Support Library
0000    31 ;
0000    32 ; ABSTRACT:
0000    33 ;
0000    34 ;       This module contains entry points to support exponentiation
0000    35 ;       (** or ^) in BASIC-PLUS-2 for FLOATING ** FLOATING.
0000    36 ;
0000    37 ; ENVIRONMENT: User Mode, AST Reentrant
0000    38 ;
0000    39 ;--
0000    40 ; AUTHOR: R. Will        , CREATION DATE: 22-NOV-78
0000    41 ;
0000    42 ; MODIFIED BY:
0000    43 ;
0000    44 ; R. Will,        : VERSION 01
0000    45 ; 1-01  - Original
0000    46 ; 1-02  - Fix comments, change BRW to JMP.  RW 7-Dec-78
0000    47 ; 1-003 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000    48 ; 1-004 - Redo case analysis for base leg 0 for compatability
0000    49 ;          with the PDP-11.  JBS 24-APR-1979
0000    50 ; 1-005 - Change shared external references to G^ RNH 25-Sep-81
```

```
                     52          .SBTTL  DECLARATIONS
                     53 ;
                     54 ; INCLUDE FILES:
                     55 ;
                     56
                     57 ;
                     58 ; EXTERNAL DECLARATIONS:
                     59 ;
                     60          .DSABL  GBL                     ; Prevent undeclared
                     61                                          ; symbols from being
                     62                                          ; automatically global.
                     63
                     64          .EXTRN  OTS$POWRR               ; OTS$ real ** real exponentiation
                     65          .EXTRN  OTS$POWRJ               ; OTS$ real ** int exponentation
                     66          .EXTRN  BAS$K_DIVBY_ZER         ; Divide by Zero
                     67          .EXTRN  BAS$K_ILLARGLOG         ; Illegal argument in LOG
                     68          .EXTRN  BAS$$STOP               ; Error reporting routine
                     69
                     70 ;
                     71 ; MACROS:
                     72 ;
                     73
                     74 ;
                     75 ; EQUATED SYMBOLS:
                     76 ;
                     77
                     78 ;
                     79 ; OWN STORAGE:
                     80 ;
                     81
                     82 ;
                     83 ; PSECT DECLARATIONS:
                     84 ;
                     85          .PSECT _BAS$CODE PIC, USR, CON, REL, LCL, SHR, -
                     86                      EXE, RD, NOWRT, LONG
                     87
```

```
                                        .SBTTL  BAS$POWRR  - BASIC floating ** floating
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       This routine takes BASE ** EXP, using the following table
                                ;       for unusual cases:
                                ;
                                ;       BASE > 0                                Call OTS$POWRR, normal case.
                                ;       BASE = 0, EXP > 0                       Return 0.0.
                                ;       BASE = 0, EXP = 0                       Return 1.0.
                                ;       BASE = 0, EXP < 0                       Error: divide by zero
                                ;       BASE < 0, EXP even integer              Call OTS$POWRJ with -BASE
                                ;       BASE < 0, EXP odd integer               Call OTS$POWRJ with -BASE, negate result
                                ;       BASE < 0, EXP not integer               Error: illegal argument in LOG.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       CALL result.wf.v = BAS$POWRR (base.rf.v, exponent.rf.v)
                                ;
                                ; INPUT PARAMETERS:
                                ;
                00000004        ;       base = 4
                00000008        ;       exponent = 8
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       NONE
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE
                                ;
                                ; FUNCTION VALUE:
                                ; COMPLETION CODES:
                                ;
                                ;       floating result of exponentiation
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       Will signal Divide By Zero or Illegal argument in LOG if its
                                ;       arguments are bad, and OTS$POWRR and OTS$POWRJ may also signal.
                                ;
                                ;--
                                
                0000'           BAS$POWRR::     .MASK OTS$POWRR         ; Entry point
                                                                        ; Since this routine uses no
                                                                        ; registers and usually transfers
                                                                        ; control to OTS$POWRR, we copy
                                                                        ; its register save mask and then
                                                                        ; JMP past its save mask and only
                                                                        ; save the registers once
        04 AC     53            TSTF    base(AP)                      ; Test base relationship to zero
           06     15            BLEQ    1$                            ; If base leq 0, do case analysis
```

```
                00000002'GF   17  0007   146          JMP     G^OTS$POWRR+2           ; Transfer control to the OTS$
                              0007   147                                              ; routine to do exponentiation
                              0007   148 ;+
                              0007   149 ; Come here if the base is less than or equal to zero.  We must filter
                              0007   150 ; several special cases, as described above.
                              0007   151 ;-
                        2E    0007   152 1$:      BEQL    4$                      ; Branch if base = 0
    50   50   08   00   08 AC 54   0009   153          EMODF   exponent(AP), #0, #1, R0, R0
                        1A    0010   154          BNEQ    3$                      ; Branch if exponent is not integer
                              0012   155 ;+
                              0012   156 ; The base is less than zero and the exponent is an integer.
                              0012   157 ; BASIC defines this as working the same way as if an integer was
                              0012   158 ; in the expression (making a floating variable which happens to
                              0012   159 ; contain an integer value equivalent to an integer variable).
                              0012   160 ;-
                   50   08 AC 4A   0012   161          CVTFL   exponent(AP), R0        ; Convert exponent to integer
                        50    DD   0016   162          PUSHL   R0                      ; Save for even/odd test
                        50    DD   0018   163          PUSHL   R0                      ; Stack as parameter to OTS$POWRJ
                      7E   04 AC 52   001A   164          MNEGF   base(AP), -(SP)         ; Stack -base also
             00000000'GF   02    FB   001E   165          CALLS   #2, G^OTS$POWRJ         ; Call integer power routines
                     03 8E    E9   0025   166          BLBC    (SP)+,2$                ; Branch if exponent even
                50   50    52   0028   167          MNEGF   R0, R0                  ; Exponent odd, negate the result
                        04    002B   168 2$:      RET                             ; and return with it.
                              002C   169 ;+
                              002C   170 ; Come here if the base is less than zero but the exponent is not
                              002C   171 ; an integer.  BASIC defines this as an error.
                              002C   172 ;-
                   7E   00'8F 9A   002C   173 3$:      MOVZBL  #BAS$K_ILLARGLOG, -(SP) ; Illegal Argument in LOG
             00000000'GF   01    FB   0030   174          CALLS   #1, G^BAS$$STOP         ; Never return.
                              0037   175 ;+
                              0037   176 ; Come here if the base is equal to zero.  The value we return depends
                              0037   177 ; upon the sign of the exponent.
                              0037   178 ;-
                08 AC    53   0037   179 4$:      TSTF    exponent(AP)            ; Test the exponent against zero
                   09    19   003A   180          BLSS    6$                      ; Branch if exponent lss 0
                   03    13   003C   181          BEQL    5$                      ; Branch if exponent is 0
                              003E   182 ;+
                              003E   183 ; Come here if the base is zero and the exponent is greater than zero.
                              003E   184 ; BASIC defines this as 0.0.
                              003E   185 ;-
                   50   D4   003E   186          CLRF    R0                      ; R0 = 0.0
                        04    0040   187          RET                             ; Return to caller
                              0041   188 ;+
                              0041   189 ; Come here if the base is zero and the exponent is zero.  BASIC defines
                              0041   190 ; this as 1.0.
                              0041   191 ;-
             50   08   50   0041   192 5$:      MOVF    #1, R0                  ; R0 = 1.0
                        04    0044   193          RET                             ; Return to caller.
                              0045   194 ;+
                              0045   195 ; Come here if the base is zero and the exponent is less than zero.
                              0045   196 ; BASIC defines this as an error.
                              0045   197 ;-
          7E   00'8F    9A   0045   198 6$:      MOVZBL  #BAS$K_DIVBY_ZER, -(SP) ; Divide by zero
00000000'GF   01    FB   0049   199          CALLS   #1, G^BAS$$STOP         ; Report error, never return.
                              0050   200 ;
                              0050   201          .END
```

```
BAS$$STOP           ********    X   00
BAS$K_DIVBY_ZER     ********    X   00
BAS$K_ILLARGLOG     ********    X   00
BAS$POWRR           00000000 RG     01
BASE              = 00000004
EXPONENT          = 00000008
OTS$POWRJ           ********    X   00
OTS$POWRR           ********    X   00
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_BAS$CODE	00000056 (86.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	38	00:00:00.08	00:00:00.58
Command processing	127	00:00:00.49	00:00:02.20
Pass 1	69	00:00:00.48	00:00:01.26
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	48	00:00:00.40	00:00:00.82
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	289	00:00:01.50	00:00:04.90

The working set limit was 750 pages.
2270 bytes (5 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 8 non-local and 6 local symbols.
201 source lines were read in Pass 1, producing 8 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

```
MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:BASPOWRR/OBJ=OBJ$:BASPOWRR MSRC$:BASPOWRR/UPDATE=(ENH$:BASPOWRR)
```

BASOPEN
LIS

BASPOWJJ
LIS

BASPOS
LIS

BASPOWDJ
LIS

BASOPENDE
LIS

BASPOWGG
LIS

BASPOWHH
LIS

BASPOWRJ
LIS

BASPOWII
LIS

BASPURIOB
LIS

BASPOWDD
LIS

BASOPENZE
LIS

BASPOWDR
LIS

BASPOWGJ
LIS

BASPOWRD
LIS

BASPOWHJ
LIS

BASPOWRR
LIS