```
                                                       M  1




BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  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     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
```

```
                                                                   E 11
**FILE**ID**BASSCALE
```

```
BBBBBBBB      AAAAAA      SSSSSSSS    SSSSSSSS    CCCCCCCC    AAAAAA    LL          EEEEEEEEEE
BBBBBBBB      AAAAAA      SSSSSSSS    SSSSSSSS    CCCCCCCC    AAAAAA    LL          EEEEEEEEEE
BB       BB AA      AA  SS          SS           CC         AA      AA  LL          EE
BB       BB AA      AA  SS          SS           CC         AA      AA  LL          EE
BB       BB AA      AA  SS          SS           CC         AA      AA  LL          EE
BB       BB AA      AA  SS          SS           CC         AA      AA  LL          EE
BBBBBBBB    AA      AA    SSSSSS      SSSSSS     CC         AA      AA  LL          EEEEEEEE
BBBBBBBB    AA      AA    SSSSSS      SSSSSS     CC         AA      AA  LL          EEEEEEEE
BB       BB AAAAAAAAAA          SS          SS   CC         AAAAAAAAAA  LL          EE
BB       BB AAAAAAAAAA          SS          SS   CC         AAAAAAAAAA  LL          EE
BB       BB AA      AA          SS          SS   CC         AA      AA  LL          EE
BB       BB AA      AA          SS          SS   CC         AA      AA  LL          EE              ....
BBBBBBBB    AA      AA  SSSSSSSS    SSSSSSSS      CCCCCCCC  AA      AA  LLLLLLLLLL  EEEEEEEEEE      ....
BBBBBBBB    AA      AA  SSSSSSSS    SSSSSSSS      CCCCCCCC  AA      AA  LLLLLLLLLL  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
```

```
0 MODULE BAS$SCALE (                                   ! Support scaling
0               IDENT = '1-008'                        ! File: BASSCALE.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-PLUS-2 Miscellaneous
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains compiled code support routines
1 !       for scaling and descaling.
1 !
1 ! ENVIRONMENT:  VAX-11 user mode
1 !
1 ! AUTHOR: John Sauter, CREATION DATE: 08-MAY-1979
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.
1 ! 1-002 - Add BAS$$SCALE_R1.  JBS 29-MAY-1979
1 ! 1-003 - Use BAS$$COPY_D to fetch double-precision numbers.
1 !         JBS 29-MAY-1979
1 ! 1-004 - Add BAS$$SCALE_L_R1.  JBS 26-JUN-1979
1 ! 1-005 - Change MTH$FLOOR_D to MTH$DFLOOR.  JBS 27-JUL-1979
1 ! 1-006 - Change BAS$$COPY_D to BAS$$COPY_D_R1.  JBS 23-AUG-1979
1 ! 1-007 - Change MTH$DFLOOR to MTH$DINT.  JBS 19-DEC-1979
1 ! 1-008 - Remove the check for a BASIC frame from BAS$SCALE_D_R1
1 !         and BAS$DSCALE_D_R1.  (This is so BAS$CHANGE can call
1 !         them.)  PLL 22-MAY-1981
1 !--
1
```

```
;   58          0058  1 !<BLF/PAGE>
;    60        0059  1  !
;    61        0060  1  ! SWITCHES:
;    62        0061  1  !
;    63        0062  1
;    64        0063  1  SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    65        0064  1
;    66        0065  1  !
;    67        0066  1  ! LINKAGES:
;    67        0067  1  !
;    69        0068  1
;    70        0069  1  REQUIRE 'RTLIN:BASLNK';                          ! BASIC linkages
;    71        0146  1
;    72        0147  1  LINKAGE
;    73        0148  1      COPY_JSB = JSB (REGISTER = 0, REGISTER = 1) :        !
;    74        0149  1      NOTUSED (2, 3, 4, 5, 6, 7, 8, 9, 10, 11);
;    75        0150  1
;    76        0151  1  !
;    77        0152  1  ! TABLE OF CONTENTS:
;    78        0153  1  !
;    79        0154  1
;    80        0155  1  FORWARD ROUTINE
;    81        0156  1      BAS$SCALE_D_R1 : NOVALUE BAS$SCALE_LINK,     ! Scale a number
;    82        0157  1      BAS$DSCALE_D_R1 : NOVALUE BAS$SCALE_LINK,    ! Descale a number
;    83        0158  1      BAS$$SCALE_R1 : NOVALUE BAS$SCALE_JSB,       ! Fetch the scale as double
;    84        0159  1      BAS$$SCALE_L_R1 : BAS$SCALE_JSB;             ! Fetch the scale power
;    85        0160  1
;    86        0161  1  !
;    87        0162  1  ! INCLUDE FILES:
;    88        0163  1  !
;    89        0164  1
;    90        0165  1  REQUIRE 'RTLIN:BASFRAME';                        ! BSF symbols
;    91        0368  1
;    92        0369  1  LIBRARY 'RTLSTARLE';                             ! symbols for strings
;    93        0370  1
;    94        0371  1  REQUIRE 'RTLIN:RTLPSECT';                        ! macros for defing psects
;    95        0466  1
;    96        0467  1  !
;    97        0468  1  ! MACROS:
;    98        0469  1  !
;    99        0470  1  !       NONE
;   100        0471  1  !
;   101        0472  1  ! EQUATED SYMBOLS:
;   102        0473  1  !
;   103        0474  1  !       NONE
;   104        0475  1  !
;   105        0476  1  ! PSECTS:
;   106        0477  1  !
;   107        0478  1  DECLARE_PSECTS (BAS);                            ! declare psects for BAS$ facility
;   108        0479  1  !
;   109        0480  1  ! OWN STORAGE:
;   110        0481  1  !
;   111        0482  1  !       NONE
;   112        0483  1  !
;   113        0484  1  ! EXTERNAL REFERENCES:
;   114        0485  1  !
;   115        0486  1
;   116        0487  1  EXTERNAL ROUTINE
;  117        0488  1      MTH$DINT,                                  ! Remove fraction part
;  118        0489  1      BAS$$COPY_D_R1 : COPY_JSB NOVALUE,         ! Move a D_float number
;  119        0490  1      BAS$$MULD,                                 ! Multiply D_float numbers
;  120        0491  1      BAS$$DIVD,                                 ! Divide D_float numbers
;  121        0492  1      BAS$HANDLER;                               ! BASIC frame marker
;  122        0493  1
```

```
1 GLOBAL ROUTINE BAS$SCALE_D_R1 (                   ! Scale a value
1         VALHI,                                    ! High 32-bits of value
1         VALLO                                     ! Low 32-bits of value
1     ) : NOVALUE BAS$SCALE_LINK =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Scale a value.  This is done by multiplying by the scale factor
1 !       and integerizing the result.
1 !
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       VAL.rd.v        The D_floating value to be scaled, presented to
1 !                       BLISS as VALHI and VALLO.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       The scale factor, in the major frame.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       The scaled, double precision number.
1 !
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       May get arithmetic faults.
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         BSF$A_MAJOR_STG : REF BLOCK [0, BYTE] FIELD (BSF$MAJOR_FRAME),
2         BSF$A_MINOR_STG,
2         BSF$A_TEMP_STG;
2
2     LOCAL
2         VAL : VECTOR [2, LONG];                   ! D_floating value being manipulated
2
2 !+
2 ! Save the input value, so we get some registers to work with.
2 !-
2     VAL [0] = .VALHI;
2     VAL [1] = .VALLO;
2
3         BEGIN
```

```
;  181          0551  3 !+
;  182          0552  3 ! Multiply the argument by the scale factor, and then integerize.
;  183          0553  3 !-
;  184          0554  3
;  185          0555  3             LOCAL
;  186          0556  3                 MAJOR_FMP : REF BLOCK [, BYTE] FIELD (BSF$FCD);
;  187          0557  3
;  188          0558  3             MAJOR_FMP = BSF$A_MAJOR_STG [BSF$FRAME_BASE];
;  189          0559  3             BAS$$MULD (VAL [0], MAJOR_FMP [BSF$D_SCALE_DOU], VAL [0]);
;  190          0560  3             MTH$DINT (VAL [0]);
;  191          0561  2             END;
;  192          0562  2         RETURN;
;  193          0563  1         END;                                    ! of BAS$SCALE_D_R1

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

                                                                          .EXTRN  MTH$DINT, BAS$$COPY_D_R1
                                                                          .EXTRN  BAS$$MULD, BAS$$DIVD
                                                                          .EXTRN  BAS$HANDLER

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

                                        5E          04  C2 00000 BAS$SCALE_D_R1::
                                                                          SUBL2   #4, SP                                   ; 0494
                                                    50  DD 00003          PUSHL   VALHI                                    ; 0547
                                   04   AE          51  D0 00005          MOVL    VALLO, VAL+4                             ; 0548
                                        50    00C3  CB  9E 00009          MOVAB   195(R11), MAJOR_FMP                      ; 0558
                                                    5E  DD 0000E          PUSHL   SP                                       ; 0559
                                                D0  A0  9F 00010          PUSHAB  -48(MAJOR_FMP)                           ;
                                                08  AE  9F 00013          PUSHAB  VAL                                      ;
                              00000000G   00        03  FB 00016          CALLS   #3, BAS$$MULD                            ;
                                                    5E  DD 0001D          PUSHL   SP                                       ; 0560
                              00000000G   00        01  FB 0001F          CALLS   #1, MTH$DINT                             ;
                                        5E          08  C0 00026          ADDL2   #8, SP                                   ; 0563
                                                        05 00029          RSB                                              ;

; Routine Size:  42 bytes,    Routine Base:  _BAS$CODE + 0000


;  194          0564  1
```

```
1 GLOBAL ROUTINE BAS$DSCALE_D_R1 (                      ! Descale a value
'         VALHI,                                        ! High 32-bits of value
1         VALLO                                         ! Low 32-bits of value
1     ) : NOVALUE BAS$SCALE_LINK =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         Descale a value.  This is done by dividing by the scale factor.
1 !
1 !
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         VAL.rd.v         The D_floating value to be descaled, presented
1 !                          to BLISS as VALHI and VALLO.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         The scale factor, in the major frame.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !         The descaled, double precision number.
1 !
1 ! COMPLETION CODES:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         May get arithmetic faults.
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         BSF$A_MAJOR_STG : REF BLOCK [0, BYTE] FIELD (BSF$MAJOR_FRAME),
2         BSF$A_MINOR_STG,
2         BSF$A_TEMP_STG;
2
2
2     LOCAL
2         VAL : VECTOR [2, LONG];                   ! D_floating value being manipulated
2
2 !+
2 ! Save the input value, so we get some registers to work with.
2 !-
2     VAL [0] = .VALHI;
2     VAL [1] = .VALLO;
2
3         BEGIN
```

```
;  253          0622  3 !+
;  254          0623  3 ! Divide the argument by the scale factor, then load into R0 and R1.
;  255          0624  3 !-
;  256          0625  3
;  257          0626  3         LOCAL
;  258          0627  3             MAJOR_FMP : REF BLOCK [, BYTE] FIELD (BSF$FCD);
;  259          0628  3
;  260          0629  3         MAJOR_FMP = BSF$A_MAJOR_STG [BSF$FRAME_BASE];
;  261          0630  3         BAS$$DIVD (MAJOR_FMP [BSF$D_SCALE_DOU], VAL [0], VAL [0]);
;  262          0631  4         BEGIN
;  263          0632  4
;  264          0633  4         REGISTER
;  265          0634  4             R0 = 0,
;  266          0635  4             R1 = 1;
;  267          0636  4
;  268          0637  4         R0 = .VAL [0];
;  269          0638  4         R1 = .VAL [1];
;  270          0639  3         END;
;  271          0640  2         END;
;  272          0641  2     RETURN;
;  273          0642  1     END;                                        ! of BAS$DSCALE_D_R1



                       5E             04  C2 00000 BAS$DSCALE_D_R1::
                                                            SUBL2   #4, SP                                        ; 0565
                                      50  DD 00003          PUSHL   VALHI                                         ; 0618
                  04   AE             51  D0 00005          MOVL    VALLO, VAL+4                                  ; 0619
                       50      00C3   CB  9E 00009          MOVAB   195(R11), MAJOR_FMP                           ; 0629
                                      5E  DD 0000E          PUSHL   SP                                            ; 0630
                                 04   AE  9F 00010          PUSHAB  VAL                                           ;
                                 D0   A0  9F 00013          PUSHAB  -48(MAJOR_FMP)                                ;
             00000000G 00             03  FB 00016          CALLS   #3, BAS$$DIVD                                 ;
                       50             8E  7D 0001D          MOVQ    VAL, R0                                       ; 0637
                                          05 00020          RSB                                                   ; 0642

; Routine Size:  33 bytes,    Routine Base:  _BAS$CODE + 002A

;  274          0643  1
```

```
1 GLOBAL ROUTINE BAS$$SCALE_R1 (                           ! Fetch the scale
1           FMP                                             ! Frame containing scale
1     ) : NOVALUE BAS$SCALE_JSB =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         Fetch the scale value from a frame.  This routine is for use by
1 !         math routines to fetch the scale from their caller.  If the
1 !         frame is not a BASIC frame, a double-precision 1.0 is returned.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         FMP.ra.v        The (possibly BASIC) frame containing the scale
1 !                         factor.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         The scale factor, in the major frame.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !         The scale factor, as a double-precision number.
1 !
1 ! COMPLETION CODES:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         May get arithmetic faults.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         FMP : REF BLOCK [, BYTE] FIELD (BSF$FCD);
2
2 !+
2 ! If this is not a BASIC frame, return 1.0.
2 !-
2
3     IF (.FMP [BSF$A_HANDLER] NEQA BAS$HANDLER)
2     THEN
3         BEGIN
3
3         REGISTER
3             R0 = 0,
3             R1 = 1;
3
3         BUILTIN
3             CVTLD;
```

```
;  333          0701  3
;  334          0702  3             CVTLD (%REF (1), R0);
;  335          0703  3             END
;  336          0704  2         ELSE
;  337          0705  3             BEGIN
;  338          0706  3  !+
;  339          0707  3  ! Otherwise return the real scale factor.
;  340          0708  3  !-
;  341          0709  3
;  342          0710  3             LOCAL
;  343          0711  3                 MAJOR_FMP : REF BLOCK [, BYTE] FIELD (BSF$FCD),
;  344          0712  3                 BSF$A_MAJOR_STG : REF BLOCK [, BYTE] FIELD (BSF$MAJOR_FRAME),
;  345          0713  3                 VAL : VECTOR [2, LONG];
;  346          0714  3
;  347          0715  3             BS'$A_MAJOR_STG = .FMP [BSF$A_BASE_R11];
;  348          0716  3             MAJOR_FMP = BSF$A_MAJOR_STG [BSF$FRAME_BASE];
;  349          0717  3             BAS$$COPY_D_R1 (MAJOR_FMP [BSF$D_SCALE_DOU], VAL);
;  350          0718  4             BEGIN
;  351          0719  4
;  352          0720  4             REGISTER
;  353          0721  4                 R0 = 0,
;  354          0722  4                 R1 = 1;
;  355          0723  4
;  356          0724  4             R0 = .VAL [0];
;  357          0725  4             R1 = .VAL [1];
;  358          0726  3             END;
;  359          0727  2             END;
;  360          0728  2
;  361          0729  2         RETURN;
;  362          0730  1         END;                                    ! of BAS$SCALE_R1



                          5E              08  C2 00000 BAS$$SCALE_R1::
                                                               SUBL2    #8, SP                                          : 0644
                          51              50  D0 00003         MOVL     R0, R1                                          :
                          50 00000000G    00  9E 00006         MOVAB    BAS$HANDLER, R0                                 : 0691
                          50              61  D1 0000D         CMPL     (FMP), R0                                       :
                                          05  13 00010         BEQL     1$                                              :
                          50              01  6E 00012         CVTLD    #1, R0                                          : 0702
                                          15  11 00015         BRB      2$                                              : 0691
                          50          F4  A1  D0 00017 1$:     MOVL     -12(FMP), BSF$A_MAJOR_STG                       : 0715
                          51              6E  9E 0001B         MOVAB    VAL, R1                                         : 0717
                          50        0093  C0  9E 0001E         MOVAB    147(R0), R0                                     :
                             00000000G    00  16 00023         JSB      BAS$$COPY_D_R1                                  :
                          50              6E  7D 00029         MOVQ     VAL, R0                                         : 0724
                          5E              08  C0 0002C 2$:     ADDL2    #8, SP                                          : 0730
                                              05 0002F         RSB                                                      :

; Routine Size:  48 bytes,    Routine Base:  _BAS$CODE + 004B


;  363          0731  1
```

```
1 GLOBAL ROUTINE BAS$SCALE_L_R1 (                      ! Fetch the scale
1           FMP                                        ! Frame containing scale
1     ) : BAS$SCALE_JSB =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         Fetch the scale value from a frame.  This routine is for use by
1 !         math routines to fetch the scale from their caller.  If the
1 !         frame is not a BASIC frame, 0 is returned.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         FMP.ra.v         The (possibly BASIC) frame containing the scale
1 !                          factor.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         The scale factor, in the major frame.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !         The scale factor, as an integer power of 10.
1 !
1 ! COMPLETION CODES:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         FMP : REF BLOCK [, BYTE] FIELD (BSF$FCD);
2
2 !+
2 ! If this is not a BASIC frame, return 0.
2 !-
2
3     IF (.FMP [BSF$A_HANDLER] NEQA BAS$HANDLER)
2     THEN
2         0
2     ELSE
3         BEGIN
3 !+
3 ! Otherwise return the real scale factor.
3 !-
3
3         LOCAL
```

```
;  422          0789  3              MAJOR_FMP : REF BLOCK [, BYTE] FIELD (BSF$FCD),
;  423          0790  3              BSF$A_MAJOR_STG : REF BLOCK [, BYTE] FIELD (BSF$MAJOR_FRAME),
;  424          0791  3              VAL : VECTOR [2, LONG];
;  425          0792  3
;  426          0793  3          BSF$A_MAJOR_STG = .FMP [BSF$A_BASE_R11];
;  427          0794  3          MAJOR_FMP = BSF$A_MAJOR_STG [BSF$FRAME_BASE];
;  428          0795  3          .MAJOR_FMP [BSF$B_SCA_V_DOU]
;  429          0796  3          END
;  430          0797  3
;  431          0798  1      END;                                        ! of BAS$SCALE_D_R1



                          5E             08  C2 00000 BAS$$SCALE_L_R1::
                                                               SUBL2    #8, SP                              ; 0732
                          51 00000000G   00  9E 00003          MOVAB    BAS$HANDLER, R1                     ; 0779
                          51             60  D1 0000A          CMPL     (FMP), R1                           ;
                                         04  13 0000D          BEQL     1$                                  ;
                                         50  D4 0000F          CLRL     R0                                  ;
                                         0D  11 00011          BRB      2$                                  ;
                          50        F4   A0  D0 00013 1$:      MOVL     -12(FMP), BSF$A_MAJOR_STG           ; 0793
                          50      00C3   C0  9E 00017          MOVAB    195(R0), MAJOR_FMP                  ; 0794
                          50        CD   A0  98 0001C          CVTBL    -51(MAJOR_FMP), R0                  ; 0795
                          5E             08  C0 00020 2$:      ADDL2    #8, SP                              ; 0798
                                             05 00023          RSB                                          ;

; Routine Size:  36 bytes,    Routine Base:  _BAS$CODE + 007B


;  432          0799  1
;  433          0800  1 END
;  434          0801  1
;  435          0802  0 ELUDOM




;                               PSECT SUMMARY
;
;      Name                           Bytes                          Attributes
;
;  _BAS$CODE                             159  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         0         0       581          00:01.0
;                                          COMMAND QUALIFIERS

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

; Size:          159 code + 0 data bytes
; Run Time:         00:07.9
; Elapsed Time:     00:21.2
; Lines/CPU Min:     6075
; Lexemes/CPU-Min: 21098
; Memory Used:  57 pages
; Compilation Complete
```

0031 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY