```
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
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
```

```
BBBBBBBB      AAAAAA      SSSSSSSS    CCCCCCCC  VV      VV  TTTTTTTTTT  RRRRRRRR    PPPPPPPP
BBBBBBBB      AAAAAA      SSSSSSSS    CCCCCCCC  VV      VV  TTTTTTTTTT  RRRRRRRR    PPPPPPPP
BB      BB   AA    AA   SS          CC          VV      VV      TT      RR      RR  PP      PP
BB      BB   AA    AA   SS          CC          VV      VV      TT      RR      RR  PP      PP
BB      BB   AA    AA   SS          CC          VV      VV      TT      RR      RR  PP      PP
BB      BB   AA    AA   SS          CC          VV      VV      TT      RR      RR  PP      PP
BBBBBBBB     AA    AA     SSSSSS    CC          VV      VV      TT      RRRRRRRR    PPPPPPPP
BBBBBBBB     AA    AA     SSSSSS    CC          VV      VV      TT      RRRRRRRR    PPPPPPPP
BB      BB   AAAAAAAAAA         SS  CC          VV      VV      TT      RR  RR      PP
BB      BB   AAAAAAAAAA         SS  CC          VV      VV      TT      RR  RR      PP
BB      BB   AA    AA           SS  CC            VV  VV        TT      RR    RR    PP          ....
BB      BB   AA    AA           SS  CC            VV  VV        TT      RR    RR    PP          ....
BBBBBBBB     AA    AA   SSSSSSSS      CCCCCCCC      VV        TT      RR      RR  PP          ....
BBBBBBBB     AA    AA   SSSSSSSS      CCCCCCCC      VV        TT      RR      RR  PP          ....
```

```
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
```

```
;    1          0001  0 %TITLE 'BAS$CVTRP - Convert real to packed'
;    2          0002  0 MODULE BAS$CVTRP (                              ! Convert real to packed
;    3          0003  0                   IDENT = '1-004'              ! File: BASCVTRP.B32 Edit: PLL1004
;    4          0004  0                   ) =
;    5          0005  1 BEGIN
;    6          0006  1 !
;    7          0007  1 !******************************************************************************
;    8          0008  1 !*                                                                            *
;    9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                   *
;   10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                     *
;   11          0011  1 !*  ALL RIGHTS RESERVED.                                                       *
;   12          0012  1 !*                                                                            *
;   13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;   14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;   15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;   16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;   17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;   18          0018  1 !*  TRANSFERRED.                                                               *
;   19          0019  1 !*                                                                            *
;   20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;   21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;   22          0022  1 !*  CORPORATION.                                                               *
;   23          0023  1 !*                                                                            *
;   24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;   25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                    *
;   26          0026  1 !*                                                                            *
;   27          0027  1 !*                                                                            *
;   28          0028  1 !******************************************************************************
;   29          0029  1 !
;   30          0030  1
;   31          0031  1 !++
;   32          0032  1 ! FACILITY:     Basic Language Support
;   33          0033  1 !
;   34          0034  1 ! ABSTRACT:
;   35          0035  1 !
;   36          0036  1 ! This module contains routines to convert real data types to packed decimal.
;   37          0037  1 ! It also contains routines to convert from packed to a real type.
;   38          0038  1 !
;   39          0039  1 ! These jacket routines are necessary because the OTS routines are JSB routines
;   40          0040  1 ! and use R9 to pass a parameter.  If an error occurs R9 will not automatically
;   41          0041  1 ! be restored and the Basic compiler expects R9 to point at some local storage.
;   42          0042  1 ! Note that CALL entry points cause R9 to be saved in the frame.
;   43          0043  1 !
;   44          0044  1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
;   45          0045  1 !
;   46          0046  1 ! AUTHOR: Pamela L. Levesque, CREATION DATE: 15-April-1982
;   47          0047  1 !
;   48          0048  1 ! MODIFIED BY:
;   49          0049  1 !
;   50          0050  1 ! 1-001 - Original.  PLL 15-Apr-1982
;   51          0051  1 ! 1-002 - Clean up some comments.  PLL 21-Apr-1982
;   52          0052  1 ! 1-003 - Add entry points for rounding.  PLL 7-Jun-1982
;   53          0053  1 ! 1-004 - Before reporting decimal overflow error, must check BASIC frame to
;   54          0054  1 !         ensure that "/OVERFLOW=NODEC" was not specified during the compile.
;   55          0055  1 !         DG 7-Mar-1984
;   56          0056  1 !--
;   57          0057  1
```

```
;    59          0058  1 %SBTTL 'Declarations'
;    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:
;    68          0067  1 !
;    69          0068  1
;    70          0069  1 LINKAGE
;    71          0070  1     JSB_CVT = JSB (REGISTER = 6, REGISTER = 7, REGISTER = 8, REGISTER = 9)
;    72          0071  1                   : PRESERVE (2, 3, 4, 5, 10, 11);
;    73          0072  1 !
;    74          0073  1 ! TABLE OF CONTENTS:
;    75          0074  1 !
;    76          0075  1
;    77          0076  1 FORWARD ROUTINE
;    78          0077  1
;    79          0078  1     BAS$CVTFP : NOVALUE,                    ! convert float to packed
;    80          0079  1     BAS$CVTDP : NOVALUE,                    ! convert double to packed
;    81          0080  1     BAS$CVTGP : NOVALUE,                    ! convert gfloat to packed
;    82          0081  1     BAS$CVTHP : NOVALUE,                    ! convert hfloat to packed
;    83          0082  1     BAS$CVTRFP : NOVALUE,                   ! convert float to packed (rounded)
;    84          0083  1     BAS$CVTRDP : NOVALUE,                   ! convert double to packed (rounded)
;    85          0084  1     BAS$CVTRGP : NOVALUE,                   ! convert gfloat to packed (rounded)
;    86          0085  1     BAS$CVTRHP : NOVALUE,                   ! convert hfloat to packed (rounded)
;    87          0086  1     BAS$CVTPF : NOVALUE,                    ! convert packed to float
;    88          0087  1     BAS$CVTPD : NOVALUE,                    ! convert packed to double
;    89          0088  1     BAS$CVTPG : NOVALUE,                    ! convert packed to gfloat
;    90          0089  1     BAS$CVTPH : NOVALUE;                    ! convert packed to hfloat
;    91          0090  1
;    92          0091  1 !
;    93          0092  1 ! INCLUDE FILES:
;    94          0093  1 !
;    95          0094  1
;    96          0095  1 LIBRARY 'RTLSTARLE';                        ! System symbols, typically from SYS$LIBRARY:STARLET.L32
;    97          0096  1
;    98          0097  1 REQUIRE 'RTLIN:RTLPSECT';                   ! Define PSECT declarations macros
;    99          0192  1 REQUIRE 'RTLIN:BASFRAME.REQ';               ! BSF symbols
;   100          0395  1
;   101          0396  1 !
;   102          0397  1 ! MACROS:
;   103          0398  1 !
;   104          0399  1
;   105          0400  1 MACRO
;   106        M 0401  1     FIND_FRAME (F) =
;   107        M 0402  1         BEGIN
;   108        M 0403  1
;   109        M 0404  1         BUILTIN
;   110        M 0405  1                 FP;                           ! Frame pointer
;   111        M 0406  1
;   112        M 0407  1         F = .FP;
;   113        M 0408  1             DO
;   114        M 0409  1                 BEGIN                         ! search back for Basic frame
;   115        M 0410  1                 F = .F [BSF$A_SAVED_FP];
```

```
;  116     M 0411  1                END
;  117     M 0412  1          UNTIL (.F [BSF$A_HANDLER] EQLA BAS$HANDLER OR
;  118     M 0413  1                 .F EQL 0);
;  119     M 0414  1
;  120     M 0415  1          END;
;  121       0416  1 %;
;  122       0417  1 !
;  123       0418  1 ! EQUATED SYMBOLS:
;  124       0419  1 !
;  125       0420  1 !          NONE
;  126       0421  1 !
;  127       0422  1 ! FIELDS:
;  128       0423  1 !
;  129       0424  1 !          NONE
;  130       0425  1 !
;  131       0426  1 ! PSECTS:
;  132       0427  1 !
;  133       0428  1 DECLARE_PSECTS (BAS);                   ! Declare PSECTs for BAS$ facility
;  134       0429  1 !
;  135       0430  1 ! OWN STORAGE:
;  136       0431  1 !
;  137       0432  1 !          NONE
;  138       0433  1 !
;  139       0434  1 ! EXTERNAL REFERENCES:
;  140       0435  1 !
;  141       0436  1
;  142       0437  1 EXTERNAL ROUTINE
;  143       0438  1
;  144       0439  1     BAS$HANDLER,
;  145       0440  1     OTS$CVTFP_R9 : JSB_CVT,             ! OTS conv float to packed
;  146       0441  1     OTS$CVTDP_R9 : JSB_CVT,             ! OTS conv dbl to packed
;  147       0442  1     OTS$CVTGP_R9 : JSB_CVT,             ! OTS conv gfloat to packed
;  148       0443  1     OTS$CVTHP_R9 : JSB_CVT,             ! OTS conv hfloat to packed
;  149       0444  1     OTS$CVTRFP_R9 : JSB_CVT,            ! OTS conv float to packed (rounded)
;  150       0445  1     OTS$CVTRDP_R9 : JSB_CVT,            ! OTS conv dbl to packed (rounded)
;  151       0446  1     OTS$CVTRGP_R9 : JSB_CVT,            ! OTS conv gfloat to packed (rounded)
;  152       0447  1     OTS$CVTRHP_R9 : JSB_CVT,            ! OTS conv hfloat to packed (rounded)
;  153       0448  1     OTS$CVTPF_R9 : JSB_CVT,             ! OTS conv packed to float
;  154       0449  1     OTS$CVTPD_R9 : JSB_CVT,             ! OTS conv packed to dbl
;  155       0450  1     OTS$CVTPG_R9 : JSB_CVT,             ! OTS conv packed to gfloat
;  156       0451  1     OTS$CVTPH_R9 : JSB_CVT,             ! OTS conv packed to hfloat
;  157       0452  1     BAS$$SIGNAL : NOVALUE;              ! signal non-fatal error
;  158       0453  1
;  159       0454  1 EXTERNAL LITERAL                                ! Condition value symbols
;  160       0455  1     BAS$K_DECERR : UNSIGNED (8);        ! decimal error or overflow
```

```
;  162          0456  1 %SBTTL 'BAS$CVT`FP - Convert float to packed'
;  163          0457  1 GLOBAL ROUTINE BAS$CVTFP (                                  ! Convert float to packed
;  164          0458  1           DEST,                                            ! place to store conv. number
;  165          0459  1           DESTLEN,                                         ! number of digits in dest
;  166          0460  1           SRC,                                             ! number to be converted
;  167          0461  1           SCALE                                            ! power of ten to mult src
;  168          0462  1       ) : NOVALUE =
;  169          0463  1
;  170          0464  1 !++
;  171          0465  1 ! FUNCTIONAL DESCRIPTION:
;  172          0466  1 !
;  173          0467  1 !       Converts a single floating number to packed.
;  174          0468  1 !
;  175          0469  1 ! CALLING SEQUENCE:
;  176          0470  1 !
;  177          0471  1 !       BAS$CVTFP (DEST.wp.r, DESTLEN.rl.v, SRC.rf.r, SCALE.rl.v)
;  178          0472  1 !
;  179          0473  1 ! FORMAL PARAMETERS:
;  180          0474  1 !
;  181          0475  1 !       DEST.wp.r       place to store the converted number
;  182          0476  1 !       DESTLEN.rl.v    number of digits in the destination
;  183          0477  1 !       SRC.rf.r        number to be converted
;  184          0478  1 !       SCALE.rl.v      power of ten by which the internal
;  185          0479  1 !                       representation of the sourc must be
;  186          0480  1 !                       multiplied to scale the same as the
;  187          0481  1 !                       internal representation of the dest.
;  188          0482  1 !
;  189          0483  1 ! IMPLICIT INPUTS:
;  190          0484  1 !
;  191          0485  1 !       NONE
;  192          0486  1 !
;  193          0487  1 ! IMPLICIT OUTPUTS:
;  194          0488  1 !
;  195          0489  1 !       NONE
;  196          0490  1 !
;  197          0491  1 ! COMPLETION STATUS:
;  198          0492  1 !
;  199          0493  1 !       NONE
;  200          0494  1 !
;  201          0495  1 ! SIDE EFFECTS:
;  202          0496  1 !
;  203          0497  1 !       May signal decimal overflow if an error occurs in the OTS
;  204          0498  1 !       conversion routine
;  205          0499  1 !
;  206          0500  1 !--
;  207          0501  1
;  208          0502  2     BEGIN
;  209          0503  2
;  210          0504  2     LOCAL
;  211          0505  2         FMP : REF BLOCK [0,BYTE] FIELD (BSF$FCD),       ! Ptr to BASIC frame
;  212          0506  2         STATUS;
;  213          0507  2
;  214          0508  2     STATUS = OTS$CVTFP_R9 (.SCALE, .SRC, .DESTLEN, .DEST);
;  215          0509  3     IF (NOT .STATUS)
;  216          0510  2     THEN
;  217          0511  3         BEGIN
;  218          0512  3
```

```
;  219      0513 3          FIND_FRAME (FMP);                                   ! Find BASIC frame
;  220      0514 3          IF (.FMP NEQ 0) AND (.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_DV) NEQ 0
;  221      0515 3          THEN                                                ! If "/OVERFLOW = NODEC" not set
;  222      0516 3              BAS$$SIGNAL (BAS$K_DECERR);
;  223      0517 3
;  224      0518 2          END;
;  225      0519 1      END;                                        ! routine BAS$CVTFP


                                                        .TITLE  BAS$CVTRP BAS$CVTRP - Convert real to packed
                                                        .IDENT  \1-004\

                                                        .EXTRN  BAS$HANDLER, OTS$CVTFP_R9
                                                        .EXTRN  OTS$CVTDP_R9, OTS$CVTGP_R9
                                                        .EXTRN  OTS$CVTHP_R9, OTS$CVTRFP_R9
                                                        .EXTRN  OTS$CVTRDP_R9, OTS$CVTRGP_R9
                                                        .EXTRN  OTS$CVTRHP_R9, OTS$CVTPF_R9
                                                        .EXTRN  OTS$CVTPD_R9, OTS$CVTPG_R9
                                                        .EXTRN  OTS$CVTPH_R9, BAS$$SIGNAL
                                                        .EXTRN  BAS$K_DECERR

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

                                  0FFC 00000            .ENTRY  BAS$CVTFP, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 0457
                                                                R10,R11
                 59      04  AC   D0 00002              MOVL    DEST, R9                                    ; 0508
                 58      08  AC   D0 00006              MOVL    DESTLEN, R8                                 ;
                 57      0C  AC   D0 0000A              MOVL    SRC, R7                                     ;
                 56      10  AC   D0 0000E              MOVL    SCALE, R6                                   ;
              00000000G      00   16 00012              JSB     OTS$CVTFP_R9                                ;
                 2B          50   E8 00018              BLBS    STATUS, 3$                                  ; 0509
                 50          5D   D0 0001B              MOVL    FP, FMP                                     ; 0513
                 50      0C  A0   D0 0001E 1$:          MOVL    12(FMP), FMP                                ;
                 51 00000000G 00  9E 00022              MOVAB   BAS$HANDLER, R1                             ;
                 51          60   D1 00029              CMPL    (FMP), R1                                   ;
                             04   13 0002C              BEQL    2$                                          ;
                             50   D5 0002E              TSTL    FMP                                         ;
                             EC   12 00030              BNEQ    1$                                          ;
                             50   D5 00032 2$:          TSTL    FMP                                         ; 0514
                             10   13 00034              BEQL    3$                                          ;
          0B        E6   A0  0A   E1 00036              BBC     #10, -26(FMP), 3$                           ;
                 7E     00G  8F   9A 0003B              MOVZBL  #BAS$K_DECERR, -(SP)                        ; 0516
           00000000G  00     01   FB 0003F              CALLS   #1, BAS$$SIGNAL                             ;
                                  04 00046 3$:          RET                                                 ; 0519

; Routine Size:  71 bytes,    Routine Base:  _BAS$CODE + 0000
```

```
;   227          0520  1 %SBTTL 'BAS$CVTDP - Convert double to packed'
;   228          0521  1 GLOBAL ROUTINE BAS$CVTDP (                                  ! Convert double to packed
;   229          0522  1         DEST,                                               ! place to store conv. number
;   230          0523  1         DESTLEN,                                            ! number of digits in dest
;   231          0524  1         SRC,                                                ! number to be converted
;   232          0525  1         SCALE                                               ! power of ten to mult src
;   233          0526  1     ) : NOVALUE =
;   234          0527  1
;   235          0528  1 !++
;   236          0529  1 ! FUNCTIONAL DESCRIPTION:
;   237          0530  1 !
;   238          0531  1 !       Converts a double floating number to packed.
;   239          0532  1 !
;   240          0533  1 ! CALLING SEQUENCE:
;   241          0534  1 !
;   242          0535  1 !       BAS$CVTDP (DEST.wp.r, DESTLEN.rl.v, SRC.rd.r, SCALE.rl.v)
;   243          0536  1 !
;   244          0537  1 ! FORMAL PARAMETERS:
;   245          0538  1 !
;   246          0539  1 !       DEST.wp.r       place to store the converted number
;   247          0540  1 !       DESTLEN.rl.v    number of digits in the destination
;   248          0541  1 !       SRC.rd.r        number to be converted
;   249          0542  1 !       SCALE.rl.v      power of ten by which the internal
;   250          0543  1 !                       representation of the sourc must be
;   251          0544  1 !                       multiplied to scale the same as the
;   252          0545  1 !                       internal representation of the dest.
;   253          0546  1 !
;   254          0547  1 ! IMPLICIT INPUTS:
;   255          0548  1 !
;   256          0549  1 !       NONE
;   257          0550  1 !
;   258          0551  1 ! IMPLICIT OUTPUTS:
;   259          0552  1 !
;   260          0553  1 !       NONE
;   261          0554  1 !
;   262          0555  1 ! COMPLETION STATUS:
;   263          0556  1 !
;   264          0557  1 !       NONE
;   265          0558  1 !
;   266          0559  1 ! SIDE EFFECTS:
;   267          0560  1 !
;   268          0561  1 !       May signal decimal overflow if overflow occurs in the OTS
;   269          0562  1 !       conversion routine
;   270          0563  1 !
;   271          0564  1 !--
;   272          0565  1
;   273          0566  2     BEGIN
;   274          0567  2
;   275          0568  2     LOCAL
;   276          0569  2         FMP : REF BLOCK [0,BYTE] FIELD (BSF$FCD),       ! Ptr to BASIC frame
;   277          0570  2         STATUS;
;   278          0571  2
;   279          0572  2     STATUS = OTS$CVTDP_R9 (.SCALE, .SRC, .DESTLEN, .DEST);
;   280          0573  3     IF (NOT .STATUS)
;   281          0574  2     THEN
;   282          0575  3         BEGIN
;   283          0576  3
```

```
;  284         0577  3              FIND_FRAME (FMP);                                  ! Find BASIC frame
;  285         0578  3              IF (.FMP NEQ 0) AND (.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_DV) NEQ 0
;  286         0579  3              THEN                                               ! If "/OVERFLOW = NODEC" not set
;  287         0580  3                  BAS$$SIGNAL (BAS$K_DECERR);
;  288         0581  3
;  289         0582  2              END;
;  290         0583  2
;  291         0584  1          END;                                           ! End of routine BAS$CVTDP


                                        0FFC 00000            .ENTRY  BAS$CVTDP, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 0521
                                                                      R10,R11
                   59         04   AC   D0 00002            MOVL    DEST, R9                                    ; 0572
                   58         08   AC   D0 00006            MOVL    DESTLEN, R8                                 ;
                   57         0C   AC   D0 0000A            MOVL    SRC, R7                                     ;
                   56         10   AC   D0 0000E            MOVL    SCALE, R6                                   ;
                     00000000G     00   16 00012            JSB     OTS$CVTDP_R9                                ;
                   2B              50   E8 00018            BLBS    STATUS, 3$                                  ; 0573
                   50              5D   D0 0001B            MOVL    FP, FMP                                     ; 0577
                   50         0C   A0   D0 0001E 1$:        MOVL    12(FMP), FMP                                ;
                   51 00000000G    00   9E 00022            MOVAB   BAS$HANDLER, R1                             ;
                   51              60   D1 00029            CMPL    (FMP), R1                                   ;
                   04              13   0002C               BEQL    2$                                          ;
                   50              D5   0002E               TSTL    FMP                                         ;
                   EC              12   00030               BNEQ    1$                                          ;
                   50              D5   00032 2$:           TSTL    FMP                                         ; 0578
                   10              13   00034               BEQL    3$                                          ;
      0B      E6   A0              0A   E1 00036            BBC     #10, -26(FMP), 3$                           ;
                   7E        00G   8F   9A 0003B            MOVZBL  #BAS$K_DECERR, -(SP)                        ; 0580
         00000000G 00              01   FB 0003F            CALLS   #1, BAS$$SIGNAL                             ;
                                        04 00046 3$:        RET                                                 ; 0584

; Routine Size:  71 bytes,    Routine Base:  _BAS$CODE + 0047
```

```
;   293          0585  1 %SBTTL 'BAS$CVTGP - Convert gfloat to packed'
;   294          0586  1 GLOBAL ROUTINE BAS$CVTGP (                                  ! Convert gfloat to packed
;   295          0587  1         DEST,                                 ! place to store conv. number
;   296          0588  1         DESTLEN,                              ! number of digits in dest
;   297          0589  1         SRC,                                  ! number to be converted
;   298          0590  1         SCALE                                 ! power of ten to mult src
;   299          0591  1     ) : NOVALUE =
;   300          0592  1
;   301          0593  1 !++
;   302          0594  1 ! FUNCTIONAL DESCRIPTION:
;   303          0595  1 !
;   304          0596  1 !       Converts a g floating number to packed.
;   305          0597  1 !
;   306          0598  1 ! CALLING SEQUENCE:
;   307          0599  1 !
;   308          0600  1 !       BAS$CVTGP (DEST.wp.r, DESTLEN.rl.v, SRC.rg.r, SCALE.rl.v)
;   309          0601  1 !
;   310          0602  1 ! FORMAL PARAMETERS:
;   311          0603  1 !
;   312          0604  1 !       DEST.wp.r       place to store the converted number
;   313          0605  1 !       DESTLEN.rl.v    number of digits in the destination
;   314          0606  1 !       SRC.rg.r        number to be converted
;   315          0607  1 !       SCALE.rl.v      power of ten by which the internal
;   316          0608  1 !                       representation of the sourc must be
;   317          0609  1 !                       multiplied to scale the same as the
;   318          0610  1 !                       internal representation of the dest.
;   319          0611  1 !
;   320          0612  1 ! IMPLICIT INPUTS:
;   321          0613  1 !
;   322          0614  1 !       NONE
;   323          0615  1 !
;   324          0616  1 ! IMPLICIT OUTPUTS:
;   325          0617  1 !
;   326          0618  1 !       NONE
;   327          0619  1 !
;   328          0620  1 ! COMPLETION STATUS:
;   329          0621  1 !
;   330          0622  1 !       NONE
;   331          0623  1 !
;   332          0624  1 ! SIDE EFFECTS:
;   333          0625  1 !
;   334          0626  1 !       May signal decimal overflow if that error occurs in the OTS
;   335          0627  1 !       conversion routine
;   336          0628  1 !
;   337          0629  1 !--
;   338          0630  1
;   339          0631  2     BEGIN
;   340          0632  2
;   341          0633  2     LOCAL
;   342          0634  2         FMP : REF BLOCK [0,BYTE] FIELD (BSF$FCD),       ! Ptr to BASIC frame
;   343          0635  2         STATUS;
;   344          0636  2
;   345          0637  2     STATUS = OTS$CVTGP_R9 (.SCALE, .SRC, .DESTLEN, .DEST);
;   346          0638  3     IF (NOT .STATUS)
;   347          0639  2     THEN
;   348          0640  3         BEGIN
;   349          0641  3
```

```
;  350          0642   3           FIND_FRAME (FMP);                            ! Find BASIC frame
;  351          0643   3           IF (.FMP NEQ 0) AND (.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_DV) NEQ 0
;  352          0644   3           THEN                                         ! If "/OVERFLOW = NODEC'' not set
;  353          0645   3               BAS$$SIGNAL (BAS$K_DECERR);
;  354          0646   3
;  355          0647   2           END;
;  356          0648   1       END;                                        ! End of routine BAS$CVTGP


                                       0FFC 00000        .ENTRY  BAS$CVTGP, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0586
                                                                 R10,R11
                   59        04   AC   D0 00002          MOVL    DEST, R9                                   ; 0637
                   58        08   AC   D0 00006          MOVL    DESTLEN, R8                                ;
                   57        0C   AC   D0 0000A          MOVL    SRC, R7                                    ;
                   56        10   AC   D0 0000E          MOVL    SCALE, R6                                  ;
                      00000000G   00   16 00012          JSB     OTS$CVTGP_R9                               ;
                   2B             50   E8 00018          BLBS    STATUS, 3$                                 ; 0638
                   50             5D   D0 0001B          MOVL    FP, FMP                                    ; 0642
                   50        0C   A0   D0 0001E  1$:     MOVL    12(FMP), FMP                               ;
                   51 00000000G   00   9E 00022          MOVAB   BAS$HANDLER, R1                            ;
                   51             60   D1 00029          CMPL    (FMP), R1                                  ;
                                  04   13 0002C          BEQL    2$                                         ;
                                  50   D5 0002E          TSTL    FMP                                        ;
                                  EC   12 00030          BNEQ    1$                                         ;
                                  50   D5 00032  2$:     TSTL    FMP                                        ; 0643
                                  10   13 00034          BEQL    3$                                         ;
        0B        E6   A0         0A   E1 00036          BBC     #10, -26(FMP), 3$                          ;
                   7E       00G   8F   9A 0003B          MOVZBL  #BAS$K_DECERR, -(SP)                       ; 0645
           00000000G   00         01   FB 0003F          CALLS   #1, BAS$$SIGNAL                            ;
                                       04 00046  3$:     RET                                                ; 0648

; Routine Size:  71 bytes,    Routine Base:  _BAS$CODE + 008E
```

```
;  358          0649  1 %SBTTL 'BAS$CVTHP - Convert hfloat to packed'
;  359          0650  1 GLOBAL ROUTINE BAS$CVTHP (                                  ! Convert hfloat to packed
;  360          0651  1         DEST,                                               ! place to store conv. number
;  361          0652  1         DESTLEN,                                            ! number of digits in dest
;  362          0653  1         SRC,                                                ! number to be converted
;  363          0654  1         SCALE                                               ! power of ten to mult src
;  364          0655  1     ) : NOVALUE =
;  365          0656  1
;  366          0657  1 !++
;  367          0658  1 ! FUNCTIONAL DESCRIPTION:
;  368          0659  1 !
;  369          0660  1 !       Converts a h floating number to packed.
;  370          0661  1 !
;  371          0662  1 ! CALLING SEQUENCE:
;  372          0663  1 !
;  373          0664  1 !       BAS$CVTHP (DEST.wp.r, DESTLEN.rl.v, SRC.rh.r, SCALE.rl.v)
;  374          0665  1 !
;  375          0666  1 ! FORMAL PARAMETERS:
;  376          0667  1 !
;  377          0668  1 !       DEST.wp.r       place to store the converted number
;  378          0669  1 !       DESTLEN.rl.v    number of digits in the destination
;  379          0670  1 !       SRC.rh.r        number to be converted
;  380          0671  1 !       SCALE.rl.v      power of ten by which the internal
;  381          0672  1 !                       representation of the sourc must be
;  382          0673  1 !                       multiplied to scale the same as the
;  383          0674  1 !                       internal representation of the dest.
;  384          0675  1 !
;  385          0676  1 ! IMPLICIT INPUTS:
;  386          0677  1 !
;  387          0678  1 !       NONE
;  388          0679  1 !
;  389          0680  1 ! IMPLICIT OUTPUTS:
;  390          0681  1 !
;  391          0682  1 !       NONE
;  392          0683  1 !
;  393          0684  1 ! COMPLETION STATUS:
;  394          0685  1 !
;  395          0686  1 !       May signal decimal overflow if that error occurs in the OTS
;  396          0687  1 !       conversion routine
;  397          0688  1 !
;  398          0689  1 ! SIDE EFFECTS:
;  399          0690  1 !
;  400          0691  1 !       NONE
;  401          0692  1 !
;  402          0693  1 !--
;  403          0694  1
;  404          0695  2     BEGIN
;  405          0696  2
;  406          0697  2     LOCAL
;  407          0698  2         FMP : REF BLOCK [0,BYTE] FIELD (BSF$FCD),       ! Ptr to BASIC frame
;  408          0699  2         STATUS;
;  409          0700  2
;  410          0701  2     STATUS = OTS$CVTHP_R9 (.SCALE, .SRC, .DESTLEN, .DEST);
;  411          0702  3     IF (NOT .STATUS)
;  412          0703  2     THEN
;  413          0704  3         BEGIN
;  414          0705  3
```

```
;  415     0706  3         FIND_FRAME (FMP);                                   ! Find BASIC frame
;  416     0707  3         IF (.FMP NEQ 0) AND (.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_DV) NEQ 0
;  417     0708  3         THEN                                                ! If "/OVERFLOW = NODEC" not set
;  418     0709  3             BAS$$SIGNAL (BAS$K_DECERR);
;  419     0710  3
;  420     0711  2         END;
;  421     0712  1     END;                                            ! End of routine BAS$CVTHP


                                   0FFC 00000          .ENTRY  BAS$CVTHP, Save R2,R3,R4,R5,R6,R7,R8,R9,-      ; 0650
                                                               R10,R11                                        ;
                59         04   AC   D0 00002          MOVL    DEST, R9                                       ; 0701
                58         08   AC   D0 00006          MOVL    DESTLEN, R8                                    ;
                57         0C   AC   D0 0000A          MOVL    SRC, R7                                        ;
                56         10   AC   D0 0000E          MOVL    SCALE, R6                                      ;
                    00000000G   00   16 00012          JSB     OTS$CVTHP_R9                                   ;
                2B              50   E8 00018          BLBS    STATUS, 3$                                     ; 0702
                50              5D   D0 0001B          MOVL    FP, FMP                                        ; 0706
                50         0C   A0   D0 0001E  1$:     MOVL    12(FMP), FMP                                   ;
                51  00000000G   00   9E 00022          MOVAB   BAS$HANDLER, R1                                ;
                51              60   D1 00029          CMPL    (FMP), R1                                      ;
                                04   13 0002C          BEQL    2$                                             ;
                                50   D5 0002E          TSTL    FMP                                            ;
                                EC   12 00030          BNEQ    1$                                             ;
                                50   D5 00032  2$:     TSTL    FMP                                            ; 0707
                                10   13 00034          BEQL    3$                                             ;
      0B          E6    A0           0A   E1 00036     BBC     #10, -26(FMP), 3$                              ;
                7E        00G   8F   9A 0003B          MOVZBL  #BAS$K_DECERR, -(SP)                           ; 0709
       00000000G  00              01   FB 0003F        CALLS   #1, BAS$$SIGNAL                                ;
                                     04 00046  3$:     RET                                                    ; 0712

; Routine Size:  71 bytes,    Routine Base:  _BAS$CODE + 00D5
```

```
;  423     0713  1 %SBTTL 'BAS$CVTRFP - Convert float to packed (rounded)'
;  424     0714  1 GLOBAL ROUTINE BAS$CVTRFP (                          ! Convert float to packed (rounded)
;  425     0715  1         DEST,                                       ! place to store conv. number
;  426     0716  1         DESTLEN,                                    ! number of digits in dest
;  427     0717  1         SRC,                                        ! number to be converted
;  428     0718  1         SCALE                                       ! power of ten to mult src
;  429     0719  1     ) : NOVALUE =
;  430     0720  1
;  431     0721  1 !++
;  432     0722  1 ! FUNCTIONAL DESCRIPTION:
;  433     0723  1 !
;  434     0724  1 !       Converts a single floating number to packed using rounding.
;  435     0725  1 !
;  436     0726  1 ! CALLING SEQUENCE:
;  437     0727  1 !
;  438     0728  1 !       BAS$CVTRFP (DEST.wp.r, DESTLEN.rl.v, SRC.rf.r, SCALE.rl.v)
;  439     0729  1 !
;  440     0730  1 ! FORMAL PARAMETERS:
;  441     0731  1 !
;  442     0732  1 !       DEST.wp.r       place to store the converted number
;  443     0733  1 !       DESTLEN.rl.v    number of digits in the destination
;  444     0734  1 !       SRC.rf.r        number to be converted
;  445     0735  1 !       SCALE.rl.v      power of ten by which the internal
;  446     0736  1 !                       representation of the sourc must be
;  447     0737  1 !                       multiplied to scale the same as the
;  448     0738  1 !                       internal representation of the dest.
;  449     0739  1 !
;  450     0740  1 ! IMPLICIT INPUTS:
;  451     0741  1 !
;  452     0742  1 !       NONE
;  453     0743  1 !
;  454     0744  1 ! IMPLICIT OUTPUTS:
;  455     0745  1 !
;  456     0746  1 !       NONE
;  457     0747  1 !
;  458     0748  1 ! COMPLETION STATUS:
;  459     0749  1 !
;  460     0750  1 !       NONE
;  461     0751  1 !
;  462     0752  1 ! SIDE EFFECTS:
;  463     0753  1 !
;  464     0754  1 !       May signal decimal overflow if an error occurs in the OTS
;  465     0755  1 !       conversion routine
;  466     0756  1 !
;  467     0757  1 !--
;  468     0758  1
;  469     0759  2     BEGIN
;  470     0760  2
;  471     0761  2     LOCAL
;  472     0762  2         FMP : REF BLOCK [0,BYTE] FIELD (BSF$FCD),       ! Ptr to BASIC frame
;  473     0763  2         STATUS;
;  474     0764  2
;  475     0765  2     STATUS = OTS$CVTRFP_R9 (.SCALE, .SRC, .DESTLEN, .DEST);
;  476     0766  3     IF (NOT .STATUS)
;  477     0767  2     THEN
;  478     0768  3         BEGIN
;  479     0769  3
```

```
0770 3          FIND_FRAME (FMP);                                          ! Find BASIC frame
0771 3          IF (.FMP NEQ 0) AND (.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_DV) NEQ 0
0772 3          THEN                                                       ! If "/OVERFLOW = NODEC" not set
0773 3              BAS$$SIGNAL (BAS$K_DECERR);
0774 3
0775 2          END;
0776 1      END;                                               ! routine BAS$CVTRFP



                                OFFC 00000           .ENTRY  BAS$CVTRFP, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 0714
                                                             R10,R11                                      ;
               59      04    AC   D0 00002           MOVL    DEST, R9                                     ; 0765
               58      08    AC   D0 00006           MOVL    DESTLEN, R8                                  ;
               57      0C    AC   D0 0000A           MOVL    SRC, R7                                      ;
               56      10    AC   D0 0000E           MOVL    SCALE, R6                                    ;
            00000000G        00   16 00012           JSB     OTS$CVTRFP_R9                                ;
               2B            50   E8 00018           BLBS    STATUS, 3$                                   ; 0766
               50            5D   D0 0001B           MOVL    FP, FMP                                      ; 0770
               50      0C    A0   D0 0001E 1$:       MOVL    12(FMP), FMP                                 ;
               51 00000000G  00   9E 00022           MOVAB   BAS$HANDLER, R1                              ;
               51            60   D1 00029           CMPL    (FMP), R1                                    ;
                             04   13 0002C           BEQL    2$                                           ;
                             50   D5 0002E           TSTL    FMP                                          ;
                             EC   12 00030           BNEQ    1$                                           ;
                             50   D5 00032 2$:       TSTL    FMP                                          ; 0771
                             10   13 00034           BEQL    3$                                           ;
   0B        E6   A0         0A   E1 00036           BBC     #10, -26(FMP), 3$                            ;
               7E      00G   8F   9A 0003B           MOVZBL  #BAS$K_DECERR, -(SP)                         ; 0773
    00000000G  00            01   FB 0003F           CALLS   #1, BAS$$SIGNAL                              ;
                                  04 00046 3$:       RET                                                  ; 0776

; Routine Size:  71 bytes,    Routine Base:  _BAS$CODE + 011C
```

```
;  488          0777  1 %SBTTL 'BAS$CVTRDP - Convert double to packed (rounded)'
;  489          0778  1 GLOBAL ROUTINE BAS$CVTRDP (                          ! Convert double to packed (rounded)
;  490          0779  1         DEST,                                       ! place to store conv. number
;  491          0780  1         DESTLEN,                                    ! number of digits in dest
;  492          0781  1         SRC,                                        ! number to be converted
;  493          0782  1         SCALE                                       ! power of ten to mult src
;  494          0783  1     ) : NOVALUE =
;  495          0784  1
;  496          0785  1 !++
;  497          0786  1 ! FUNCTIONAL DESCRIPTION:
;  498          0787  1 !
;  499          0788  1 !       Converts a double floating number to packed using rounding.
;  500          0789  1 !
;  501          0790  1 ! CALLING SEQUENCE:
;  502          0791  1 !
;  503          0792  1 !       BAS$CVTRDP (DEST.wp.r, DESTLEN.rl.v, SRC.rd.r, SCALE.rl.v)
;  504          0793  1 !
;  505          0794  1 ! FORMAL PARAMETERS:
;  506          0795  1 !
;  507          0796  1 !       DEST.wp.r       place to store the converted number
;  508          0797  1 !       DESTLEN.rl.v    number of digits in the destination
;  509          0798  1 !       SRC.rd.r        number to be converted
;  510          0799  1 !       SCALE.rl.v      power of ten by which the internal
;  511          0800  1 !                       representation of the sourc must be
;  512          0801  1 !                       multiplied to scale the same as the
;  513          0802  1 !                       internal representation of the dest.
;  514          0803  1 !
;  515          0804  1 ! IMPLICIT INPUTS:
;  516          0805  1 !
;  517          0806  1 !       NONE
;  518          0807  1 !
;  519          0808  1 ! IMPLICIT OUTPUTS:
;  520          0809  1 !
;  521          0810  1 !       NONE
;  522          0811  1 !
;  523          0812  1 ! COMPLETION STATUS:
;  524          0813  1 !
;  525          0814  1 !       NONE
;  526          0815  1 !
;  527          0816  1 ! SIDE EFFECTS:
;  528          0817  1 !
;  529          0818  1 !       May signal decimal overflow if overflow occurs in the OTS
;  530          0819  1 !       conversion routine
;  531          0820  1 !
;  532          0821  1 !--
;  533          0822  1
;  534          0823  2     BEGIN
;  535          0824  2
;  536          0825  2     LOCAL
;  537          0826  2         FMP : REF BLOCK [0,BYTE] FIELD (BSF$FCD),       ! Ptr to BASIC frame
;  538          0827  2         STATUS;
;  539          0828  2
;  540          0829  2     STATUS = OTS$CVTRDP_R9 (.SCALE, .SRC, .DESTLEN, .DEST);
;  541          0830  3     IF (NOT .STATUS)
;  542          0831  2     THEN
;  543          0832  3         BEGIN
;  544          0833  3
```

```
;        0834  3          FIND_FRAME (FMP);                                     ! Find BASIC frame
;        0835  3          IF (.FMP NEQ 0) AND (.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_DV) NEQ 0
;        0836  3          THEN                                                  ! If "/OVERFLOW = NODEC" not set
;        0837  3              BAS$$SIGNAL (BAS$K_DECERR);
;        0838  3
;        0839  2          END;
;        0840  1      END;                                              ! End of routine BAS$CVTRDP



                                              0FFC 00000          .ENTRY  BAS$CVTRDP, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 0778
                                                                          R10,R11                                      ;
                     59        04   AC   D0 00002          MOVL    DEST, R9                                     ; 0829
                     58        08   AC   D0 00006          MOVL    DESTLEN, R8                                  ;
                     57        0C   AC   D0 0000A          MOVL    SRC, R7                                      ;
                     56        10   AC   D0 0000E          MOVL    SCALE, R6                                    ;
                      00000000G   00   16 00012          JSB     OTS$CVTRDP_R9                                ;
                     2B             50   E8 00018          BLBS    STATUS, 3$                                   ; 0830
                     50             5D   D0 0001B          MOVL    FP, FMP                                      ; 0834
                     50        0C   A0   D0 0001E 1$:      MOVL    12(FMP), FMP                                 ;
                     51 00000000G   00   9E 00022          MOVAB   BAS$HANDLER, R1                              ;
                     51             60   D1 00029          CMPL    (FMP), R1                                    ;
                                    04   13 0002C          BEQL    2$                                           ;
                                    50   D5 0002E          TSTL    FMP                                          ;
                                    EC   12 00030          BNEQ    1$                                           ;
                                    50   D5 00032 2$:      TSTL    FMP                                          ; 0835
                                    10   13 00034          BEQL    3$                                           ;
           0B         E6   A0       0A   E1 00036          BBC     #10, -26(FMP), 3$                            ;
                          7E       00G   8F   9A 0003B          MOVZBL  #BAS$K_DECERR, -(SP)                         ; 0837
                 00000000G   00       01   FB 0003F          CALLS   #1, BAS$$SIGNAL                              ;
                                          04 00046 3$:      RET                                                  ; 0840

; Routine Size:  71 bytes,    Routine Base:  _BAS$CODE + 0163
```

```
;  553          0841  1 %SBTTL 'BAS$CVTRGP - Convert gfloat to packed (rounded)'
;  554          0842  1 GLOBAL ROUTINE BAS$CVTRGP (                          ! Convert gfloat to packed (rounded)
;  555          0843  1           DEST,                                      ! place to store conv. number
;  556          0844  1           DESTLEN,                                   ! number of digits in dest
;  557          0845  1           SRC,                                       ! number to be converted
;  558          0846  1           SCALE                                      ! power of ten to mult src
;  559          0847  1       ) : NOVALUE =
;  560          0848  1
;  561          0849  1 !++
;  562          0850  1 ! FUNCTIONAL DESCRIPTION:
;  563          0851  1 !
;  564          0852  1 !       Converts a g floating number to packed using rounding.
;  565          0853  1 !
;  566          0854  1 ! CALLING SEQUENCE:
;  567          0855  1 !
;  568          0856  1 !       BAS$CVTRGP (DEST.wp.r, DESTLEN.rl.v, SRC.rg.r, SCALE.rl.v)
;  569          0857  1 !
;  570          0858  1 ! FORMAL PARAMETERS:
;  571          0859  1 !
;  572          0860  1 !       DEST.wp.r       place to store the converted number
;  573          0861  1 !       DESTLEN.rl.v    number of digits in the destination
;  574          0862  1 !       SRC.rg.r        number to be converted
;  575          0863  1 !       SCALE.rl.v      power of ten by which the internal
;  576          0864  1 !                       representation of the sourc must be
;  577          0865  1 !                       multiplied to scale the same as the
;  578          0866  1 !                       internal representation of the dest.
;  579          0867  1 !
;  580          0868  1 ! IMPLICIT INPUTS:
;  581          0869  1 !
;  582          0870  1 !       NONE
;  583          0871  1 !
;  584          0872  1 ! IMPLICIT OUTPUTS:
;  585          0873  1 !
;  586          0874  1 !       NONE
;  587          0875  1 !
;  588          0876  1 ! COMPLETION STATUS:
;  589          0877  1 !
;  590          0878  1 !       NONE
;  591          0879  1 !
;  592          0880  1 ! SIDE EFFECTS:
;  593          0881  1 !
;  594          0882  1 !       May signal decimal overflow if that error occurs in the OTS
;  595          0883  1 !       conversion routine
;  596          0884  1 !
;  597          0885  1 !--
;  598          0886  1
;  599          0887  2     BEGIN
;  600          0888  2
;  601          0889  2     LOCAL
;  602          0890  2         FMP : REF BLOCK [0,BYTE] FIELD (BSF$FCD),       ! Ptr to BASIC frame
;  603          0891  2         STATUS;
;  604          0892  2
;  605          0893  2     STATUS = OTS$CVTRGP_R9 (.SCALE, .SRC, .DESTLEN, .DEST);
;  606          0894  3     IF (NOT .STATUS)
;  607          0895  2     THEN
;  608          0896  3         BEGIN
;  609          0897  3
```

```
;  610       0898  3        FIND_FRAME (FMP);                                  ! Find BASIC frame
;  611       0899  3        IF (.FMP NEQ 0) AND (.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_DV) NEQ 0
;  612       0900  3        THEN                                               ! If "/OVERFLOW = NODEC" not set
;  613       0901  3            BAS$$SIGNAL (BAS$K_DECERR);
;  614       0902  3
;  615       0903  2        END;
;  616       0904  1    END;                                       ! End of routine BAS$CVTRGP


                                            0FFC 00000          .ENTRY  BAS$CVTRGP, Save R2,R3,R4,R5,R6,R7,R8,R9,-    ; 0842
                                                                        R10,R11
                     59        04    AC  D0 00002          MOVL    DEST, R9                                      ; 0893
                     58        08    AC  D0 00006          MOVL    DESTLEN, R8                                   ;
                     57        0C    AC  D0 0000A          MOVL    SRC, R7                                       ;
                     56        10    AC  D0 0000E          MOVL    SCALE, R6                                     ;
                       00000000G     00  16 00012          JSB     OTS$CVTRGP_R9                                 ;
                     2B              50  E8 00018          BLBS    STATUS, 3$                                    ; 0894
                     50              5D  D0 0001B          MOVL    FP, FMP                                       ; 0898
                     50        0C    A0  D0 0001E  1$:     MOVL    12(FMP), FMP                                  ;
                     51 00000000G    00  9E 00022          MOVAB   BAS$HANDLER, R1                               ;
                     51              60  D1 00029          CMPL    (FMP), R1                                     ;
                                     04  13 0002C          BEQL    2$                                            ;
                                     50  D5 0002E          TSTL    FMP                                           ;
                                     EC  12 00030          BNEQ    1$                                            ;
                                     50  D5 00032  2$:     TSTL    FMP                                           ; 0899
                                     10  13 00034          BEQL    3$                                            ;
            0B        E6    A0       0A  E1 00036          BBC     #10, -26(FMP), 3$                             ;
                     7E       00G    8F  9A 0003B          MOVZBL  #BAS$K_DECERR, -(SP)                          ; 0901
         00000000G   00              01  FB 0003F          CALLS   #1, BAS$$SIGNAL                               ;
                                         04 00046  3$:     RET                                                   ; 0904

; Routine Size:  71 bytes,    Routine Base:  _BAS$CODE + 01AA
```

```
;  618     0905  1 %SBTTL 'BAS$CVTRHP - Convert hfloat to packed (rounded)'
;  619     0906  1 GLOBAL ROUTINE BAS$CVTRHP (                          ! Convert hfloat to packed (rounded)
;  620     0907  1         DEST,                                       ! place to store conv. number
;  621     0908  1         DESTLEN,                                    ! number of digits in dest
;  622     0909  1         SRC,                                        ! number to be converted
;  623     0910  1         SCALE                                       ! power of ten to mult src
;  624     0911  1     ) : NOVALUE =
;  625     0912  1
;  626     0913  1 !++
;  627     0914  1 ! FUNCTIONAL DESCRIPTION:
;  628     0915  1 !
;  629     0916  1 !       Converts a h floating number to packed using rounding.
;  630     0917  1 !
;  631     0918  1 ! CALLING SEQUENCE:
;  632     0919  1 !
;  633     0920  1 !       BAS$CVTRHP (DEST.wp.r, DESTLEN.rl.v, SRC.rh.r, SCALE.rl.v)
;  634     0921  1 !
;  635     0922  1 ! FORMAL PARAMETERS:
;  636     0923  1 !
;  637     0924  1 !       DEST.wp.r       place to store the converted number
;  638     0925  1 !       DESTLEN.rl.v    number of digits in the destination
;  639     0926  1 !       SRC.rh.r        number to be converted
;  640     0927  1 !       SCALE.rl.v      power of ten by which the internal
;  641     0928  1 !                       representation of the sourc must be
;  642     0929  1 !                       multiplied to scale the same as the
;  643     0930  1 !                       internal representation of the dest.
;  644     0931  1 !
;  645     0932  1 ! IMPLICIT INPUTS:
;  646     0933  1 !
;  647     0934  1 !       NONE
;  648     0935  1 !
;  649     0936  1 ! IMPLICIT OUTPUTS:
;  650     0937  1 !
;  651     0938  1 !       NONE
;  652     0939  1 !
;  653     0940  1 ! COMPLETION STATUS:
;  654     0941  1 !
;  655     0942  1 !       May signal decimal overflow if that error occurs in the OTS
;  656     0943  1 !       conversion routine
;  657     0944  1 !
;  658     0945  1 ! SIDE EFFECTS:
;  659     0946  1 !
;  660     0947  1 !       NONE
;  661     0948  1 !
;  662     0949  1 !--
;  663     0950  1
;  664     0951  2     BEGIN
;  665     0952  2
;  666     0953  2     LOCAL
;  667     0954  2         FMP : REF BLOCK [0,BYTE] FIELD (BSF$FCD),       ! Ptr to BASIC frame
;  668     0955  2         STATUS;
;  669     0956  2
;  670     0957  2     STATUS = OTS$CVTRHP_R9 (.SCALE, .SRC, .DESTLEN, .DEST);
;  671     0958  3     IF (NOT .STATUS)
;  672     0959  2     THEN
;  673     0960  3         BEGIN
;  674     0961  3
```

```
;  675      0962 3          FIND_FRAME (FMP);                                ! Find BASIC frame
;  676      0963 3          IF (.FMP NEQ 0) AND (.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_DV) NEQ 0
;  677      0964 3          THEN                                             ! If "/OVERFLOW = NODEC" not set
;  678      0965 3              BAS$$SIGNAL (BAS$K_DECERR);
;  679      0966 3
;  680      0967 2          END;
;  681      0968 1      END;                                         ! End of routine BAS$CVTRHP



                                        0FFC 00000          .ENTRY  BAS$CVTRHP, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0906
                                                                    R10,R11                                    ;
                     59       04   AC   D0 00002          MOVL    DEST, R9                                   ; 0957
                     58       08   AC   D0 00006          MOVL    DESTLEN, R8                                ;
                     57       0C   AC   D0 0000A          MOVL    SRC, R7                                    ;
                     56       10   AC   D0 0000E          MOVL    SCALE, R6                                  ;
                        00000000G  00   16 00012          JSB     OTS$CVTRHP_R9                              ;
                     2B            50   E8 00018          BLBS    STATUS, 3$                                 ; 0958
                     50            5D   D0 0001B          MOVL    FP, FMP                                    ; 0962
                     50       0C   A0   D0 0001E 1$:      MOVL    12(FMP), FMP                               ;
                     51 00000000G  00   9E 00022          MOVAB   BAS$HANDLER, R1                            ;
                     51            60   D1 00029          CMPL    (FMP), R1                                  ;
                                   04   13 0002C          BEQL    2$                                         ;
                                   50   D5 0002E          TSTL    FMP                                        ;
                                   EC   12 00030          BNEQ    1$                                         ;
                                   50   D5 00032 2$:      TSTL    FMP                                        ; 0963
                                   10   13 00034          BEQL    3$                                         ;
           0B         E6   A0      0A   E1 00036          BBC     #10, -26(FMP), 3$                          ;
                     7E      00G   8F   9A 0003B          MOVZBL  #BAS$K_DECERR, -(SP)                       ; 0965
             00000000G  00         01   FB 0003F          CALLS   #1, BAS$$SIGNAL                            ;
                                        04 00046 3$:      RET                                                ; 0968

; Routine Size:  71 bytes,    Routine Base:  _BAS$CODE + 01F1
```

```
;  683          0969  1 %SBTTL 'BAS$CVTPF - Convert packed to float'
;  684          0970  1 GLOBAL ROUTINE BAS$CVTPF (                                  ! Convert packed to float
;  685          0971  1         DEST,                                               ! place to store conv. number
;  686          0972  1         SRC,                                                ! number to be converted
;  687          0973  1         SRCLEN,                                             ! number of digits in source
;  688          0974  1         SCALE                                               ! power of ten to mult src
;  689          0975  1     ) : NOVALUE =
;  690          0976  1
;  691          0977  1 !++
;  692          0978  1 ! FUNCTIONAL DESCRIPTION:
;  693          0979  1 !
;  694          0980  1 !       Converts a packed number to single floating.
;  695          0981  1 !
;  696          0982  1 ! CALLING SEQUENCE:
;  697          0983  1 !
;  698          0984  1 !       BAS$CVTPF (DEST.wf.r, SRC.rp.r, SRCLEN.rl.v, SCALE.rl.v)
;  699          0985  1 !
;  700          0986  1 ! FORMAL PARAMETERS:
;  701          0987  1 !
;  702          0988  1 !       DEST.wf.r       place to store the converted number
;  703          0989  1 !       SRC.rf.r        number to be converted
;  704          0990  1 !       SRCLEN.rl.v     number of digits in the source
;  705          0991  1 !       SCALE.rl.v      power of ten by which the internal
;  706          0992  1 !                       representation of the sourc must be
;  707          0993  1 !                       multiplied to scale the same as the
;  708          0994  1 !                       internal representation of the dest.
;  709          0995  1 !
;  710          0996  1 ! IMPLICIT INPUTS:
;  711          0997  1 !
;  712          0998  1 !       NONE
;  713          0999  1 !
;  714          1000  1 ! IMPLICIT OUTPUTS:
;  715          1001  1 !
;  716          1002  1 !       NONE
;  717          1003  1 !
;  718          1004  1 ! COMPLETION STATUS:
;  719          1005  1 !
;  720          1006  1 !       NONE
;  721          1007  1 !
;  722          1008  1 ! SIDE EFFECTS:
;  723          1009  1 !
;  724          1010  1 !       NONE
;  725          1011  1 !
;  726          1012  1 !--
;  727          1013  1
;  728          1014  2     BEGIN
;  729          1015  2
;  730          1016  2     OTS$CVTPF_R9 (.SCALE, .SRCLEN, .SRC, .DEST);
;  731          1017  2
;  732          1018  1     END;                                                    ! End of routine BAS$CVTPF


                                        0FFC 00000          .ENTRY  BAS$CVTPF, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 0970
                                                                    R10,R11                                         ;
```

```
     59         04   AC  D0 00002        MOVL    DEST, R9                          ; 1016
     58         08   AC  D0 00006        MOVL    SRC, R8                           ;
     57         0C   AC  D0 0000A        MOVL    SRCLEN, R7                        ;
     56         10   AC  D0 0000E        MOVL    SCALE, R6                         ;
          00000000G  00  16 00012        JSB     OTS$CVTPF_R9                      ;
                         04 00018        RET                                       ; 1018

; Routine Size:  25 bytes,    Routine Base:  _BAS$CODE + 0238
```

```
;  734          1019  1 %SBTTL 'BAS$CVTPD - Convert packed to double'
;  735          1020  1 GLOBAL ROUTINE BAS$CVTPD (                                  ! Convert packed to double
;  736          1021  1         DEST,                                               ! place to store conv. number
;  737          1022  1         SRC,                                                ! number to be converted
;  738          1023  1         SRCLEN,                                             ! number of digits in source
;  739          1024  1         SCALE                                               ! power of ten to mult src
;  740          1025  1     ) : NOVALUE =
;  741          1026  1
;  742          1027  1 !++
;  743          1028  1 ! FUNCTIONAL DESCRIPTION:
;  744          1029  1 !
;  745          1030  1 !       Converts a packed number to double floating.
;  746          1031  1 !
;  747          1032  1 ! CALLING SEQUENCE:
;  748          1033  1 !
;  749          1034  1 !       BAS$CVTPD (DEST.wp.r, SRC.rf.r, SRCLEN.rl.v, SCALE.rl.v)
;  750          1035  1 !
;  751          1036  1 ! FORMAL PARAMETERS:
;  752          1037  1 !
;  753          1038  1 !       DEST.wd.r       place to store the converted number
;  754          1039  1 !       SRC.rp.r        number to be converted
;  755          1040  1 !       SRCLEN.rl.v     number of digits in source
;  756          1041  1 !       SCALE.rl.v      power of ten by which the internal
;  757          1042  1 !                       representation of the sourc must be
;  758          1043  1 !                       multiplied to scale the same as the
;  759          1044  1 !                       internal representation of the dest.
;  760          1045  1 !
;  761          1046  1 ! IMPLICIT INPUTS:
;  762          1047  1 !
;  763          1048  1 !       NONE
;  764          1049  1 !
;  765          1050  1 ! IMPLICIT OUTPUTS:
;  766          1051  1 !
;  767          1052  1 !       NONE
;  768          1053  1 !
;  769          1054  1 ! COMPLETION STATUS:
;  770          1055  1 !
;  771          1056  1 !       NONE
;  772          1057  1 !
;  773          1058  1 ! SIDE EFFECTS:
;  774          1059  1 !
;  775          1060  1 !       NONE
;  776          1061  1 !
;  777          1062  1 !--
;  778          1063  1
;  779          1064  2     BEGIN
;  780          1065  2
;  781          1066  2     OTS$CVTPD_R9 (.SCALE, .SRCLEN, .SRC, .DEST);
;  782          1067  2
;  783          1068  1     END;                                                    ! End of routine BAS$CVTPD


                                                0FFC 00000          .ENTRY  BAS$CVTPD, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 1020
                                                                            R10,R11                                         ;
```

```
          59        04   AC  D0 00002        MOVL    DEST, R9                ; 1066
          58        08   AC  D0 00006        MOVL    SRC, R8                 ;
          57        0C   AC  D0 0000A        MOVL    SRCLEN, R7              ;
          56        10   AC  D0 0000E        MOVL    SCALE, R6               ;
             00000000G   00  16 00012        JSB     OTS$CVTPD_R9            ;
                             04 00018        RET                             ; 1068

; Routine Size:  25 bytes,    Routine Base:  _BAS$CODE + 0251
```

```
;  785          1069  1 %SBTTL 'BAS$CVTPG - Convert packed to gfloat'
;  786          1070  1 GLOBAL ROUTINE BAS$CVTPG (                          ! Convert packed to gfloat
;  787          1071  1           DEST,                                    ! place to store conv. number
;  788          1072  1           SRC,                                     ! number to be converted
;  789          1073  1           SRCLEN,                                  ! number of digits in source
;  790          1074  1           SCALE                                    ! power of ten to mult src
;  791          1075  1       ) : NOVALUE =
;  792          1076  1
;  793          1077  1 !++
;  794          1078  1 ! FUNCTIONAL DESCRIPTION:
;  795          1079  1 !
;  796          1080  1 !       Converts a packed number to g floating.
;  797          1081  1 !
;  798          1082  1 ! CALLING SEQUENCE:
;  799          1083  1 !
;  800          1084  1 !       BAS$CVTPG (DEST.wg.r, SRC.rp.r, SRCLEN.rl.v, SCALE.rl.v)
;  801          1085  1 !
;  802          1086  1 ! FORMAL PARAMETERS:
;  803          1087  1 !
;  804          1088  1 !       DEST.wg.r       place to store the converted number
;  805          1089  1 !       SRC.rp.r        number to be converted
;  806          1090  1 !       SRCLEN.rl.v     number of digits in the destination
;  807          1091  1 !       SCALE.rl.v      power of ten by which the internal
;  808          1092  1 !                       representation of the sourc must be
;  809          1093  1 !                       multiplied to scale the same as the
;  810          1094  1 !                       internal representation of the dest.
;  811          1095  1 !
;  812          1096  1 ! IMPLICIT INPUTS:
;  813          1097  1 !
;  814          1098  1 !       NONE
;  815          1099  1 !
;  816          1100  1 ! IMPLICIT OUTPUTS:
;  817          1101  1 !
;  818          1102  1 !       NONE
;  819          1103  1 !
;  820          1104  1 ! COMPLETION STATUS:
;  821          1105  1 !
;  822          1106  1 !       NONE
;  823          1107  1 !
;  824          1108  1 ! SIDE EFFECTS:
;  825          1109  1 !
;  826          1110  1 !       NONE
;  827          1111  1 !
;  828          1112  1 !--
;  829          1113  1
;  830          1114  2     BEGIN
;  831          1115  2
;  832          1116  2     OTS$CVTPG_R9 (.SCALE, .SRCLEN, .SRC, .DEST);
;  833          1117  2
;  834          1118  1     END;                                           ! End of routine BAS$CVTPG


                                    0FFC 00000          .ENTRY  BAS$CVTPG, Save R2,R3,R4,R5,R6,R7,R8,R9,-    ; 1070
                                                                R10,R11                                          ;
```

```
              59         04  AC  D0 00002        MOVL    DEST, R9                        ; 1116
              58         08  AC  D0 00006        MOVL    SRC, R8                         ;
              57         0C  AC  D0 0000A        MOVL    SRCLEN, R7                      ;
              56         10  AC  D0 0000E        MOVL    SCALE, R6                       ;
                  00000000G  00  16 00012        JSB     OTS$CVTPG_R9                    ;
                                 04 00018        RET                                     ; 1118

; Routine Size:  25 bytes,    Routine Base:  _BAS$CODE + 026A
```

```
;  836     1119  1 %SBTTL 'BAS$CVTPH - Convert packed to hfloat'
;  837     1120  1 GLOBAL ROUTINE BAS$CVTPH (                                  ! Convert packed to hfloat
;  838     1121  1         DEST,                                               ! place to store conv. number
;  839     1122  1         SRC,                                                ! number to be converted
;  840     1123  1         SRCLEN,                                             ! number of digits in the source
;  841     1124  1         SCALE                                               ! power of ten to mult src
;  842     1125  1     ) : NOVALUE =
;  843     1126  1
;  844     1127  1 !++
;  845     1128  1 ! FUNCTIONAL DESCRIPTION:
;  846     1129  1 !
;  847     1130  1 !       Converts a packed number to hfloating.
;  848     1131  1 !
;  849     1132  1 ! CALLING SEQUENCE:
;  850     1133  1 !
;  851     1134  1 !       BAS$CVTPH (DEST.wh.r, SRC.rp.r, SRCLEN.rl.v, SCALE.rl.v)
;  852     1135  1 !
;  853     1136  1 ! FORMAL PARAMETERS:
;  854     1137  1 !
;  855     1138  1 !       DEST.wh.r       place to store the converted number
;  856     1139  1 !       SRC.rp.r        number to be converted
;  857     1140  1 !       SRCLEN.rl.v     number of digits in the source
;  858     1141  1 !       SCALE.rl.v      power of ten by which the internal
;  859     1142  1 !                       representation of the sourc must be
;  860     1143  1 !                       multiplied to scale the same as the
;  861     1144  1 !                       internal representation of the dest.
;  862     1145  1 !
;  863     1146  1 ! IMPLICIT INPUTS:
;  864     1147  1 !
;  865     1148  1 !       NONE
;  866     1149  1 !
;  867     1150  1 ! IMPLICIT OUTPUTS:
;  868     1151  1 !
;  869     1152  1 !       NONE
;  870     1153  1 !
;  871     1154  1 ! COMPLETION STATUS:
;  872     1155  1 !
;  873     1156  1 !       NONE
;  874     1157  1 !
;  875     1158  1 ! SIDE EFFECTS:
;  876     1159  1 !
;  877     1160  1 !       NONE
;  878     1161  1 !
;  879     1162  1 !--
;  880     1163  1
;  881     1164  2     BEGIN
;  882     1165  2
;  883     1166  2     OTS$CVTPH_R9 (.SCALE, .SRCLEN, .SRC, .DEST);
;  884     1167  2
;  885     1168  1     END;                                                    ! End of routine BAS$CVTPH


                    0FFC 00000          .ENTRY  BAS$CVTPH, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 1120
                                                R10,R11                                     ;
```

```
        59        04  AC  D0 00002        MOVL    DEST, R9              ; 1166
        58        08  AC  D0 00006        MOVL    SRC, R8               ;
        57        0C  AC  D0 0000A        MOVL    SRCLEN, R7            ;
        56        10  AC  D0 0000E        MOVL    SCALE, R6             ;
           00000000G  00  16 00012        JSB     OTS$CVTPH_R9          ;
                          04 00018        RET                           ; 1168

; Routine Size:  25 bytes,    Routine Base:  _BAS$CODE + 0283


;  886          1169  1 !<BLF/PAGE>
```

```
;  888         1170  1 END                                          ! End of module BAS$CVTRP
;  889         1171  1
;  890         1172  0 ELUDOM
```

; PSECT SUMMARY

Name	Bytes	Attributes
_BAS$CODE	668	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

; Library Statistics

File	-------- Symbols -------- Total	Loaded	Percent	Pages Mapped	Processing Time
_255DUA28:[SYSLIB]STARLET.L32;1	9776	0	0	581	00:01.1

; COMMAND QUALIFIERS

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

```
; Size:           668 code + 0 data bytes
; Run Time:          00:15.1
; Elapsed Time:      00:32.6
; Lines/CPU Min:     4669
; Lexemes/CPU-Min: 14836
; Memory Used:   67 pages
; Compilation Complete
```

BASDET
LIS

BASDISPAT
LIS

BASCVTTP
LIS

BASDELETE
LIS

BASDATETI
LIS

BASECHO
LIS

BASCVTRP
LIS