```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

FILEID**OTSCVTRHP

H 5

```
  OOOOOO    TTTTTTTTTT    SSSSSSSS    CCCCCCCC  VV      VV  TTTTTTTTTT  RRRRRRRR    HH      HH  PPPPPPPP
  OOOOOO    TTTTTTTTTT    SSSSSSSS    CCCCCCCC  VV      VV  TTTTTTTTTT  RRRRRRRR    HH      HH  PPPPPPPP
OO      OO      TT      SS          CC          VV      VV      TT      RR      RR  HH      HH  PP      PP
OO      OO      TT      SS          CC          VV      VV      TT      RR      RR  HH      HH  PP      PP
OO      OO      TT      SS          CC          VV      VV      TT      RR      RR  HH      HH  PP      PP
OO      OO      TT      SS          CC          VV      VV      TT      RR      RR  HH      HH  PP      PP
OO      OO      TT        SSSSSS    CC          VV      VV      TT      RRRRRRRR    HHHHHHHHHH  PPPPPPPP
OO      OO      TT        SSSSSS    CC          VV      VV      TT      RRRRRRRR    HHHHHHHHHH  PPPPPPPP
OO      OO      TT              SS  CC          VV      VV      TT      RR  RR      HH      HH  PP
OO      OO      TT              SS  CC          VV      VV      TT      RR  RR      HH      HH  PP
OO      OO      TT              SS  CC            VV  VV        TT      RR    RR    HH      HH  PP              ....
OO      OO      TT              SS  CC            VV  VV        TT      RR    RR    HH      HH  PP              ....
  OOOOOO        TT      SSSSSSSS      CCCCCCCC      VV          TT      RR      RR  HH      HH  PP              ....
  OOOOOO        TT      SSSSSSSS      CCCCCCCC      VV          TT      RR      RR  HH      HH  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
```

Table of contents

(2)	49	HISTORY	; Detailed Current Edit History
(3)	63	DECLARATIONS	
(4)	101	OTS$CVTRHP_R9	

```
0000      .TITLE  OTS$CVTRHP_R9    Convert Rounded H Floating to Packed
0000      .IDENT  /1-006/          ; File: OTSCVTRHP.MAR Edit: MDL1006
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ; FACILITY: LANGUAGE INDEPENDENT SUPPORT
0000    ;++
0000    ; ABSTRACT:
0000    ;       This module contains the routine that converts H floating
0000    ;       numbers to packed with rounding.
0000    ;
0000    ;
0000    ;--
0000    ;
0000    ; VERSION: 1
0000    ;
0000    ; HISTORY:
0000    ;
0000    ; AUTHOR:
0000    ;       Pamela Levesque, 21-Jan-1982
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;
0000    ;
```

```
0000            .SBTTL  HISTORY         ; Detailed Current Edit History
0000
0000
0000    ; Edit History for Version 1 of OTSCVTRHP
0000    ;
0000    ; 1-001 - Original.  PLL 21-Jan-1982
0000    ; 1-002 - Call h floating conversion routine.  PLL 2-Mar-1982
0000    ; 1-003 - Expect a 2 digit exponent from the conversion routine. PLL 8-Mar-1982
0000    ; 1-004 - determine the number of digits in the exponent on the fly.
0000    ;         MDL 9-Apr-1984
0000    ; 1-005 - Make sure result will not overflow destination after calculating
0000    ;         longword exponent.  DG 9-Apr-1984
0000    ; 1-006 - complete edit 1-004.  MDL 12-Apr-1984
```

```
    0000        .SBTTL  DECLARATIONS
    0000
    0000  ;
    0000  ; INCLUDE FILES:
    0000  ;
    0000        $DSCDEF
    0000
    0000  ;
    0000  ; EXTERNAL SYMBOLS:
    0000
    0000        .DSABL  GBL                 ; Prevent undeclared symbols from being
    0000                                    ; automatically global
    0000
    0000        .EXTRN  OTS$CNVOUT_H        ; E-format conversion
    0000        .EXTRN  LIB$STOP            ; Error halt
    0000        .EXTRN  OTS$_FATINTERR      ; Fatal internal error code
    0000  ;
    0000
    0000  ;
    0000  ; MACROS:
    0000        NONE
    0000  ;
    0000
    0000  ;
    0000  ; PSECT DECLARATIONS:
00000000        .PSECT  _OTS$CODE           PIC, SHR, LONG, EXE, NOWRT,-
    0000                                    USR, CON, REL, LCL, RD
    0000
    0000  ;
    0000  ; EQUATED SYMBOLS:
    0000        NONE
    0000  ;
    0000
    0000  ;
    0000  ; OWN STORAGE:
    0000        NONE
    0000  ;
```

```
                         0000        .SBTTL  OTS$CVTRHP_R9
                         0000
                         0000  ;++
                         0000  ; FUNCTIONAL DESCRIPTION:
                         0000  ;
                         0000  ;       Converts an h floating number to packed with rounding.
                         0000  ;
                         0000  ; CALLING SEQUENCE:
                         0000  ;
                         0000  ;       JSB OTS$CVTRHP_R9 (scale.rl.v, src.rh.r, dstlen.rl.v, dst.wp.r)
                         0000  ;
                         0000  ;       Arguments are passed in R6, R7, R8 and R9.
                         0000  ;
                         0000  ; INPUT PARAMETERS:
                         0000  ;
                         0000  ;       SCALE.rl.v              The power of ten by which the internal
                         0000  ;                               representation of the source must be
                         0000  ;                               multiplied to scale the same as the
                         0000  ;                               internal representation of the dest.
                         0000  ;       SRC.rh.r                The number to be converted
                         0000  ;       DSTLEN.rl.v             The number of digits in the destination
                         0000  ;
                         0000  ; IMPLICIT INPUTS:
                         0000  ;
                         0000  ;       All of the trap bits in the PSL are assumed off.
                         0000  ;
                         0000  ; OUTPUT PARAMETERS:
                         0000  ;
                         0000  ;       DST.wp.r                The place to store the converted number
                         0000  ;
                         0000  ; IMPLICIT OUTPUTS:
                         0000  ;
                         0000  ;       NONE
                         0000  ;
                         0000  ; FUNCTION VALUE:
                         0000  ;
                         0000  ;       1 = SUCCESS, 0 = FAILURE
                         0000  ;
                         0000  ; SIDE EFFECTS:
                         0000  ;
                         0000  ;       Destroys registers R0 through R9.
                         0000  ;
                         0000  ;--
                         0000
                         0000
                         0000  OTS$CVTRHP_R9::
             5E   38  C2 0000        SUBL2   #56,SP                  ; Allocate temp space
 6E  38  00  6E   00  2C 0003        MOVC5   #0,(SP),#^A'',#56,(SP)  ; Clear temp space
             6E   67 70FD 0009       MOVH    (R7),(SP)               ; Get input number
                         000D  ;
                         000D  ; Make a descriptor for the temporary string.
                         000D  ;
          10 AE       9F 000D        PUSHAB  16(SP)                  ; Address = space reserved
             7E   01  90 0010        MOVB    #DSC$K_CLASS_S,-(SP)    ; Class = static
             7E   0E  90 0013        MOVB    #DSC$K_DTYPE_T,-(SP)    ; Data type = ASCII string
             7E   26  B0 0016        MOVW    #38,-(SP)               ; Length in bytes
                         0019  ;
```

```
                              0019       ; Call OTS$CNVOUT.
                              0019       ;
                  1F    DD    0019               PUSHL   #31                      ; Digits in fraction
               04 AE    9F    001B               PUSHAB  4(SP)                    ; Output string descriptor
               10 AE    9F    001E               PUSHAB  16(SP)                   ; Number to convert
         00000000'GF 03    FB    0021            CALLS   #3,G^OTS$CNVOUT_H        ; Call conversion routine
               63 50    E9    0028               BLBC    R0,20$                   ; Should never fail
                              002B       ;
                              002B       ; Determine the number of digits in the exponent.
                              002B       ;
                              002B                                                ; find beginning of exponent (sign byte)
       1C AE    22   2B    3A    002B            LOCC    #^A'+',#34,28(SP)        ; look for sign byte (+)
                     07    12    0030            BNEQ    5$                       ; branch if we found it
       1C AE    22   2D    3A    0032            LOCC    #^A'-',#34,28(SP)        ; look for sign byte (-)
                     55    13    0037            BEQL    20$                      ; branch to error if we didn't find it
                              0039                                                ;
           08 AE    51    D0    0039     5$:     MOVL    R1,8(SP)                 ; save location of sign byte
                              003D                                                ; (contained in R1 after above LOCC)
                              003D                                                ;
       08 BE    05   00    3A    003D            LOCC    #^A'',#5,@B^8(SP)        ; look for NUL (end of string)
    0C AE    51    08 AE    C3    0042           SUBL3   8(SP),R1,12(SP)          ; determine number of digits by subtracting
                              0048                                                ; location of beginning of exponent from
                              0048                                                ; location of end of exponent, and save it
               0C AE    D7    0048               DECL    12(SP)                   ; adjust for sign byte
                              004B                                                ; at this point, 12(SP) = length of exponent
                              004B       ;
                              004B       ; Convert the exponent and correct for scale factor.
                              004B       ;
  6E    0C AE    08 BE    0C AE    09    004B    CVTSP   12(SP),@B^8(SP),12(SP),(SP)      ; Make packed exponent
           50    6E    0C AE    36    0053       CVTPL   12(SP),(SP),R0           ; Make longword exponent
                              0058               ;
                              0058               ; Here must check that the result will not overflow
                              0058               ; the destination.  The destination length (total number of
                              0058               ; digits) minus the scale factor will give the number of digits
                              0058               ; to the left of the decimal point.  If that number is greater than
                              0058               ; the longword exponent, then the result WILL overflow the dest.
                              0058               ;
           54    58    56    C3    0058          SUBL3   R6,R8,R4                 ; Determine num of digits on left
                 54    50    D1    005C          CMPL    R0,R4                    ; Compare with longword exponent
                       23    14    005F          BGTR    15$                      ; Branch if overflowed
                              0061
           56    E2 A640    9E    0061           MOVAB   -30(R6)[R0],R6           ; Correct for fraction size and scale
                              0066       ;
                              0066       ; Convert the fraction to packed.
                              0066       ;
       1A AE    18 AE    90    0066              MOVB    24(SP),26(SP)            ; Move sign over "."
     6E    1E    1A AE    1E    09    006B       CVTSP   #30,26(SP),#30,(SP)      ; Make packed fraction
  69    58    05    6E    1E    56    F8    0071 ASHP    R6,#30,(SP),#5,R8,(R9)   ; Scale to destination
                              0078                                                ;    (also clears R0)
                       02    1D    0078          BVS     10$                      ; Branch if overflowed
                       50    D6    007A          INCL    R0                       ; Indicate success, R0 = 1
        5E    00000040 8F    C0    007C    10$:  ADDL2   #64,SP                   ; Delete stack temps
                       05    0083                RSB
                              0084
                       50    D4    0084    15$:  CLRL    R0                       ; Indicate failure, R0 = 0
        5E    00000040 8F    C0    0086          ADDL2   #64,SP                   ; Delete stack temps
                       05    008D                RSB
```

```
                              008E   215 ;
                              008E   216 ; Come here on failure of OTS$CNVOUT.  This should never happen.
                              008E   217 ;
        00000000'8F      DD   008E   218 20$:    PUSHL   #OTS$_FATINTERR         ; OTS fatal error message
   00000000'GF     01    FB   0094   219         CALLS   #1,G^LIB$STOP           ; Signal and don't return
                              009B   220 ;
                              009B   221         .END
```

```
DSC$K_CLASS_S                  = 00000001
DSC$K_DTYPE_T                  = 0000000E
LIB$STOP                         ********    X   00
OTS$CNVOUT_H                     ********    X   00
OTS$CVTRHP_R9                    00000000 RG     02
OTS$_FATINTERR                   ********    X   00
```

```
                        +-----------------+
                        ! Psect synopsis !
                        +-----------------+
```

```
PSECT name         Allocation             PSECT No.   Attributes
----------         ----------             ---------   ----------
.  ABS  .          00000000  (     0.)    00 (   0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$              00000000  (     0.)    01 (   1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
_OTS$CODE          0000009B  (   155.)    02 (   2.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC LONG
```

```
                        +------------------------+
                        ! Performance indicators !
                        +------------------------+
```

```
Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                      32    00:00:00.03    00:00:01.01
Command processing                 129    00:00:00.31    00:00:01.82
Pass 1                             132    00:00:01.24    00:00:05.98
Symbol table sort                    0    00:00:00.11    00:00:00.71
Pass 2                              57    00:00:00.36    00:00:01.97
Symbol table output                  2    00:00:00.01    00:00:00.20
Psect synopsis output                3    00:00:00.02    00:00:00.02
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals               357    00:00:02.08    00:00:11.71
```

The working set limit was 1050 pages.
9083 bytes (18 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 135 non-local and 4 local symbols.
221 source lines were read in Pass 1, producing 10 object records in Pass 2.
8 pages of virtual memory were used to define 7 macros.

```
                        +--------------------------+
                        ! Macro library statistics !
                        +--------------------------+
```

```
Macro library name                        Macros defined
------------------                        --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                4
```

190 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

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

0212 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

OTSCVTPD
LIS

OTSCVTTF
LIS

OTSCVTRFP
LIS

OTSCVTRT
LIS

OTSCVTTOL
LIS

OTSCVTPG
LIS

OTSCVTTR
LIS

OTSCVTRHP
LIS

OTSLINKAG
LIS

OTSCVTRDP
LIS

OTSCVTTIL
LIS

OTSCVTTLL
LIS

OTSCVTPF
LIS

OTSCVTRGP
LIS

OTSMOVE
LIS

OTSMSG
LIS

OTSCVTPH
LIS

OTSLUN
LIS