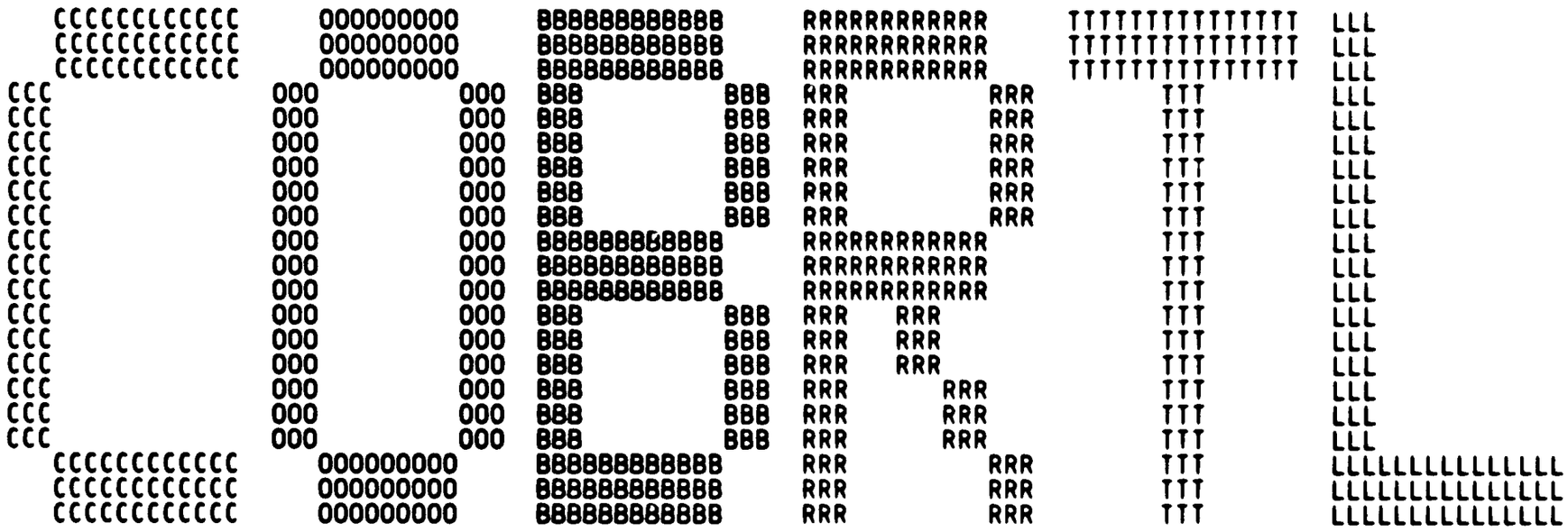
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBBBBBBBBBBB RRRRRRRRRRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCC 000 000 BBB BBB RRR RRR TTT LLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
CCCCCCCCCCC 000000000 BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL

```
  CCCCCCCC    000000    BBBBBBBB      CCCCCCCC  VV      VV  TTTTTTTTTT  RRRRRRRR      QQQQQQ    PPPPPPPP
  CCCCCCCC    000000    BBBBBBBB      CCCCCCCC  VV      VV  TTTTTTTTTT  RRRRRRRR      QQQQQQ    PPPPPPPP
CC           00      00  BB      BB  CC          VV      VV      TT      RR      RR  QQ      QQ  PP      PP
CC           00      00  BB      BB  CC          VV      VV      TT      RR      RR  QQ      QQ  PP      PP
CC           00      00  BB      BB  CC          VV      VV      TT      RR      RR  QQ      QQ  PP      PP
CC           00      00  BB      BB  CC          VV      VV      TT      RR      RR  QQ      QQ  PP      PP
CC           00      00  BBBBBBBB    CC          VV      VV      TT      RRRRRRRR    QQ      QQ  PPPPPPPP
CC           00      00  BBBBBBBB    CC          VV      VV      TT      RRRRRRRR    QQ      QQ  PPPPPPPP
CC           00      00  BB      BB  CC          VV      VV      TT      RR   RR     QQ  QQ  QQ  PP
CC           00      00  BB      BB  CC          VV      VV      TT      RR   RR     QQ  QQ  QQ  PP
CC           00      00  BB      BB  CC            VV  VV        TT      RR    RR    QQ    QQ    PP
CC           00      00  BB      BB  CC            VV  VV        TT      RR    RR    QQ    QQ    PP
  CCCCCCCC    000000    BBBBBBBB      CCCCCCCC      VV          TT      RR      RR    QQQQ  QQ  PP
  CCCCCCCC    000000    BBBBBBBB      CCCCCCCC      VV          TT      RR      RR    QQQQ  QQ  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

```
0000     1          .TITLE  COB$CVTRQP_R9   COBOL Convert Rounded Quad to Packed
0000     2          .IDENT  /1-007/         ; File: COBCVTRQP.MAR
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28  ; FACILITY: COBOL TYPE CONVERSION
0000    29  ;++
0000    30  ; ABSTRACT:
0000    31  ;       This module contains the routine which converts signed quadwords
0000    32  ;       to packed with rounding.
0000    33  ;
0000    34  ;
0000    35  ;--
0000    36  ;
0000    37  ; VERSION: 1
0000    38  ;
0000    39  ; HISTORY:
0000    40  ;
0000    41  ; AUTHOR:
0000    42  ;       John Sauter, 29-DEC-78
0000    43  ;
0000    44  ; MODIFIED BY:
0000    45  ;
0000    46  ;
0000    47  ;
```

```
0000     49          .SBTTL  HISTORY         , Detailed Current Edit History
0000     50
0000     51
0000     52 ; Edit History for Version 1 of COBCVTRQP
0000     53 ;
0000     54 ; 1-001 - Original from Marty Jack.
0000     55 ; 1-002 - Use byte displacement lengths.  JBS 29-DEC-78
0000     56 ; 1-003 - Make entry point symbol global.  JBS 03-JAN-1979
0000     57 ; 1-004 - Minor editing cleanup.  JBS 11-JAN-1979
0000     58 ; 1-005 - Bug fixes and cleanup.  MLJ 10-Mar-1979
0000     59 ; 1-006 - 19 digit temps.  MLJ 13-Mar-1979
0000     60 ; 1-007 - Cosmetic changes.  RKR 18-OCT-79
```

```
                                        62          .SBTTL  DECLARATIONS
                                        63
                                        64 ;
                                        65 ; INCLUDE FILES:
                                        66 ;
                                        67
                                        68 ;
                                        69 ; EXTERNAL SYMBOLS:
                                        70 ;       NONE
                                        71 ;
                                        72
                                        73 ;
                                        74 ; MACROS:
                                        75 ;       NONE
                                        76 ;
                                        77
                                        78 ;
                                        79 ; PSECT DECLARATIONS:
                                  00000000  80          .PSECT  _COB$CODE       PIC, SHR, LONG, EXE, NOWRT
                                        81
                                        82 ;
                                        83 ; EQUATED SYMBOLS:
                                        84 ;       NONE
                                        85 ;
                                        86
                                        87 ;
                                        88 ; OWN STORAGE:
                                        89 ;
                                        90 ;+
                                        91 ; The following constant has the value 2**32.  It is used for scaling
                                        92 ; the high 32 bits and for compensating for unsigned arithmetic.
                                        93 ;-
                6C 29 67 49 29 04       94 BIAS:    .PACKED 4294967296      ; 2**32
                          0000000A      95 BIAS_DIGITS=10
                                        96 ;
```

```
                                        0006    98          .SBTTL  COB$CVTRQP_R9
                                        0006    99
                                        0006   100 ;++
                                        0006   101 ; FUNCTIONAL DESCRIPTION:
                                        0006   102 ;
                                        0006   103 ;       Converts a 64-bit (quadword) integer to packed with rounding.
                                        0006   104 ;
                                        0006   105 ; CALLING SEQUENCE:
                                        0006   106 ;
                                        0006   107 ;       JSB COB$CVTRQP_R9 (scale.rl.v, src.rq.r, dstlen.rl.v, dst.wp.r)
                                        0006   108 ;
                                        0006   109 ;       Arguments are passed in R6, R7, R8 and R9.
                                        0006   110 ;
                                        0006   111 ; INPUT PARAMETERS:
                                        0006   112 ;
                                        0006   113 ;       SCALE.rl.v              The power of ten by which the internal
                                        0006   114 ;                               representation of the source must be
                                        0006   115 ;                               multiplied to scale the same as the
                                        0006   116 ;                               internal representation of the dest.
                                        0006   117 ;       SRC.rq.r                The number to be converted
                                        0006   118 ;       DSTLEN.rl.v             The number of digits in the destination
                                        0006   119 ;
                                        0006   120 ; IMPLICIT INPUTS:
                                        0006   121 ;
                                        0006   122 ;       All of the trap bits in the PSL are assumed off.
                                        0006   123 ;
                                        0006   124 ; OUTPUT PARAMETERS:
                                        0006   125 ;
                                        0006   126 ;       DST.wp.r                The place to store the converted number
                                        0006   127 ;
                                        0006   128 ; IMPLICIT OUTPUTS:
                                        0006   129 ;
                                        0006   130 ;       NONE
                                        0006   131 ;
                                        0006   132 ; COMPLETION CODES:
                                        0006   133 ;
                                        0006   134 ;       NONE
                                        0006   135 ;
                                        0006   136 ; SIDE EFFECTS:
                                        0006   137 ;
                                        0006   138 ;       Destroys registers R0 through R9.
                                        0006   139 ;
                                        0006   140 ;--
                                        0006   141
                                        0006   142 COB$CVTRQP_R9::
                                  5E    0006   143          SUBL2   #20,SP                  ; Allocate temp space
                              04  A7    0009   144          CMPV    #31,#1,(R7),4(R7)       ; Is number in longword range?
                                  13    000F   145          BNEQ    11$                     ; Br if not to do slower code
                                  6E    0011   146          CVTLP   (R7),#10,(SP)           ; Convert low order longword
                              69  58    0015   147          ASHP    R6,#10,(SP),#5,R8,(R9)  ; Scale to destination
                                        001C   148                                          ;     (also clears R0)
                                  02    001C   149          BVS     10$                     ; Br if overflowed
                                  50    001E   150          INCL    R0                      ; Indicate success, R0 = 1
                                  5E    0020   151 10$:     ADDL2   #20,SP                  ; Deallocate temp space
                                  05    0023   152          RSB                             ; Return
                                        0024   153 ;+
                                        0024   154 ; Come here if the input number is not in longword range.  Sum the
```

```
                                0024   155 ; converted high order longword, multiplied by 2**32, and the converted
                                0024   156 ; low order longword, considered as an unsigned number.
                                0024   157 ;-
            6E   0A   04 A7     F9 0024   158 11$:    CVTLP   4(R7),#10,(SP)           ; Convert high order longword
 13   6E   0A   03 AF   0A     25 0029   159          MULP    #BIAS_DIGITS,BIAS,#10,(SP),#19,8(SP)
                     08 AE                0030
                                0032   160                                            ; Multiply by 2**32
                6E   0A   67     F9 0032   161          CVTLP   (R7),#10,(SP)           ; Convert low order longword
                          06     18 0036   162          BGEQ    12$                     ; Br if nonnegative
        6E   0A   C4 AF   0A     20 0038   163          ADDP4   #BIAS_DIGITS,BIAS,#10,(SP)
                                003E   164                                            ; Correct for signed conversion
     08 AE   13   6E   0A     20 003E   165 12$:    ADDP4   #10,(SP),#19,8(SP)      ; Sum low and high order parts
69   58   05   63   03   56   F8 0044   166          ASHP    R6,#19,(R3),#5,R8,(R9)  ; Scale to destination
                                004B   167                                            ;     (also clears R0)
                          02     1D 004B   168          BVS     13$                     ; Br if overflowed
                          50     D6 004D   169          INCL    R0                      ; Indicate success, R0 = 1
                     5E   14     C0 004F   170 13$:    ADDL2   #20,SP                  ; Deallocate temp space
                          05     05 0052   171          RSB                             ; Return
                                0053   172 ;
                                0053   173          .END
```

```
BIAS              00000000 R      01
BIAS_DIGITS     = 0000000A
COB$CVTRQP_R9     00000006 RG     01
```

Psect synopsis

```
PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
. ABS .                         00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
_COB$CODE                       00000053  (   83.)  01 (  1.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD  NOWRT NOVEC LONG
```

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:01.66
Command processing	117	00:00:00.32	00:00:02.40
Pass 1	69	00:00:00.27	00:00:01.49
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	48	00:00:00.22	00:00:02.32
Symbol table output	2	00:00:00.01	00:00:00.01
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	269	00:00:00.88	00:00:07.90

```
The working set limit was 750 pages.
2204 bytes (5 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 3 non-local and 4 local symbols.
173 source lines were read in Pass 1, producing 8 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.
```

Macro library statistics

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

```
0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

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

COBCVTRPQ
LIS

COBCVTDQ
LIS

COBCANCEL
LIS

COBCVTQP
LIS

COBACCTIM
LIS

COBCVTPQ
LIS

COBCVTRFQ
LIS

COBCVTQF
LIS

COBCVTRQP
LIS

COBCALL
LIS

COBCVTFQ
LIS

COBACCEPT
LIS

COBCVTRDQ
LIS

COBCVTQD
LIS

COBDIVQ
LIS

COBFINDNA
LIS

COBDBEXCE
LIS

COBEXPI
LIS

COBDEEDIT
LIS

COBDISPLA
LIS

COBESCGEN
LIS

COBERROR
LIS

COBDHANDL
LIS