COBRTL

```
CCCCCCCC   000000    BBBBBBBB    CCCCCCCC  VV      VV  TTTTTTTTTT  PPPPPPPP      QQQQQQ
CCCCCCCC   000000    BBBBBBBBB   CCCCCCCC  VV      VV  TTTTTTTTTT  PPPPPPPPP     QQQQQQ
CC         00    00  BB      BB  CC        VV      VV      TT      PP      PP  QQ      QQ
CC         00    00  BB      BB  CC        VV      VV      TT      PP      PP  QQ      QQ
CC         00    00  BB      BB  CC        VV      VV      TT      PP      PP  QQ      QQ
CC         00    00  BBBBBBBB    CC        VV      VV      TT      PPPPPPPP    QQ      QQ
CC         00    00  BBBBBBBB    CC        VV      VV      TT      PPPPPPPP    QQ      QQ
CC         00    00  BB      BB  CC        VV      VV      TT      PP          QQ      QQ
CC         00    00  BB      BB  CC         VV    VV       TT      PP          QQ   QQ QQ
CC         00    00  BB      BB  CC         VV    VV       TT      PP          QQ    QQ QQ
CC         00    00  BB      BB  CC          VV  VV        TT      PP          QQ     QQ  ....
CCCCCCCC   000000    BBBBBBBB    CCCCCCCC     VV           TT      PP          QQQQ QQ    ....
CCCCCCCC   000000    BBBBBBBB    CCCCCCCC     VV           TT      PP          QQQQ QQ    ....
```

```
LL           IIIIII    SSSSSSSS
LL           IIIIII    SSSSSSSS
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
```

```
0000      1            .TITLE  COB$CVTPQ_R9    COBOL Convert Packed to Quad
0000      2            .IDENT  /1-006/         ; File: COBCVTPQ.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 packed
0000     32    ;       decimal numbers to quadword (64-bit) binary.
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, 16-JAN-1979
0000     43    ;
0000     44    ; MODIFIED BY:
0000     45    ;
0000     46    ;
```

COB$CVTPQ_R9
1-006

I 12

COBOL Convert Packed to Quad   15-SEP-1984 23:39:37 VAX/VMS Macro V04-00   Page 2
HISTORY ; Detailed Current Edit History 6-SEP-1984 10:43:23 [COBRTL.SRC]COBCVTPQ.MAR;1   (2)

```
0000    48          .SBTTL  HISTORY          ; Detailed Current Edit History
0000    49
0000    50
0000    51 ; Edit History for Version 1 of COBCVTPQ
0000    52 ;
0000    53 ; 1-001 - Original.  JBS 16-JAN-1979
0000    54 ; 1-002 - Bug fixes and cleanup.  MLJ 10-Mar-1979
0000    55 ; 1-003 - 19 digit temps.  MLJ 13-Mar-1979
0000    56 ; 1-004 - Correct problem with high order longword.  22-Mar-1979
0000    57 ; 1-005 - Correct round towards zero problem.  PDG 12-Jul-1979
0000    58 ; 1-006 - Cosmetic changes.  RKR 18-OCT-79
```

```
                         0000       60                .SBTTL  DECLARATIONS
                         0000       61
                         0000       62 ;
                         0000       63 ; INCLUDE FILES:
                         0000       64 ;
                         0000       65
                         0000       66
                         0000       67 ; EXTERNAL SYMBOLS:
                         0000       68 ;     NONE
                         0000       69 ;
                         0000       70
                         0000       71
                         0000       72 ; MACROS:
                         0000       73 ;     NONE
                         0000       74 ;
                         0000       75
                         0000       76
                         0000       77 ; PSECT DECLARATIONS:
                  00000000       78        .PSECT  _COB$CODE        PIC, SHR, LONG, EXE, NOWRT
                         0000       79
                         0000       80 ;
                         0000       81 ; EQUATED SYMBOLS:
                         0000       82 ;     NONE
                         0000       83 ;
                         0000       84
                         0000       85
                         0000       86 ; OWN STORAGE:
                         0000       87
                         0000       88 ;+
                         0000       89 ; The following constant has the value 2**32.  It is used for scaling
                         0000       90 ; the high 32 bits and for compensating for unsigned arithmetic.
                         0000       91 ;-
6C 29 67 49 29 04        0000       92 BIAS:   .PACKED 4294967296       ; 2**32
                         0006       93 ;+
                         0006       94 ; The following constant is 2**32-1.  It's subtracted from negative numbers,
                         0006       95 ; to compensate for rounding towards zero.
                         0006       96 ;-
5C 29 67 49 29 04        0006       97 BIAS_1: .PACKED 4294967295
        0000000A         000C       98 BIAS_DIGITS=10
                         000C       99 ;
```

```
                                  000C    101              .SBTTL  COB$CVTPQ_R9
                                  000C    102
                                  000C    103  ;++
                                  000C    104  ; FUNCTIONAL DESCRIPTION:
                                  000C    105  ;
                                  000C    106  ;       Converts packed to quadword (64-bit integer)
                                  000C    107  ;
                                  000C    108  ; CALLING SEQUENCE:
                                  000C    109  ;
                                  000C    110  ;       JSB COB$CVTPQ_R9 (scale.rl.v, srclen.rl.v, src.rp.r, dst.wq.r)
                                  000C    111  ;
                                  000C    112  ;       Arguments are passed in R6, R7, R8 and R9.
                                  000C    113  ;
                                  000C    114  ; INPUT PARAMETERS:
                                  000C    115  ;
                                  000C    116  ;       SCALE.rl.v                    The power of ten by which the internal
                                  000C    117  ;                                     representation of the source must be
                                  000C    118  ;                                     multiplied to scale the same as the
                                  000C    119  ;                                     internal representation of the dest.
                                  000C    120  ;       SRCLEN.rl.v                   The number of digits in the source
                                  000C    121  ;       SRC.rp.r                      The number to be converted
                                  000C    122  ;
                                  000C    123  ; IMPLICIT INPUTS:
                                  000C    124  ;
                                  000C    125  ;       All of the trap bits in the PSL are assumed off.
                                  000C    126  ;
                                  000C    127  ; OUTPUT PARAMETERS:
                                  000C    128  ;
                                  000C    129  ;       DST.wq.r                      The place to store the converted number
                                  000C    130  ;
                                  000C    131  ; IMPLICIT OUTPUTS:
                                  000C    132  ;
                                  000C    133  ;       NONE
                                  000C    134  ;
                                  000C    135  ; FUNCTION VALUE:
                                  000C    136  ;
                                  000C    137  ;       1 = SUCCESS, 0 = FAILURE
                                  000C    138  ;
                                  000C    139  ; SIDE EFFECTS:
                                  000C    140  ;
                                  000C    141  ;       Destroys registers R0 through R9.
                                  000C    142  ;--
                                  000C    143
                                  000C    144
                                  000C    145
                                  000C    146  COB$CVTPQ_R9::
            5E  18  C2          000C    147              SOBL2   #24,SP                ; Make room for temp storage
6E  13  00  68  57  56  F8      000F    148              ASHP    R6,R7,(R8),#0,#19,(SP) ; Scale and integerize number
                                  0016    149  ;                                       (also clears R0)
                    0D  1D      0016    150              BVS     11$                   ; If overflow, won't fit in 64 bits
                                  0018    151  ;+
                                  0018    152  ; Since quadwords often have their high 32 bits unused, try to convert
                                  0018    153  ; the packed number to a longword.  If it succeeds, we need only spread
                                  0018    154  ; the sign bit.  If it fails we will have more work to do.
                                  0018    155  ;-
            69  6E  13  36      0018    156              CVTPL   #19,(SP),(R9)         ; Convert to longword
                                  001C    157  ;                                       (also clears R0)
```

```
                    0B  1D  001C  158            BVS     10$                     ; Can't fit in 32 bits
          69  89  E1 8F  78  001E  159            ASHL    #-31,(R9)+,(R9)         ; Success: spread sign bit
                    50  D6  0023  160            INCL    R0                      ; Indicate success, R0 = 1
                 5E  18  C0  0025  161 11$:       ADDL2   #24,SP                  ; Remove temp storage
                    05  0028  162            RSB                             ; Return to caller.
                        0029  163 ;+
                        0029  164 ; Come here if the packed number won't fit in 32 bits.
                        0029  165 ; Divide by 2**32 to get the high 32 bits of the quadword.
                        0029  166 ;-
              06 09 AE  E9  0029  167 10$:       BLBC    9(SP),13$               ; Skip if positive
     6E  13  D5 AF  0A  22  002D  168            SUBP4   #BIAS_DIGITS,BIAS_1,#19,(SP)  ; Make more negative
  13  6E  13  C9 AF  0A  27  0033  169 13$:       DIVP    #BIAS_DIGITS,BIAS_,#19,(SP),#19,12(SP)
                 0C AE      003A
           04 A9  65  13  36  003C  170            CVTPL   #19,(R5),4(R9)          ; Convert & store high bits (clears R0)
                    02  1D  0041  171            BVS     12$                     ; Number too large for a 64-bit integer
                    50  D6  0043  172            INCL    R0                      ; Indicate success, R0 = 1
                 5E  18  C0  0045  173 12$:       ADDL2   #24,SP                  ; Remove temp storage
                    05  0048  174            RSB                             ; Return to caller
                        0049  175
                        0049  176            .END
```

```
BIAS              00000000 R    01
BIAS_1            00000006 R    01
BIAS_DIGITS    =  0000000A      01
COB$CVTPQ_R9      0000000C 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	00000049 (	73.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT NOVEC	LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:00.95
Command processing	108	00:00:00.34	00:00:02.63
Pass 1	70	00:00:00.29	00:00:03.31
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	44	00:00:00.25	00:00:03.18
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	257	00:00:00.97	00:00:10.11

The working set limit was 900 pages.
2087 bytes (5 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 4 non-local and 4 local symbols.
176 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$:COBCVTPQ/OBJ=OBJ$:COBCVTPQ MSRC$:COBCVTPQ/UPDATE=(ENH$:COBCVTPQ)

COBCVTRPQ
LIS

COBCVTDQ
LIS

COBCANCEL
LIS

COBCVTQP
LIS

COBACCTIM
LIS

COBCVTPQ
LIS

COBCVTRFQ
LIS

COBCVTQF
LIS

COBCVTRQP
LIS

COBCALL
LIS

COBCVTFQ
LIS

COBACCEPT
LIS

COBCVTRQQ
LIS

COBCVTQD
LIS