



```

CCCCCCCC 000000 BBBB8888 CCCCCCCC VV VV TTTTTTTTTT RRRRRRRR PPPPPPPP QQQQQQ
CCCCCCCC 000000 BBBB8888 CCCCCCCC VV VV TTTTTTTTTT RRRRRRRR PPPPPPPP QQQQQQ
CC         00     00 BB         BB CC         VV VV TT         RR         RR PP         PP QQ         QQ
CC         00     00 BB         BB CC         VV VV TT         RR         RR PP         PP QQ         QQ
CC         00     00 BB         BB CC         VV VV TT         RR         RR PP         PP QQ         QQ
CC         00     00 BB         BB CC         VV VV TT         RR         RR PP         PP QQ         QQ
CC         00     00 BBB88888 BB CC         VV VV TT         RRRRRRRR PPPPPPPP QQ         QQ
CC         00     00 BBB88888 BB CC         VV VV TT         RRRRRRRR PPPPPPPP QQ         QQ
CC         00     00 BB         BB CC         VV VV TT         RR RR PP         PP QQ         QQ
CC         00     00 BB         BB CC         VV VV TT         RR RR PP         PP QQ         QQ
CC         00     00 BB         BB CC         VV VV TT         RR RR PP         PP QQ         QQ
CC         00     00 BB         BB CC         VV VV TT         RR RR PP         PP QQ         QQ
CCCCCCCC 000000 BBBB8888 CCCCCCCC VV VV TT         RR RR PP         PP QQ         QQ
CCCCCCCC 000000 BBBB8888 CCCCCCCC VV VV TT         RR RR PP         PP QQ         QQ

```

```

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

```

(2) 49  
(3) 61  
(4) 103

HISTORY ; Detailed Current Edit History  
DECLARATIONS  
COBSCVTRPQ\_R9

```
0000 1 .TITLE COBSCVTRPQ_R9 COBOL Convert Rounded Packed to Quad
0000 2 .IDENT /1-006/ ; File: COBCVTRPQ.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 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, 16-JAN-1979
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 COBSCVTRPQ
0000 53 :
0000 54 : 1-001 - Original. JBS 16-JAN-1979
0000 55 : 1-002 - Bug fixes and cleanup. MLJ 10-Mar-1979
0000 56 : 1-003 - 19 digit temps. MLJ 13-Mar-1979
0000 57 : 1-004 - Correct problem with high order longword. MLJ 22-Mar-1979
0000 58 : 1-005 - Correct round toward zero problem. PDG 12-Jul-1979
0000 59 : 1-006 - Cosmetic changes. RKR 18-OCT-79
```

```
0000 61 .SBTTL DECLARATIONS
0000 62
0000 63 :
0000 64 : INCLUDE FILES:
0000 65 :
0000 66 :
0000 67 :
0000 68 : EXTERNAL SYMBOLS:
0000 69 : NONE
0000 70 :
0000 71 :
0000 72 :
0000 73 : MACROS:
0000 74 : NONE
0000 75 :
0000 76 :
0000 77 :
0000 78 : PSECT DECLARATIONS:
00000000 79 : .PSECT _COB$CODE PIC, SHR, LONG, EXE, NOWRT
0000 80
0000 81 :
0000 82 : EQUATED SYMBOLS:
0000 83 : NONE
0000 84 :
0000 85 :
0000 86 :
0000 87 : OWN STORAGE:
0000 88 :
0000 89 :+
0000 90 : The following constant has the value 2**32. It is used for scaling
0000 91 : the high 32 bits and for compensating for unsigned arithmetic.
0000 92 :-
6C 29 67 49 29 04 0000 93 BIAS: .PACKED 4294967296 ; 2**32
0006 94 :+
0006 95 : The following constant is 2**32-1. It is subtracted from negative numbers,
0006 96 : to compensate for DIVP truncating toward zero.
0006 97 :-
5C 29 67 49 29 04 0006 98 BIAS_1: .PACKED 4294967295 ; 2**32-1
000C 99
0000000A 000C 100 BIAS_DIGITS=10
000C 101 :
```

```

000C 103      .SBTTL COB$CVTRPQ_R9
000C 104
000C 105      :++
000C 106      : FUNCTIONAL DESCRIPTION:
000C 107      :
000C 108      :     Converts packed to quadword (64-bit integer) with rounding
000C 109      :
000C 110      : CALLING SEQUENCE:
000C 111      :
000C 112      :     JSB COB$CVTRPQ_R9 (scale.rl.v, srclen.rl.v, src.rp.r, dst.wq.r)
000C 113      :
000C 114      :     Arguments are passed in R6, R7, R8 and R9.
000C 115      :
000C 116      : INPUT PARAMETERS:
000C 117      :
000C 118      :     SCALE.rl.v           The power of ten by which the internal
000C 119      :                       representation of the source must be
000C 120      :                       multiplied to scale the same as the
000C 121      :                       internal representation of the dest.
000C 122      :     SRCLEN.rl.v         The number of digits in the source
000C 123      :     SRC.rp.r           The number to be converted
000C 124      :
000C 125      : IMPLICIT INPUTS:
000C 126      :
000C 127      :     All of the trap bits in the PSL are assumed off.
000C 128      :
000C 129      : OUTPUT PARAMETERS:
000C 130      :
000C 131      :     DST.wq.r           The place to store the converted number
000C 132      :
000C 133      : IMPLICIT OUTPUTS:
000C 134      :
000C 135      :     NONE
000C 136      :
000C 137      : FUNCTION VALUE:
000C 138      :
000C 139      :     1 = SUCCESS, 0 = FAILURE
000C 140      :
000C 141      : SIDE EFFECTS:
000C 142      :
000C 143      :     Destroys registers R0 through R9.
000C 144      :
000C 145      :--
000C 146
000C 147
000C 148 COB$CVTRPQ R9::
6E 13 05 68 5E 18 C2 000C 149      SUBL2  #24,SP           ; Make room for temp storage
57 56 F8 000F 150      ASHP   R6,R7,(R8),#5,#19,(SP) ; Scale and integerize number
                                ; (also clears R0)
                                ;
                                ; If overflow, won't fit in 64 bits
                                ;
0016 151
2D 1D 0016 152      BVS    11$
0018 153      :+
0018 154      : Since quadwords often have their high 32 bits unused, try to convert
0018 155      : the packed number to a longword. If it succeeds, we need only spread
0018 156      : the sign bit. If it fails we will have more work to do.
0018 157      :-
69 6E 13 36 0018 158      CVTPL  #19,(SP),(R9)           ; Convert to longword
001C 159      ; (also clears R0)

```

```

        69  89  E1  8F  1D  001C  160      BVS      10$      ; Can't fit in 32 bits
        50  18  05  0023  161      ASHL     #-31,(R9)+,(R9) ; Success: spread sign bit
        5E  18  05  0025  162      INCL     R0          ; Indicate success, R0 = 1
        05  0028  163      ADDL2    #24,SP     ; Remove temp storage
        0029  164      RSB          ; Return to caller.
        0029  165      ;+
        0029  166      ; Come here if the packed number won't fit in 32 bits.
        0029  167      ; Divide by 2**32 to get the high 32 bits of the quadword.
        0029  168      ;-
13  6E  13  06  09  AE  E9  0029  169  10$:  BLBC     9(SP),12$   ; Skip if positive
        6E  13  D5  AF  0A  22  002D  170      SUBP4    #BIAS_DIGITS,BIAS_1,#19,(SP) ; Make more negative
        6E  13  C9  AF  0A  27  0033  171  12$:  DIVP     #BIAS_DIGITS,BIAS,#19,(SP),#19,12(SP)
        04  A9  65  0C  AE  003A
        02  1D  003C  172      CVTPL    #19,(R5),4(R9) ; Convert and store high bits
        50  18  05  0041  173      ; (also clears R0)
        5E  18  05  0041  174      BVS      11$      ; Number too large for a 64-bit integer
        05  0043  175      INCL     R0          ; Indicate success, R0 = 1
        05  0045  176  11$:  ADDL2    #24,SP     ; Remove temp storage
        05  0048  177      RSB          ; Return to caller
        0049  178      ;
        0049  179      .END

```

COB\$CVTRPQ\_R9  
Symbol table

COBOL Convert Rounded Packed to Quad I 16

15-SEP-1984 23:41:53  
6-SEP-1984 10:43:53

VAX/VMS Macro V04-00  
[COBRTL.SRC]COB\$CVTRPQ.MAR;1

Page 6  
(4)

BIAS	00000000	R	01
BIAS_1	00000006	R	01
BIAS_DIGITS =	0000000A		
COB\$CVTRPQ_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	32	00:00:00.05	00:00:00.59
Command processing	111	00:00:00.32	00:00:02.65
Pass 1	69	00:00:00.30	00:00:02.08
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	45	00:00:00.26	00:00:02.01
Symbol table output	2	00:00:00.00	00:00:00.04
Psect synopsis output	2	00:00:00.01	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	263	00:00:00.94	00:00:07.40

The working set limit was 900 pages.  
 2106 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 3 local symbols.  
 179 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
_\$255\$DUA28:[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\$:COB\$CVTRPQ/OBJ=OBJ\$:COB\$CVTRPQ MSRC\$:COB\$CVTRPQ/UPDATE=(ENH\$:COB\$CVTRPQ)

