



```

CCCCCCCC 000000 MM MM RRRRRRRR AAAAAA DDDDDDDD 5555555555 000000
CCCCCCCC 000000 MM MM RRRRRRRR AAAAAA DDDDDDDD 5555555555 000000
CC        00        00 MMMM MMMM RR RR AA AA DD DD 55 00 00
CC        00        00 MMMM MMMM RR RR AA AA DD DD 55 00 00
CC        00        00 MM MM MM RR RR AA AA DD DD 555555 00 0000
CC        00        00 MM MM MM RR RR AA AA DD DD 555555 00 0000
CC        00        00 MM MM MM RRRRRRRR AA AA DD DD 55 00 00
CC        00        00 MM MM MM RRRRRRRR AA AA DD DD 55 00 00
CC        00        00 MM MM MM RRRRRRRR AA AA DD DD 55 00 00
CC        00        00 MM MM MM RR RR AAAAAAAAAA DD DD 55 0000 00
CC        00        00 MM MM MM RR RR AAAAAAAAAA DD DD 55 0000 00
CC        00        00 MM MM MM RR RR AA AA DD DD 55 55 00 00
CC        00        00 MM MM MM RR RR AA AA DD DD 55 55 00 00
CCCCCCCC 000000 MM MM RR RR AA AA DDDDDDDD 555555 000000
CCCCCCCC 000000 MM MM RR PR AA AA DDDDDDDD 555555 000000

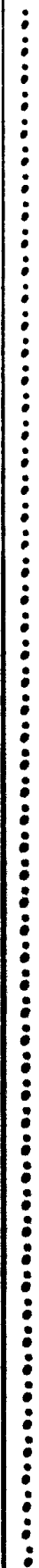
```

....  
....  
....  
....

```

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) 67  
(4) 97

HISTORY ; Detailed Current Edit History  
DECLARATIONS  
RAD50 - CONVERT 6-CHAR ASCII STRING INTO RADIX-50 STRING



```
0000 1 .TITLE COMRAD50 ; FORTRAN COMPATIBILITY - ASCII to RAD50 conversion
0000 2 .IDENT /1-004/ ; File: COMRAD50.MAR Edit: JAW1004
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 :
0000 29 : FACILITY: FORTRAN COMPATABILITY LIBRARY
0000 30 :++
0000 31 : ABSTRACT:
0000 32 :
0000 33 : FORTRAN COMPATABILITY library routine RAD50 converts 6 ASCII
0000 34 : characters to 2 RAD50 words.
0000 35 :
0000 36 :--
0000 37 :
0000 38 : VERSION: 0
0000 39 :
0000 40 : HISTORY:
0000 41 :
0000 42 : AUTHOR:
0000 43 : Peter Yuo, 12-Sep-77: Version 0
0000 44 :
0000 45 : MODIFIED BY:
0000 46 :
0000 47 :
```

```
0000 49      .SBTTL  HISTORY      ; Detailed Current Edit History
0000 50
0000 51 : Edit History for Version 01 of ASCR50
0000 52 :
0000 53 : 0-03 Clear RADIX_VALUE at initialization in R50WD_R6
0000 54 : 00-06 - Define formal for RAD50 so no access via. TNH 5-Jan-78
0000 55 : 00-07 - Make PSECT be F4PCOMPAT$CODE. TNH 5-Jan-78
0000 56 : 0-8 - Bug fix for RAD50. JMT 5-Jan-78
0000 57 : 0-9 - Another bug fix for RAD50. JMT 9-Jan-77
0000 58 : 1-1 - Break module COM$ASCR50 into 3 modules:
0000 59 :          COM$RAD50 - routine RAD50
0000 60 :          COM$IRAD50 - routine IRAD50
0000 61 :          COM$$R50WD - common ASCII to RAD50 conversion routine
0000 62 : 1-002 - Update copyright notice. JBS 16-NOV-78
0000 63 : 1-003 - Add "" to PSECT directive. JBS 21-DEC-78
0000 64 : 1-004 - Allow argument to be passed either by descriptor or by
0000 65 :          reference. JAW 13-Feb-1981
```

```
0000 67      .SBTTL  DECLARATIONS
0000 68
0000 69 :
0000 70 : INCLUDE FILES:
0000 71 :
0000 72 :
0000 73 :
0000 74 : EXTERNAL SYMBOLS:
0000 75 :
0000 76      .DSABL  GBL
0000 77      .EXTRN  COM$$R50WD_R6
0000 78
0000 79 :
0000 80 : MACROS:
0000 81 :
0000 82      $DSCDEF                ; Define descriptor symbols.
0000 83
0000 84 :
0000 85 : PSECT DECLARATIONS:
0000 86 :
00000000 87      .PSECT  _F4PCOMPAT$CODE PIC,USR,CON,REL,LCL,SHR,EXE,RD,NOWRT
0000 88
0000 89 :
0000 90 : EQUATED SYMBOLS:
0000 91 :
0000 92 :
0000 93 :
0000 94 : OWN STORAGE:
0000 95 :
```

```

0000 97          .SBTTL RAD50 - CONVERT 6-CHAR ASCII STRING INTO RADIX-50 STRING
0000 98
0000 99
0000 100 :++
0000 101 : FUNCTIONAL DESCRIPTION:
0000 102 :
0000 103 :   Algorithmic steps:
0000 104 :   1) Initialization
0000 105 :       CHARS_REM = 6
0000 106 :       NEXT_INPUT_POSITION = char_array.rbu.ra
0000 107 :   2) Call COM$SR50WD_R6 to convert one word at a time.
0000 108 :
0000 109 : CALLING SEQUENCE:
0000 110 :
0000 111 :   radix50_array.wbu.v = RAD50 (ascii_array.rbu.ra)
0000 112 :   ASCII_ARRAY = 4           ; arg list offset
0000 113 :
0000 114 :
0000 115 :
0000 116 : INPUT PARAMETERS:
0000 117 :
0000 118 :   ascii_array.rbu.ra           ; ascii string to be converted
0000 119 :
0000 120 : IMPLICIT INPUTS:
0000 121 :   NONE
0000 122 :
0000 123 : OUTPUT PARAMETERS:
0000 124 :
0000 125 :   radix50_array.wlu.v         ; output location for the result
0000 126 :                               ; of the conversion
0000 127 :
0000 128 : IMPLICIT OUTPUTS:
0000 129 :   NONE
0000 130 :
0000 131 : COMPLETION CODES:
0000 132 :   NONE
0000 133 :
0000 134 : SIDE EFFECTS:
0000 135 :   NONE
0000 136 :
0000 137 : --
007C 0000 139 .ENTRY RAD50, ^M<R2, R3, R4, R5, R6>
0002 140           ; standard call-by-reference entry
0002 141
0002 142 :
0002 143 : Initialization
0002 144 :
0002 145 :
0002 146         MOVL    #6, R5           ; CHARS_REM = 6
0005 147         MOVL    ascii_array(AP), R2 ; R2 = address of input string
0009 148           ; or descriptor
0009 149         CMPW    DSC$W_LENGTH(R2), #255 ; Is length <= 255?
000E 150         BGTRU   5$              ; If not, assume by reference.
0010 151         CMPB    DSC$B_DTYPE(R2), #DSC$K_DTYPE_T ; Is data type T?
0014 152         BNEQU   5$              ; If not, assume by reference.
0016 153         CMPB    DSC$B_CLASS(R2), #DSC$K_CLASS_S ; Is class S?
  
```

```

52  04  A2  12  001A  154          BNEQU  5$          ; If not, assume by reference.
      001C  155          MOVL   DSC$A_POINTER(R2), R2 ; Use address in descriptor.
      0020  156          :
      0020  157          :
      0020  158          : Call COM$$R50WD_R6
      0020  159          :
      0020  160          :
00000000'EF  16  0020  161  5$:   JSB    COM$$R50WD_R6          ; convert first word
      0026  162          :
      0026  163          :
      0026  164          : Save RADIX_VALUE on stack and call COM$$R50WD_R6 again
      0026  165          :
      0026  166          :
      0026  167          :
00000000'EF  51  DD  0026  167          PUSHL  R1          ; save RADIX_VALUE on stack
      16  0028  168          JSB    COM$$R50WD_R6          ; call COM$$R50WD_R6 to convert 2nd word
      002E  169          :
      002E  170          :
      002E  171          : LP(R0) = saved RADIX_VALUE on stack (1st word)
      002E  172          : HP(R0) = RADIX_VALUE (2nd word)
      002E  173          :
      002E  174          :
6E  10  10  51  F0  002E  175          INSV   R1, #16, #16, (SP) ; HP(SP) = first word converted
      50  6E  D0  0033  176          MOVL  (SP), R0      ; giving result in R0
      04  0036  177          RET    ; return with result in R0
      0037  178          :
      0037  179          :
      0037  180          .END

```



```

ASCII ARRAY = 00000004
COMSR50WD R6 ***** X 00
DSCSA_POINTER = 00000004
DSCSB_CLASS = 00000003
DSCSB_DTYPE = 00000002
DSCSK_CLASS_S = 00000001
DSCSK_DTYPE_T = 0000000E
DSCSW_LENGTH = 00000000
RAD50 00000000 RG 02
    
```

-----  
 ! Psect synopsis !  
 -----

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_F4PCOMPAT\$CODE	00000037 ( 55.)	02 ( 2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC BYTE

-----  
 ! Performance indicators !  
 -----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:01.36
Command processing	120	00:00:00.57	00:00:03.15
Pass 1	132	00:00:01.76	00:00:05.94
Symbol table sort	0	00:00:00.18	00:00:00.30
Pass 2	47	00:00:00.49	00:00:02.21
Symbol table output	3	00:00:00.02	00:00:00.16
Psect synopsis output	2	00:00:00.03	00:00:00.06
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	335	00:00:03.13	00:00:13.19

The working set limit was 1050 pages.  
 8112 bytes (16 pages) of virtual memory were used to buffer the intermediate code.  
 There were 10 pages of symbol table space allocated to hold 134 non-local and 1 local symbols.  
 180 source lines were read in Pass 1, producing 13 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\$:COMRAD50/OBJ=OBJ\$:COMRAD50 MSRC\$:COMRAD50/UPDAIE=(ENH\$:COMRAD50)

