```
FFFFFFFFFFFFFF      000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFF      000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFF      000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFF              000        000   RRR        RRR   RRR        RRR        TTT         LLL
FFF              000        000   RRR        RRR   RRR        RRR        TTT         LLL
FFF              000        000   RRR        RRR   RRR        RRR        TTT         LLL
FFF              000        000   RRR        RRR   RRR        RRR        TTT         LLL
FFF              000        000   RRR        RRR   RRR        RRR        TTT         LLL
FFFFFFFFFFF      000        000   RRRRRRRRRRRR     RRRRRRRRRRRR         TTT         LLL
FFFFFFFFFFF      000        000   RRRRRRRRRRRR     RRRRRRRRRRRR         TTT         LLL
FFFFFFFFFFF      000        000   RRRRRRRRRRRR     RRRRRRRRRRRR         TTT         LLL
FFF              000        000   RRR   RRR        RRR   RRR            TTT         LLL
FFF              000        000   RRR    RRR       RRR    RRR           TTT         LLL
FFF              000        000   RRR     RRR      RRR     RRR          TTT         LLL
FFF              000        000   RRR      RRR     RRR      RRR         TTT         LLL
FFF              000        000   RRR       RRR    RRR       RRR        TTT         LLL
FFF              000        000   RRR        RRR   RRR        RRR       TTT         LLL
FFF                000000000      RRR        RRR   RRR        RRR       TTT         LLLLLLLLLLLLLL
FFF                000000000      RRR        RRR   RRR        RRR       TTT         LLLLLLLLLLLLLL
FFF                000000000      RRR        RRR   RRR        RRR       TTT         LLLLLLLLLLLLLL
```

```
CCCCCCCC    000000    MM      MM   IIIIII   RRRRRRRR      AAAAAA    DDDDDDD    5555555555     000000
CCCCCCCC    000000    MM      MM   IIIIII   RRRRRRRR      AAAAAA    DDDDDDD    5555555555     000000
CC         00    00   MMMM  MMMM     II     RR      RR  AA      AA  DD    DD   55           00      00
CC         00    00   MMMM  MMMM     II     RR      RR  AA      AA  DD    DD   55           00      00
CC         00    00   MM  MM  MM     II     RR      RR  AA      AA  DD    DD   555555       00    0000
CC         00    00   MM  MM  MM     II     RR      RR  AA      AA  DD    DD   555555       00    0000
CC         00    00   MM      MM     II     RRRRRRRR    AA      AA  DD    DD         55     00  00  00
CC         00    00   MM      MM     II     RRRRRRRR    AA      AA  DD    DD         55     00  00  00
CC         00    00   MM      MM     II     RR  RR      AAAAAAAAAA  DD    DD         55     0000    00
CC         00    00   MM      MM     II     RR  RR      AAAAAAAAAA  DD    DD         55     0000    00
CC         00    00   MM      MM     II     RR    RR    AA      AA  DD    DD   55    55     00      00
CC         00    00   MM      MM     II     RR    RR    AA      AA  DD    DD   55    55     00      00
CCCCCCCC    000000    MM      MM   IIIIII   RR      RR  AA      AA  DDDDDDD     555555       000000
CCCCCCCC    000000    MM      MM   IIIIII   RR      RR  AA      AA  DDDDDDD     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
```

COM$IRAD50
1-004

K 15
; FORTRAN COMPATIBILITY - ASCII to RAD50 15-SEP-1984 23:47:22 VAX/VMS Macro V04-00   Page 1
                                          6-SEP-1984 10:53:06 [FORRTL.SRC]COMIRAD50.MAR;1   (1)

```
0000      1              .TITLE  COM$IRAD50    ; FORTRAN COMPATIBILITY - ASCII to RAD50 conversion
0000      2              .IDENT  /1-004/       ; File: COMIRAD50.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 IRAD50 converts a stream
0000     34      ;       of ASCII characters to RAD50 words.
0000     35      ;
0000     36      ;--
0000     37      ;
0000     38      ; VERSION: 1-002
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      ;
```

COM$IRAD50
1-004

L 15
; FORTRAN COMPATIBILITY - ASCII to RAD50 15-SEP-1984 23:47:22 VAX/VMS Macro V04-00   Page 2
HISTORY ; Detailed Current Edit History 6-SEP-1984 10:53:06 [FORRTL.SRC]COMIRAD50.MAR;1   (2)

```
              0000   49            .SBTTL  HISTORY           ; Detailed Current Edit History
              0000   50
              0000   51   ; Edit History for Version 01 of ASCR50
              0000   52   ;
              0000   53   ; 00-06 - Define formal for RAD50 so no access vio.  TNH 5-Jan-78
              0000   54   ; 00-07 - Make PSECT be F4PCOMPAT$CODE.   TNH 5-Jan-78
              0000   55   ; 0-8   - Bug fix for RAD50.  JMT 5-Jan-78
              0000   56   ; 0-9   - Another bug fix for RAD50.  JMT 9-Jan-77
              0000   57   ; 1-1   - Break module COM$ASCR50 into 3 modules:
              0000   58   ;               COM$RAD50 - routine RAD50
              0000   59   ;               COM$IRAD50 - routine IRAD50
              0000   60   ;               COM$$R50WD - common ASCII to RAD50 conversion routine
              0000   61   ; 1-002 - Update copyright notice.  JBS 16-NOV-78
              0000   62   ; 1-003 - Add " " to PSECT directive.  JBS 21-DEC-78
              0000   63   ; 1-004 - Allow second argument to be passed either by descriptor or by
              0000   64   ;               reference.  SPR 11-35539.  JAW 04-Feb-1981
```

COM$IRAD50
1-004

M 15
; FORTRAN COMPATIBILITY - ASCII to RAD50 15-SEP-1984 23:47:22 VAX/VMS Macro V04-00   Page   3
DECLARATIONS   6-SEP-1984 10:53:06 [FORRTL.SRC]COMIRAD50.MAR;1   (3)

```
        0000   66           .SBTTL  DECLARATIONS
        0000   67
        0000   68  ;
        0000   69  ; INCLUDE FILES:
        0000   70  ;
        0000   71
        0000   72
        0000   73  ; EXTERNAL SYMBOLS:
        0000   74  ;
        0000   75           .DSABL  GBL
        0000   76           .EXTRN  COM$$R50WD_R6
        0000   77
        0000   78
        0000   79  ; MACROS:
        0000   80  ;
        0000   81           $DSCDEF                          ; Define descriptor symbols.
        0000   82
        0000   83  ; PSECT DECLARATIONS:
        0000   84  ;
        0000   85  ;
    00000000   86           .PSECT  _F4PCOMPAT$CODE PIC,USR,CON,REL,LCL,SHR,EXE,RD,NOWRT
        0000   87
        0000   88
        0000   89  ; EQUATED SYMBOLS:
        0000   90  ;
        0000   91
        0000   92  ;
        0000   93  ; OWN STORAGE:
        0000   94  ;
```

N 15

COM$IRAD50   ; FORTRAN COMPATIBILITY - ASCII to RAD50 15-SEP-1984 23:47:22 VAX/VMS Macro V04-00   Page   4
1-004   IRAD50 - CONVERT HOLLERITH STRINGS TO RA 6-SEP-1984 10:53:06 [FORRTL.SRC]COMIRAD50.MAR;1   (4)

```
                        0000      96              .SBTTL  IRAD50 - CONVERT HOLLERITH STRINGS TO RADIX-50 REPRESATATION
                        0000      97
                        0000      98      ;++
                        0000      99      ; FUNCTIONAL DESCRIPTION:
                        0000     100      ;
                        0000     101      ;       Algorithmic steps:
                        0000     102      ;       1) Initialization
                        0000     103      ;          CHARS_REM = max_char_cnt.rbu.ra
                        0000     104      ;          NEXT_INPUT_POSITION = char_array.rbu.ra
                        0000     105      ;          NEXT_OUTPUT_POSITION = radix50_array.rbu.ra
                        0000     106      ;          ACTUAL_CHAR_COUNT = 0
                        0000     107      ;       2) Call COM$$R50WD_R6 to convert one word at a time.
                        0000     108      ;          If CHARS_REM =<0 then return with function_value = ACTUAL_CHAR_COUNT,
                        0000     109      ;       NOTE: Three characters of ASCII input are packed into each word
                        0000     110      ;       of output in radix-50 format. The number of output word modified
                        0000     111      ;       is computed by the expression (in integer mode) (ICNT+2)/3.
                        0000     112      ;
                        0000     113      ; CALLING SEQUENCE:
                        0000     114      ;
                        0000     115      ;       [no_char_conv.ww.v] = IRAD50 (max_char_cnt.rw.r, char_array.rbu.ra,
                        0000     116      ;                                radix50_array.wbu.ra)
                        0000     117      ;
      00000004          0000     118      ;       max_char_cnt    = 4                     ; max_char_cnt.rw.r
      00000008          0000     119      ;       char_array      = 8                     ; char_array.rbu.ra
      0000000C          0000     120      ;       radix50_array   = 12                    ; radix50_array.rbu.ra
                        0000     121      ;
                        0000     122      ;
                        0000     123      ;
                        0000     124      ; INPUT PARAMETERS:
                        0000     125      ;
                        0000     126      ;       max_char_cnt.rw.r                       ; maximum number of chars to convert
                        0000     127      ;       char_array.rbu.ra                       ; ascii string to be converted
                        0000     128      ;
                        0000     129      ; IMPLICIT INPUTS:
                        0000     130      ;       NONE
                        0000     131      ;
                        0000     132      ; OUTPUT PARAMETERS:
                        0000     133      ;
                        0000     134      ;       radix50_array.wbu.ra                    ; output location for the result
                        0000     135      ;                                               ; of the conversion
                        0000     136      ;
                        0000     137      ; IMPLICIT OUTPUTS:
                        0000     138      ;       NONE
                        0000     139      ;
                        0000     140      ; COMPLETION CODES:
                        0000     141      ;       NONE
                        0000     142      ;
                        0000     143      ; SIDE EFFECTS:
                        0000     144      ;       NONE
                        0000     145      ;
                        0000     146      ;--
                        0000     147
                        0000     148
                        0000     149
      007C              0000     150              .ENTRY  IRAD50, ^M<R2, R3, R4, R5, R6>
                        0002     151                                                      ; standard call-by-reference entry
                        0002     152
```

COM$IRAD50
1-004

B 16
; FORTRAN COMPATIBILITY - ASCII to RAD50 15-SEP-1984 23:47:22 VAX/VMS Macro V04-00   Page   5
  IRAD50 - CONVERT HOLLERITH STRINGS TO RA 6-SEP-1984 10:53:06 [FORRTL.SRC]COMIRAD50.MAR;1   (4)

```
                          0002    153  ;
                          0002    154  ; Initialization
                          0002    155  ;
                          0002    156
      55   04 BC   3C     0002    157          MOVZWL   @max_char_cnt(AP), R5    ; R5 = maximum number of chars
                          0006    158                                           ; to be converted
      52   08 AC   D0     0006    159          MOVL     char_array(AP), R2       ; R2 = address of input string
                          000A    160                                           ;   or descriptor
  00FF 8F   62   B1       000A    161          CMPW     DSC$W_LENGTH(R2), #255   ; Is length <= 255?
           10   1A        000F    162          BGTRU    5$                       ; If not, assume by reference.
      0E   02 A2   91     0011    163          CMPB     DSC$B_DTYPE(R2), #DSC$K_DTYPE_T ; Is data type T?
           0A   12        0015    164          BNEQU    5$                       ; If not, assume by reference.
      01   03 A2   91     0017    165          CMPB     DSC$B_CLASS(R2), #DSC$K_CLASS_S ; Is class S?
           04   12        001B    166          BNEQU    5$                       ; If not, assume by reference.
      52   04 A2   D0     001D    167          MOVL     DSC$A_POINTER(R2), R2    ; Use address in descriptor.
      54   0C AC   D0     0021    168  5$:     MOVL     radix50_array(AP), R4    ; R4 = address of the output location
           50   D4        0025    169          CLRL     R0                       ; R0 = ACTUAL_CHAR_COUNT = 0
                          0027    170
                          0027    171  ;
                          0027    172  ; If CHARS_REM =< 0 then return with function_value equal to ACTUAL_CHAR_COUNT
                          0027    173  ; else call R50WD_R5 to convert one word at a time.
                          0027    174  ;
                          0027    175
                          0027    176  10$:
  00000000'EF   16        0027    177          JSB      COM$$R50WD_R6            ; convert one word at a time
           84   51   B0   002D    178          MOVW     R1, (R4)+               ; output one word at a time
                55   D5   0030    179          TSTL     R5                      ; any more?
                F3   14   0032    180          BGTR     10$                     ; branch if so
                     04   0034    181          RET                             ; return with R0 = ACTUAL_CHAR_COUNT
                          0035    182
                          0035    183
                          0035    184          .END
```

C 16

COM$IRAD50   ; FORTRAN COMPATIBILITY - ASCII to RAD50 15-SEP-1984 23:47:22   VAX/VMS Macro V04-00   Page   6
Symbol table   6-SEP-1984 10:53:06   [FORRTL.SRC]COMIRAD50.MAR;1   (4)

```
CHAR_ARRAY                  = 00000008
COM$$R50WD_R6               ********    X    00
DSC$A_POINTER               = 00000004
DSC$B_CLASS                 = 00000003
DSC$B_DTYPE                 = 00000002
DSC$K_CLASS_S               = 00000001
DSC$K_DTYPE_T               = 0000000E
DSC$W_LENGTH                = 00000000
IRAD50                        00000000  RG    02
MAX_CHAR_CNT                = 00000004
RADIX50_ARRAY               = 0000000C
```

```
                            +-------------------+
                            ! 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
_F4PCOMPAT$CODE               00000035  (     53.)  02 (  2.)  PIC     USR  CON  REL  LCL  SHR    EXE    RD     NOWRT  NOVEC  BYTE
```

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

Phase   Page faults   CPU Time   Elapsed Time
-----   -----------   --------   ------------
```
Initialization              29          00:00:00.12     00:00:00.72
Command processing         104          00:00:00.56     00:00:02.20
Pass 1                     137          00:00:01.76     00:00:06.94
Symbol table sort            0          00:00:00.16     00:00:00.27
Pass 2                      46          00:00:00.52     00:00:01.86
Symbol table output          3          00:00:00.02     00:00:00.02
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       323          00:00:03.17     00:00:12.07
```

The working set limit was 1050 pages.
8165 bytes (16 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 136 non-local and 2 local symbols.
184 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$:COMIRAD50/OBJ=OBJ$:COMIRAD50 MSRC$:COMIRAD50/UPDATE=(ENH$:COMIRAD50)

COMASSIGN
LIS

FORIOSDEF
SDL

FORFMT
REQ

FOROPN
REQ

COMR50ASC
LIS

COMERRTST
LIS

FDLTRFVEC
LIS

FORPAR
SDL

FORLIB
REQ

COMERRSET
LIS

COMIRAD50
LIS

FORPROLOG
REQ

FORRTL

FORRCE
SDL

FORMACROS
REQ

FORRTL
MAP

COMFDBSET
LIS

FORERR
SDL

FORDEF
FOR

COMEST
REQ

FORNML
REQ

COMCLOSE
LIS