```
; FILE    Macsbug-print.text            ; print utilities for Macsbug
;
;         Modification history
;
;          6-Sep-84 New Today
;          8-Sep-84 Added Print(N)Hex routines, moved Bin2Char routine from Init
;         16-Sep-84 Moved MEight, MError from Init to here
;         14-Nov-84 Move register printing code from Init to here

;-------------------------------------------------------------------
; PntPascal -- A0 = Ptr to Pascal string, print it out. Trashes
; D0, max string is 63
;-------------------------------------------------------------------

PntPascal
          BSR.S     @1                  ; stuff leading "
          MOVEQ     #0,D0               ; clear D0
          MOVE.B    (A0)+,D0            ; get the length byte
          AND.B     #$3F,D0             ; constrain length to < 64

@0        MOVE.B    (A0)+,(A6)+         ; push to buffer
          DBRA      D0,@0               ; keep looping

@1        MOVE.B    #'"',(A6)+          ; and finish off with "
          RTS

;-------------------------------------------------------------------
; Display utilities -- print out hex digits. D0 = input value,
; trashes D1/D2
;-------------------------------------------------------------------

PNTZHX                                            ; signed display
            TST.L       D0                        ; test number
            BPL.S       @0
            NEG.L       D0
            MOVE.B      #$2D,(A6)+                ; stuff a '-'
@0

PNT8HX
            SWAP        D0
            BSR.S       PNT4HX
            SWAP        D0
            BRA.S       PNT4HX
PNT6HX
            SWAP        D0
            BSR.S       PNT2HX
            SWAP        D0
PNT4HX
            MOVE.W      D0,D1
            ROXR.W      #8,D0
            BSR.S       PNT2HX
            MOVE.W      D1,D0
PNT2HX
            MOVE.W      D0,D2
            ROR.W       #4,D0
            BSR.S       PUTHEX
            MOVE.W      D2,D0
PUTHEX
            ANDI.B      #$0F,D0
            ORI.B       #$30,D0
            CMPI.B      #$39,D0
            BLE.S       @0
            ADDQ.W     #7,D0
@0
            MOVE.B     D0,(A6)+
            RTS

; The following routines have a hex value to print in D0, and a location to
; put the string in A1. Trashes A0-A2/D0-D2

SetupIOReg
          MOVE.L    (SP)+,A2                ; save call address
          MOVE.L    A6,-(SP)                ; save old io buffer pointer
          MOVE.L    A1,A6                   ; set A6 (current IO input location) to new
value
          JSR       (A2)                    ; call the IO routine
          MOVE.L    A6,A1                   ; bump A1 to new value
          MOVE.L    (SP)+,A6                ; restore A6
          RTS                               ; and return

Print8Hex
          PEA       PNT8HX
          BRA.S     SetupIOReg

Print6Hex
          PEA       PNT6HX
          BRA.S     SetupIOReg

Print4Hex
          PEA       PNT4HX
          BRA.S     SetupIOReg

Print2Hex
          PEA       PNT2HX
          BRA.S     SetupIOReg

Print1Hex
          PEA       PUTHEX
          BRA.S     SetupIOReg

;--------------------------------------------------------------------------------------
--
; Routine Name      Bin2Char
;
; Registers         D0.B (input)            : value to print out as ascii
;
; Function          Convert D0.B to an ascii character and stuff into IO buffer
;--------------------------------------------------------------------------------------
--

Bin2Char
          CMPI.B    #' ',D0
          BGE.S     @0
          MOVE.B    #'.',D0                 ; jam a period
@0        MOVE.B    D0,(A6)+
          RTS

;--------------------------------------------------------------------------------------
--
; Routine Name      MError
;
; Registers         D0 (input)              ; word offset from MText of text string
;                   A0 (trash)
;
```

```
; Function            Print out four characters indexed D0 bytes from Mtext, followed by
;                     ' ERR'
;--------------------------------------------------------------------------------------
--

MERROR
            BSR         FixBuf
            BSR.S       MFOUR
            MOVE.L      #' ERR',(A6)+
            RTS

MFOUR
            LEA         Mtext,A0
            ADD         D0,A0
            MOVE.L      (A0),(A6)+
            RTS

;--------------------------------------------------------------------------------------
--
; Routine Name        MEight
;
; Registers           D0 (input)          ; word offset from MText of text string
;                     A0 (trash)
;
; Function            Print out the eight characters indexed D0 bytes from Mtext
;--------------------------------------------------------------------------------------
--

MEIGHT
            BSR         FixBuf
            LEA         Mtext,A0
            ADD         D0,A0
            MOVE.L      (A0)+,(A6)+
            MOVE.L      (A0),(A6)+
            RTS


;--------------------------------------------------------------------------------------
--
; Routine Name        TabIt
;
; Registers           D0.W (input)        ; position to tab out to (w/1 = 1st char
position
;                     A0 (trash)
;
; Function            Print spaces at A6 until out to tab position
;--------------------------------------------------------------------------------------
--

TabIt
          MOVE.L    A5,A0                 ; get initial start position
          SUBQ.B    #1,D0                 ; adjust position
          ADD.W     D0,A0                 ; calc new position

@0        CMP.L     A6,A0                 ; are we at or past the end position?
          BGT.S     @1                    ; no, end pos > lO pos

          RTS                             ; return

@1        MOVE.B    #' ',(A6)+            ; stuff a space
          BRA.S     @0                    ; and loop
```

```
; Print reg groups utilities

PNTCLS
            BSR         FIXBUF
            CLR.L       D6
PNTCLS1
            BSR.S       PNTREG
            CMPI.B      #4,D6
            BNE.S       PNTCLS2
            BSR         WriteLine
            BRA.S       PNTCLS1
PNTCLS2
            CMPI.B      #8,D6
            BNE.S       PNTCLS1
            BSR         WriteLine
            RTS
PNTREG
            MOVE.B      D7,(A6)+
            MOVE.B      D6,D0
            BSR         PUTHEX
            MOVE.B      #'=',(A6)+
            MOVE.L      D6,D0
            LSL.L       #2,D0
            ADD.L       A3,D0
            MOVE.L      D0,A4
PNTREG1
            MOVE.L      (A4),D0
            BSR         PNT8HX
            MOVE.B      #' ',(A6)+
            ADDQ        #1,D6
            RTS

PrintR1
            BSR         FIXBUF
            ADDQ        #2,A6                   ; skip the reg char
            MOVE.B      #'=',(A6)+
            BRA.S       pntreg1
```