L 12

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

_$

Ps
--

YZ

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

```
**FILE**ID**IPCONTROL                                         D 11




            IIIIII    PPPPPPPP      CCCCCCCC    OOOOOO    NN      NN  TTTTTTTTTT  RRRRRRRR      OOOOOO    LL
            IIIIII    PPPPPPPP      CCCCCCCC    OOOOOO    NN      NN  TTTTTTTTTT  RRRRRRRR      OOOOOO    LL
              II      PP      PP  CC          OO      OO  NN      NN      TT      RR      RR  OO      OO  LL
              II      PP      PP  CC          OO      OO  NN      NN      TT      RR      RR  OO      OO  LL
              II      PP      PP  CC          OO      OO  NNNN    NN      TT      RR      RR  OO      OO  LL
              II      PP      PP  CC          OO      OO  NNNN    NN      TT      RR      RR  OO      OO  LL
              II      PPPPPPPP    CC          OO      OO  NN  NN  NN      TT      RRRRRRRR    OO      OO  LL
              II      PPPPPPPP    CC          OO      OO  NN  NN  NN      TT      RRRRRRRR    OO      OO  LL
              II      PP          CC          OO      OO  NN    NNNN      TT      RR  RR      OO      OO  LL
              II      PP          CC          OO      OO  NN    NNNN      TT      RR  RR      OO      OO  LL
              II      PP          CC          OO      OO  NN      NN      TT      RR    RR    OO      OO  LL               ....
              II      PP          CC          OO      OO  NN      NN      TT      RR    RR    OO      OO  LL               ....
            IIIIII    PP            CCCCCCCC    OOOOOO    NN      NN      TT      RR      RR    OOOOOO    LLLLLLLLLL       ....
            IIIIII    PP            CCCCCCCC    OOOOOO    NN      NN      TT      RR      RR    OOOOOO    LLLLLLLLLL       ....

          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
```

Table of contents

```
0000            .TITLE  IPCONTROL - IPL 12 INTERRUPT PROCESSOR
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ;++
0000    ;
0000    ; FACILITY:
0000    ;
0000    ;       NONPAGED VAX/VMS EXEC
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;       THIS MODULE CONTAINS THE ROUTINE NECESSARY TO PROCESS THE IPL 12
0000    ;       INTERRUPTS.  THESE ARE GENERATED MANUALLY FROM THE CONSOLE AS A
0000    ;       LAST DITCH EFFORT TO CORRECT SOME SYSTEM MALADY BEFORE REBOOTING
0000    ;       SYSTEM.
0000    ;
0000    ; AUTHOR:
0000    ;
0000    ;       L. MARK PILANT  28-JAN-1982
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-010 LMP0294         L. Mark Pilant,         2-Aug-1984  15:54
0000    ;               Add the dollar sign to the list of legal characters.
0000    ;
0000    ;       V03-009 PRB0333         Paul Beck       3-May-1984  22:26
0000    ;               Fix fork block initialization for Q command.
0000    ;
0000    ;       V03-008 ACG0416         Andrew C. Goldstein,    18-Apr-1984  9:51
0000    ;               Use common I/O search routines to support cluster devices
0000    ;
0000    ;       V03-007 PRB0328         Paul Beck       9-Apr-1984  15:59
0000    ;               Add retry logic for call to CNX$CHANGE_QUORUM.
```

```
0000     ;
0000     ;       V03-006 TCM0003         Trudy C. Matthews       09-Apr-1984
0000     ;               Yet another change in RXCS/TXCS handling: use routine
0000     ;               CON$RELEASECTY to restore the state of the registers.
0000     ;
0000     ;       V03-005 PRB0325         Paul Beck       25-Mar-1984  15:59
0000     ;               Include "Q" command to recalculate cluster quorum
0000     ;
0000     ;       V03-004 TCM0002         Trudy C. Matthews       13-Dec-1983
0000     ;               New interface to CON$OWNCTY: it now returns the value to
0000     ;               restore to TXCS in R0 and the value to restore to RXCS
0000     ;               in R1.
0000     ;
0000     ;       V03-003 TCM0001         Trudy C. Matthews       16-Feb-1983
0000     ;               Set up to talk to console terminal in a CPU-dependent
0000     ;               fashion.
0000     ;
0000     ;       V03-002 LMP0024         L. Mark Pilant          23-Apr-1982  13:45
0000     ;               Correct a problem introduced by some bit shaving.
0000     ;
0000     ;       V03-001 LMP0020         L. Mark Pilant           2-Apr-1982  15:05
0000     ;               Modify the logic so that only a device that has mount
0000     ;               verification in progess can be disabled.
0000     ;
0C᠎ᴗ0     ;       V02-001 LMP0010         L. Mark Pilant           9-Feb-1982   9:00
0000     ;               Strip parity and other garbage from incoming characters.
0000     ;
0000     ;--
```

```
0000     87           .SBTTL  USEFUL MACRO DEFINITIONS
0000     88
0000     89           .MACRO  RJMP     ADDR
0000     90           JMP     ADDR                        ;CALLED ROUTINE DOES THE RSB
0000     91           .ENDM   RJMP
0000     92
0000     93           .MACRO  CON_MSG MESSAGE,RETURN
0000     94           MOVAB   W^MESSAGE,R2                ;SET MESSAGE ADDRESS
0000     95 .IF       B,      RETURN
0000     96           JSB     (R10)                       ;SEND THE MESSAGE
0000     97 .IFF
0000     98           RJMP    (R10)                       ;SEND THE MESSAGE
0000     99 .ENDC;    B,      RETURN
0000    100           .ENDM   CON_MSG
```

```
          0000          .SBTTL  SYMBOL DEFINITIONS
          0000
          0000  ; MACRO LIBRARY CALLS
          0000
          0000          $CLUBDEF
          0000          $CSBDEF
          0000          $DCDEF
          0000          $DDBDEF
          0000          $DEVDEF
          0000          $DYNDEF
          0000          $FKBDEF
          0000          $IOCDEF
          0000          $IPLDEF
          0000          $IRPDEF
          0000          $PRDEF
          0000          $UCBDEF
          0000          $VCBDEF
          0000
          0000
          0000  ; SPECIAL TERMINAL INPUT CHARACTERS
          0000
00000007  0000          IPC_C_BELL=     7                  ;BELL (FOR FULL BUFFER & ILLEGAL CHARS)
0000000A  0000          IPC_C_LINFEED=  10                 ;LINE-FEED (ECHOED AFTER <CR>)
0000000D  0000          IPC_C_CARRETN=  13                 ;CARRIAGE-RETURN
00000015  0000          IPC_C_CNTRLU=   21                 ;CONTROL-U (IGNORE THE LINE)
0000001A  0000          IPC_C_CNTRLZ=   26                 ;CONTROL-Z (EXIT)
00000020  0000          IPC_C_SPACE=    32                 ;SPACE
0000003A  0000          IPC_C_COLON=    58                 ;DEVICE NAME TERMINATOR
00000043  0000          IPC_C_CHAR_C=   67                 ;CHARACTER C (CANCEL MOUNT VERIFICATION)
00000058  0000          IPC_C_CHAR_X=   88                 ;CHARACTER X (ENTER XDELTA)
0000005F  0000          IPC_C_UNDER=    95                 ;UNDERSCORE
00000061  0000          IPC_C_LOWERA=   97                 ;LOWER CASE A
0000007F  0000          IPC_C_DELETE=   127                ;DELETE A CHARACTER (RUBOUT)
00000051  0000          IPC_C_CHAR_Q=   81                 ;CHARACTER Q (SET CLUSTER QUORUM)
          0000
          0000  ; MISCELLANEOUS CONSTANTS
          0000
0000000B  0000          IPC_C_MAXLINSIZ=        11         ;MAXIMUM INPUT LINE SIZE
          0000                                             ; (THIS INCLUDE STORAGE FOR THE <CR>)
          0000
          0000  ; LOCAL STORAGE OFFSETS
          0000
          0000          $DEFINI IPC
          0000
          0000          $DEF    IPC_C_START
          0000          $DEF    IPC_W_INPSIZE   .BLKW   1       ;INPUT LINE SIZE IN BYTES
          0002          $DEF    IPC_T_INPLINE   .BLKB   IPC_C_MAXLINSIZ ;STORAGE FOR THE TEXT
          000D          $DEF    IPC_Q_DEVICE    .BLKQ   1       ;DEVICE DESCRIPTOR
          0015          $DEF    IPC_B_FLAGS     .BLKB   1       ;USEFUL FLAGS
          0016          _VIELD  IPC,0,<-                ;DEFINE FLAG BITS
          0016                  <DELETE,,M>,-           ;DELETE CHARACTER SEEN
          0016                  <NEWLINE,,M>,-          ;STARTING A NEW LINE
          0016                  <CNTRLZ,,M>,-           ;CONTROL-Z (EXIT) SEEN
          0016                  >
          0016          $DEF    IPC_L_OLDGETNXT .BLKL   1       ;PREVIOUS CHAR INPUT ROUTINE
          001A
          001A          $DEF    IPC_L_SAVRXCS   .BLKL   1       ;SAVE ORIGINAL RXCS
```

```
    001E   159         $DEF    IPC_L_SA TXCS   .BLKL   1          ;SAVE ORIGINAL TXCS
    0022   160         $DEF    IPC_B_P         .BLKB   4-<<.-IPC_C_START>&7>   ;LONGWORD GRANULARIT
    0024   161         $DEF    IPC_C_LENGTH            ;LENGTH OF LOCAL STORAGE
    0024   162
    0024   163         $DEFEND IPC
```

```
IP
Ps

PS
--
.
$A
WM
WI

Ph
--
!n
Co
Pa
Sy
Pa
Sy
Ps
Cr
As

Th
81
Th
56
27

Ma
--
-$
-$
TO

16

Th

MA
```

```
                                       0000          .SBTTL  ERROR MESSAGES AND MISC STRINGS
                                       0000
                                   00000000          .PSECT  WMOUNTVERMSG
                                       0000
                                       0000  IPC_ILLCOMMAND:
6D 6D 6F 63 20 6C 61 67 65 6C 6C 49    0000          .ASCIZ  \Illegal command\<13><10>
                   00 0A 0D 64 6E 61   000C
                                       0012  IPC_ILLDEVICE:
69 76 65 64 20 6C 61 67 65 6C 6C 49    0012          .ASCIZ  \Illegal device\<13><10>
                      00 0A 0D 65 63   001E
                                       0023  IPC_NOMNTVER:
69 66 69 72 65 76 20 74 6E 75 6F 4D    0023          .ASCIZ  \Mount verification not in progress\<13><10>
69 20 74 6F 6E 20 6E 6F 69 74 61 63    002F
0A 0D 73 73 65 72 67 6F 72 70 20 6E    003B
                                  00   0047
                                       0048  IPC_NOCLUSTER:
20 6E 69 20 74 6F 6E 20 65 64 6F 4E    0048          .ASCIZ  \Node not in VAXcluster\<13><10>
0A 0D 72 65 74 73 75 6C 63 58 41 56    0054
                                  00   0060
                                       0061
                                       0061  IPC_CRLFSTR:
                            00 0A 0D   0061          .BYTE   IPC_C_CARRETN,IPC_C_LINFEED,0   ;<CR>,<LF>,0
                                       0064
                                       0064  IPC_PROMPT:
                   00 20 3E 43 50 49   0064          .ASCIZ  \IPC> \
                                       006A
                                       006A          .SBTTL  CHARACTER VALIDITY TABLES
                                       006A
                                       006A  IPC_V_LEGAL:
                            04202000   006A          .LONG   ^X04202000,-            ;<CR>,<^U>,<^Z>
                                       006E                  ^X07FF0011,-            ;<SP>,<$>,<0-9>,<:>
                            07FF0011   006E                  ^X87FFFFFE,-            ;<A-Z>,<_>
                   87FFFFFE 87FFFFFE   0072                  ^X87FFFFFE              ;<a-z>,<RUB>
                                       007A  IPC_V_ARROW:
00000000 00000000 00000000 FFFFDB7F    007A          .LONG   ^XFFFFDB7F,0,0,0        ;<BEL>,<LF>,<CR>
                                       008A
```

```
                         008A           .SBTTL  COMMAND INTERPRETER AND DISPATCHER
                         0C^A
                         008A    ;++
                         008A    ; EXE$IPCONTROL - IPL 12 (DECIMAL) INTERRUPT HANDLER
                         008A    ;
                         008A    ; THIS MODULE IS CALLED WHEN AN IPL 12 INTERRUPT OCCURS.  THIS IS NORMALLY
                         008A    ; INVOKED FROM THE CONSOLE WHEN THE SYSTEM IS IN SOME DEAD LOCK CONDITION.
                         008A    ;
                         008A    ; INPUTS:
                         008A    ;
                         008A    ;       AN INPUT COMMAND LINE IS ACCEPTED FROM THE CONSOLE FOR ACTION.
                         008A    ;
                         008A    ; OUTPUTS:
                         008A    ;
                         008A    ;       THE APPROPRIATE ACTION IS TAKEN BASED UPON THE CONSOLE INPUT.
                         008A    ;       THIS IS:
                         008A    ;
                         008A    ;           1)  CLEAR MOUNT VERIFICATION ON A SPECIFIED DEVICE
                         008A    ;           2)  TRANSFER CONTROL TO XDELTA
                         008A    ;           3)  REDUCE CLUSTER QUORUM
                         008A    ;           4)  EXIT (DISMISS THE IPL 12 INTERRUPT)
                         008A    ;
                         008A    ;--
                         008A
                     00000000           .PSECT  WIONONPAGED,LONG
                         0000
                         0000    EXE$IPCONTROL::
     0FFF 8F       BB    0000           PUSHR   #^M<R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>      ;SAVE ALL REGS
  5E    DC AE      9E    0004           MOVAB   -IPC_C_LENGTH(SP),SP    ;MAKE ROOM FOR LOCAL STORAGE
     5B    5E      D0    0008           MOVL    SP,RT1                  ;SAVE ADDRESS OF LOCAL STORAGE BLOCK
5A   01E0'CF       9E    000B           MOVAB   W^IPC_OUTPUTZ,R10       ;SET ADDRESS OF .ASCIZ OUTPUT ROUTINE
 00000000'EF       16    0010    10$:   JSB     CON$OWNCTY              ;SET UP CONSOLE TERMINAL REGISTERS
  1E AB    50      D0    0016           MOVL    R0,IPC_L_SAVTXCS(R11)   ;VALUE TO RESTORE TO TXCS
  1A AB    51      D0    001A           MOVL    R1,IPC_L_SAVRXCS(R11)   ;VALUE TO RESTORE TO RXCS
     56    6B      9E    001E           MOVAB   IPC_W_INPSIZE(R11),R6   ;SET ADDRESS OF INPUT STRING SIZE
  57    02 AB      9E    0021           MOVAB   IPC_T_INPLINE(R11),R7   ;SET ADDRESS OF INPUT COMMAND BUFFER
  58    15 AB      9E    0025           MOVAB   IPC_B_FLAGS(R11),R8     ;SET ADDRESS OF THE FLAG BYTE
                         0029           CON_MSG IPC_PROMPT              ;ISSUE A PROMPT
       00DF        30    0030           BSBW    IPC_GETLINE             ;GET A COMMAND LINE FROM THE CONSOLE
  0B 68    02      E1    0033           BBC     #IPC_V_CNTRLZ,(R8),15$  ;XFER IF NO CONTROL-Z SEEN
  5E    24 AE      9E    0037           MOVAB   IPC_C_LENGTH(SP),SP     ;ELSE CLEAN THE STACK
           50      10    003B           BSBB    IPC_ENABLE              ;GO RE-ENABLE THE CONSOLE
     0FFF 8F       BA    003D           POPR    #^M<R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>      ;RESTORE REGS
           02            0041           REI                             ;DISMISS THE INTERRUPT
           66      B5    0042    15$:   TSTW    (R6)                    ;ANYTHING TYPED?
           CA      13    0044           BEQL    10$                     ;XFER IF NOT...RE-ISSUE THE PROMPT
  43 8F    67      91    0046    20$:   CMPB    (R7),#IPC_C_CHAR_C      ;ELSE CHECK FOR CANCEL
           04      12    004A           BNEQ    30$                     ;XFER IF NOT
           52      10    004C           BSBB    IPC_CANCEL              ;ELSE GO CANCEL MOUNT VERI, CATION
           C0      11    004E           BRB     10$                     ;TRY FOR ANOTHER COMMAND
  58 8F    67      91    0050    30$:   CMPB    (R7),#IPC_C_CHAR_X      ;XFER TO XDELTA?
           07      12    0054           BNEQ    40$                     ;XFER 'F NOT
           35      10    0056           BSBB    IPC_ENABLE              ;GO RE-ENABLF THE CONSOLE
       FFA5'       30    0058           BSBW    W^INI$BRK               ;ELSE CAUSE XDELTA BREAKPOINT
           B3      11    005B           BRB     10$                     ;TRY FOR ANOTHER COMMAND
  51 8F    67      91    005D    40$:   CMPB    (R7),#IPC_C_CHAR_Q      ;REDUCE CLUSTER QUORUM?
           20      12    0061           BNEQ    50$                     ;XFER IF NOT
     50  00000000'GF  D0  0063              MOVL    G^CLU$GL_CLUB,R0         ;ARE WE IN A CLUSTER?
                  09  13  006A              BEQL    42$                      ;IF NEQ, YES
           50  10 A0  D0  006C              MOVL    CLUB$L_LOCAL_CSB(R0),R0 ;GET LOCAL NODE'S STATUS
        09 60 A0  01  E0  0070              BBS     #CSB$V_MEMBER,CSB$L_STATUS(R0),45$ ;MAKE SURE WE'RE ACTIVE
                          0075      42$:    CON_MSG IPC_NOCLUSTER            ;REPORT ERROR: NOT IN CLUSTER
                  92  11  007C              BRB     10$                      ;TRY FOR ANOTHER COMMAND
                016D  30  007E      45$:    BSBW    IPC_FORK_QUORUM          ;FORK DOWN TO SYNCH TO REDUCE QUORUM
                  8D  11  0081              BRB     10$                      ;TRY FOR ANOTHER COMMAND
                          0083
                          0083      ; ILLEGAL COMMAND GIVEN
                          0083
                          0083      50$:    CON_MSG IPC_ILLCOMMAND           ;ISSUE AN ERROR MESSAGE
                FF83  31  008A              BRW     10$                      ;GO TRY FOR ANOTHER
                          008D
                          008D      ; LOCAL ROUTINE TO RE-ENABLE THE CONSOLE FOR INTERRUPTS
                          008D
                          008D      IPC_ENABLE:
                  03  BB  008D              PUSHR   #^M<R0,R1>               ;SAVE REGISTERS
           50  1E AB  D0  008F              MOVL    IPC_L_SAVTXCS(R11),R0    ;TO BE RESTORED TO TXCS
           51  1A AB  D0  0093              MOVL    IPC_L_SAVRXCS(R11),R1    ;TO BE RESTORED TO RXCS
         00000000'EF  16  0097              JSB     CON$RELEASECTY           ;RESTORE CONSOLE CTY STATE
                  03  BA  009D              POPR    #^M<R0,R1>               ;RESTORE REGISTERS
                  05  009F                  RSB                              ;RETURN TO CALLER
```

```
                              00A0         .SBTTL  CANCEL A PENDING MOUNT VERIFICATION
                              00A0
                              00A0   ;++
                              00A0   ; IPC_CANCEL - DISABLE MOUNT VERIFICATION IF IN PROGRESS
                              00A0   ;
                              00A0   ; THIS ROUTINE DISABLES MOUNT VERIFICATION IF IT IS IN PROGRESS.  THIS IS
                              00A0   ; DEFINED AS BEING IN THE STATE IN WHICH MOUNT VERIFICATION IS ENABLED
                              00A0   ; AND EITHER THE DEVICE UCB HAS THE MOUNT VERIFICATION IN PROGRESS STATUS
                              00A0   ; BIT SET OR THE IRP AT THE HEAD OF THE DEVICE IRP QUEUE HAS THE MOUNT
                              00A0   ; VERIFICATION STATUS BIT SET.
                              00A0   ;
                              00A0   ; INPUTS:
                              00A0   ;
                              00A0   ;       R7 - ADDRESS OF THE CONSOLE INPUT LINE BUFFER
                              00A0   ;
                              00A0   ; OUTPUTS:
                              00A0   ;
                              00A0   ;           1)  THE MOUNT VERFICATION BIT IS CLEARED, OR
                              00A0   ;           2)  A WARNING MESSAGE IS ISSUED
                              00A0   ;
                              00A0   ;--
                              00A0
                              00A0   IPC_CANCEL:
              50    57   D0   00A0           MOVL    R7,R0                   ;SET ADDRESS OF COMMAND LINE
                    80   95   00A3           TSTB    (R0)+                   ;SKIP OVER COMMAND CHARACTER
              20    80   91   00A5           CMPB    (R0)+,#^A/ /            ;CHECK FOR CORRECT SEPARATOR
                    07   13   00A8           BEQL    10$                     ;XFER IF CORRECT
                              00AA           CON_MSG IPC_ILLCOMMAND,RETURN   ;ISSUE MESSAGE AND RETURN
           5F 8F    80   91   00B1   10$:    CMPB    (R0)+,#IPC_C_UNDER      ;ELSE CHECK FOR AN UNDERSCORE
                    FA   13   00B5           BEQL    10$                     ;XFER IF THERE IS ONE
                    50   D7   00B7           DECL    R0                      ;ELSE BACK UP OVER NON-UNDERSCORE
              0D AB      D4   00B9           CLRL    IPC_Q_DEVICE(R11)       ;RESET DEVICE NAME SIZE
           11 AB    50   D0   00BC           MOVL    R0,IPC_Q_DEVICE+4(R11)  ;SET STARTING ADDRESS OF DEVICE NAME
              0D    60   91   00C0   20$:    CMPB    (R0),#IPC_C_CARRETN     ;END OF THE STRING?
                    0A   13   00C3           BEQL    30$                     ;XFER IF SO
              3A    80   91   00C5           CMPB    (R0)+,#IPC_C_COLON      ;ALTERNATE TERMINATOR?
                    05   13   00C8           BEQL    30$                     ;XFER IF SO
              0D AB      B6   00CA           INCW    IPC_Q_DEVICE(R11)       ;ELSE COUNT ONE MORE CHARACTER
                    F1   11   00CD           BRB     20$                     ;GO TRY THE NEXT CHARACTER
                              00CF
                              00CF   ; FIND GENERIC DEVICE NAME (WITH CONTROLLER) AND UNIT NUMBER (IN BINARY)
                              00CF
              7E    5A   7D   00CF   30$:    MOVQ    R10,-(SP)               ;SAVE R10 & R11 THROUGH SEARCH CALL
        58    0D AB      7D   00D2           MOVQ    IPC_Q_DEVICE(R11),R8    ;GET DEVICE NAME DESCRIPTOR
5A   00000000 00000041 8F   7D   00D6        MOVQ    #IOC$M_PHY!IOC$M_ANY,R10 ;PHYSICAL SEARCH, NO AVAILABILITY CHECKS
                              00E1                                           ;LKSB (R11) = 0
                  FF1C'  30   00E1           BSBW    IOC$PARSDEVNAM          ;PARSE THE SUPPLIED DEVICE NAME
                  03 50  E9   00E4           BLBC    R0,40$                  ;BRANCH OUT ON ERROR
                  FF16'  30   00E7           BSBW    IOC$SEARCHINT           ;SEARCH THE I/O DATABASE
              5A    8E   7D   00EA   40$:    MOVQ    (SP)+,R10               ;RESTORE R10 & R11
                 14 50   E9   00ED           BLBC    R0,70$                  ;OUT ON ERROR
        01     40 A5     91   00F0           CMPB    UCB$B_DEVCLASS(R5),#DC$_DISK    ;IS DEVICE A DISK?
                    0E   12   00F4           BNEQ    70$                     ;BRANCH IF NOT
     10 64 A5       0E   E1   00F6   50$:    BBC     #UCB$V_MNTVERIP,UCB$W_STS(R5),80$ ;BRANCH IF NOT IN PROGRESS
        52     34 A5     D0   00FB   60$:    MOVL    UCB$L_VCB(R5),R2        ;GET VCB ADDRESS
        53 A2       04   8A   00FF           BICB2   #1@VCB$V_MOUNTVER,VCB$B_STATUS2(R2)     ;ZAP MOUNT VERIFICATION
                         05   0103           RSB                             ;RETURN NOW
```

```
    0104    333
    0104    334 70$:    CON_MSG IPC_ILLDEVICE,RETURN    ;ELSE ISSUE MESSAGE AND RETURN
    010B    335 80$:    CON_MSG IPC_NOMNTVER,RETURN     ;ELSE ISSUE ERROR AND RETURN
```

```
                        0112          .SBTTL  COMMAND LINE INPUT ROUTINE
                        0112
                        0112  ;++
                        0112  ;
                        0112  ; IPC_GETLINE - GET A LINE OF INPUT FROM THE CONSOLE
                        0112  ;
                        0112  ; THIS ROUTINE INPUTS A LINE FROM THE CONSOLE. IT IGNORES LEADING SPACES,
                        0112  ; COMPRESSES MULTIPLE SPACES TO A SINGLE SPACE, CONVERTS LOWER CASE TO UPPER
                        0112  ; CASE, HANDLES CHARACTER (RUBOUT) AND LINE (CONTROL-U) DELETION, AND CHECKS
                        0112  ; THE LEGALITY OF ANY CHARACTER RECEIVED.
                        0112  ;
                        0112  ; INPUTS:
                        0112  ;
                        0112  ;       R6 - ADDRESS OF THE INPUT LINE SIZE STORAGE
                        0112  ;       R7 - ADDRESS OF THE INPUT LINE BUFFER
                        0112  ;       R11 - ADDRESS OF THE LOCAL STORAGE AREA
                        0112  ;
                        0112  ; OUTPUTS:
                        0112  ;
                        0112  ;       IPC_W_INPSIZE(R11) - SIZE OF USER INPUT LINE AFTER COMPRESSION
                        0112  ;       IPC_T_INPLINE(R11) - STORAGE FOR THE CONSOLE INPUT
                        0112  ;
                        0112  ;       R0,R1,R2, AND R8 ARE DESTROYED
                        0112  ;
                        0112  ;--
                        0112
                        0112  IPC_GETLINE:
              68    94  0112          CLRB    (R8)                    ;RESET ALL FLAGS
         52   57    D0  0114          MOVL    R7,R2                   ;COPY BUFFER ADDRESS
              66    B4  0117  10$:    CLRW    (R6)                    ;RESET INPUT LINE SIZE
         68   02    88  0119          BISB2   #IPC_M_NEWLINE,(R8)     ;NEW LINE ONLY USEFUL FLAG
         50   20    DB  011C  20$:    MFPR    #PR$_RXCS,R0            ;GET DEVICE STATUS
      F9 50   07    E1  011F          BBC     #7,R0,20$               ;XFER IF NOT READY
         50   21    DB  0123          MFPR    #PR$_RXDB,R0            ;ELSE GET THE CHARACTER
   50   FF00  8F    B3  0126          BITW    #^XFF00,R0              ;ERROR OF NOT CONSOLE INPUT?
              EF    12  012B          BNEQ    20$                     ;YES, IGNORE
   50     80  8F    8A  012D          BICB2   #^X80,R0                ;ZAP PARITY BIT IF ANY
49 006A'CF    50    E1  0131          BBC     R0,W^IPC_V_LEGAL,70$    ;CHECK LEGALITY
      7F 8F   50    91  0137          CMPB    R0,#IPC_C_DELETE        ;CHECK FOR CHARACTER DELETE
              19    12  013B          BNEQ    35$                     ;XFER IF NOT
      DB 68   01    E0  013D          BBS     #IPC_V_NEWLINE,(R8),20$ ;IGNORE IF NOTHING THERE
      06 68   00    E2  0141          BBSS    #IPC_V_DELETE,(R8),30$  ;XFER IF HERE ONCE ALREADY
   50     5C  8F    9A  0145          MOVZBL  #^A/\/,R0               ;SET DELIMITER
              6B    10  0149          BSBB    IPC_OUTCHR              ;TYPE IT OUT
              66    B7  014B  30$:    DECW    (R6)                    ;ANYTHING TO DELETE?
              C8    19  014D          BLSS    10$                     ;XFER IF NOT...SET NEW LINE
         50   72    9A  014F          MOVZBL  -(R2),R0                ;GET CHARACTER
              62    10  0152          BSBB    IPC_OUTCHR              ;TYPE IT OUT
              C6    11  0154          BRB     20$                     ;GET NEXT CHARACTER FROM THE CONSOLE
         1A   50    91  0156  35$:    CMPB    R0,#IPC_C_CNTRLZ        ;END OF THE LINE?
              05    12  0159          BNEQ    40$                     ;XFER IF NOT
         68   04    88  015B          BISB2   #IPC_M_CNTRLZ,(R8)      ;ELSE NOTE IT
              05    11  015E          BRB     45$                     ;GO FINISH UP
         15   50    91  0160  40$:    CMPB    R0,#IPC_C_CNTRLU        ;DELETE ENTIRE LINE?
              07    12  0163          BNEQ    50$                     ;XFER IF NOT
              4F    10  0165  45$:    BSBB    IPC_OUTCHR              ;ECHO ^U
              72    10  0167          BSBB    IPC_CRLF                ;NEW LINE
```

```
       66  B4  0169        CLRW    (R6)                      ;NOTHING THERE (FORCE NEW PROMPT)
           05  016B        RSB                               ;RETURN TO MAIN ROUTINE
    20 50  91  016C  50$:  CMPB    R0,#IPC_C_SPACE           ;SPACE CHARACTER?
       0A  12  016F        BNEQ    60$                       ;XFER IF NOT
 A7 68 01  E0  0171        BBS     #IPC_V_NEWLINE,(R8),20$   ;XFER IF LEADING SPACE
 20  FF A2  91  0175        CMPB    -1(R2),#IPC_C_SPACE       ;PREVIOUS CHARACTER A SPACE?
       A1  13  0179        BEQL    20$                       ;IGNORE IF SO
    0B 66  B1  017B  60$:  CMPW    (R6),#IPC_C_MAXLINSIZ     ;TOO MANY CHARACTERS?
       07  19  017E        BLSS    80$                       ;XFER IF NOT
    50 07  9A  0180  70$:  MOVZBL  #IPC_C_BELL,R0            ;ELSE SET NEW CHARACTER
       31  10  0183        BSBB    IPC_OUTCHR                ;LET USER KNOW
       95  11  0185        BRB     20$                       ;GO GET THE NEXT CHARACTER
 61 8F 50  91  0187  80$:  CMPB    R0,#IPC_C_LOWERA          ;LOWER CASE CHARACTER?
       03  19  018B        BLSS    90$                       ;XFER IF NOT
    50 20  8A  018D        BICB2   #^X20,R0                  ;ELSE CONVERT TO UPPER CASE
 0A 68 00  E1  0190  90$:  BBC     #IPC_V_DELETE,(R8),100$   ;FINISH OUT DELETE STRING?
       01  BB  0194        PUSHR   #^M<R0>                   ;YES, SAVE CURRENT CHARACTER
 50   5C 8F  9A  0196        MOVZBL  #^A/\/,R0                 ;SET TERMINATOR
       1A  10  019A        BSBB    IPC_OUTCHR                ;TYPE IT OUT
       01  BA  019C        POPR    #^M<R0>                   ;RESTORE ORIGINAL CHARACTER
    82 50  90  019E 100$:  MOVB    R0,(R2)+                  ;SAVE THE CHARACTER
    68 02  8A  01A1        BICB2   #IPC_M_NEWLINE,(R8)       ;NO LONGER EMPTY LINE
       10  10  01A4        BSBB    IPC_OUTCHR                ;TYPE OUT THE CHARACTER
    0D 50  91  01A6        CMPB    R0,#IPC_C_CARRETN         ;END OF THE LINE?
       06  12  01A9        BNEQ    110$                      ;XFER IF MORE TO COME
    50 0A  9A  01AB        MOVZBL  #IPC_C_LINFEED,R0         ;ELSE FOLLOW WITH A <LF>
       06  10  01AE        BSBB    IPC_OUTCHR
           05  01B0        RSB                               ;RETURN TO CALLER
       66  B6  01B1 110$:  INCW    (R6)                      ;ONE MORE CHARACTER
     FF66  31  01B3        BRW     20$                       ;GO GET ANOTHER
```

```
                             01B6   425         .SBTTL  MISCELLANEOUS CONSOLE OUTPUT ROUTINES
                             01B6   426
                             01B6   427 ;++
                             01B6   428 ;
                             01B6   429 ; IPC_OUTCHR - OUTPUT A SINGLE CHARACTER
                             01B6   430 ;
                             01B6   431 ; THIS ROUTINE OUTPUTS A SINGLE CHARACTER TO THE CONSOLE.  IF IS A SPECIAL
                             01B6   432 ; ACTION CONTROL CHARACTER (CR, BELL, OR LF) THEN IT IS TYPED DIRECTLY.
                             01B6   433 ; OTHERWISE IT IS PRECEEDED BY AN UP-ARROW (^).
                             01B6   434 ;
                             01B6   435 ; INPUTS:
                             01B6   436 ;
                             01B6   437 ;       R0 - CHARACTER TO TYPE
                             01B6   438 ;
                             01B6   439 ; OUTPUTS:
                             01B6   440 ;
                             01B6   441 ;       NONE
                             01B6   442 ;
                             01B6   443 ;       R1 DESTROYED
                             01B6   444 ;
                             01B6   445 ;--
                             01B6   446
                             01B6   447 IPC_OUTCHR:
                  50     DD  01B6   448         PUSHL   R0                      ;SAVE THE CHARACTER
      12 007A'CF  6E     E1  01B8   449         BBC     (SP),W^IPC_V_ARROW,20$  ;PRECEED WITH AN UP ARROW?
              51  22     DB  01BE   450 10$:    MFPR    #PR$_TXCS,R1            ;YES, GET CONSOLE STATUS
           F9 51  07     E1  01C1   451         BBC     #7,R1,10$               ;XFER IF NOT READY
   23   0000005E  8F     DA  01C5   452         MTPR    #^A\^\,#PR$_TXDB        ;ELSE TYPE THE UP ARROW
         6E    40 8F     88  01CC   453         BISB2   #^X40,(SP)              ;CONVERT TO A PRINTABLE CHARACTER
              51  22     DB  01D0   454 20$:    MFPR    #PR$_TXCS,R1            ;GET DEVICE STATUS
           F9 51  07     E1  01D3   455         BBC     #7,R1,20$               ;XFER IF NOT READY
              23  8E     DA  01D7   456         MTPR    (SP)+,#PR$_TXDB         ;ELSE OUTPUT THE CHARACTER
                         05  01DA   457         RSB                             ;RETURN TO CALLER
                             01DB   458
                             01DB   459 ;--
                             01DB   460 ;
                             01DB   461 ; IPC_CRLF - OUTPUT A <CR,LF> SEQUENCE
                             01DB   462 ;
                             01DB   463 ; THIS ROUTINE OUTPUTS A <CR,LF> SEQUENCE TO THE CONSOLE.  IT FALLS THROUGH
                             01DB   464 ; TO THE STRING OUTPUT ROUTINE TO AVOID THE UNNEEDED SUBROUTINE CALL AND RETURN.
                             01DB   465 ; (THE RETURN IS DONE BY THE STRING OUTPUT ROUTINE.)
                             01DB   466 ;
                             01DB   467 ; INPUTS:
                             01DB   468 ;
                             01DB   469 ;       NONE
                             01DB   470 ;
                             01DB   471 ; OUTPUTS:
                             01DB   472 ;
                             01DB   473 ;       NONE
                             01DB   474 ;
                             01DB   475 ;--
                             01DB   476
                             01DB   477 IPC_CRLF:
        52    0061'CF    9E  01DB   478         MOVAB   W^IPC_CRLFSTR,R2        ;SET MESSAGE ADDRESS
                             01E0   479                                         ;FALL THROUGH INTO STRING ROUTINE
                             01E0   480
                             01E0   481 ;++
                        01E0  ;
                        01E0  ; IPC_OUTPUTZ - TYPE OUT AN .ASCIZ STRING ON THE CONSOLE
                        01E0  ;
                        01E0  ; THIS ROUTINE TYPES OUT A ZERO TERMINATED STRING ON THE CONSOLE.
                        01E0  ;
                        01E0  ; INPUTS:
                        01E0  ;
                        01E0  ;       R2 - ADDRESS OF THE STRING TO TYPE
                        01E0  ;
                        01E0  ; OUTPUTS:
                        01E0  ;
                        01E0  ;       NONE
                        01E0  ;
                        01E0  ;--
                        01E0
                        01E0  IPC_OUTPUTZ:
     50    82    9A     01E0          MOVZBL  (R2)+,R0                ;GET A CHARACTER
           04    13     01E3          BEQL    10$                     ;XFER WHEN DONE
           CF    10     01E5          BSBB    IPC_OUTCHR              ;ELSE TYPE OUT THE CHARACTER
           F7    11     01E7          BRB     IPC_OUTPUTZ             ;GO GET THE NEXT
                 05     01E9  10$:    RSB                             ;RETURN WHEN DONE
                        01EA
                        01EA
                        01EA  ;++
                        01EA  ;
                        01EA  ; IPC_FORK_QUORUM
                        01EA  ;
                        01EA  ; THIS ROUTINE QUEUES A FORK TO IPL$_SYNCH IN ORDER TO SYNCHRONIZE WITH
                        01EA  ; THE CLUSTER CONNECTION MANAGER. THE FORK ROUTINE REQUESTS A RECALCULATION
                        01EA  ; OF DYNAMIC QUORUM BASED ON THE CURRENT CLUSTER CONFIGURATION. THIS IS DONE
                        01EA  ; WHEN THE CLUSTER IS HUNG DUE TO LACK OF QUORUM BECAUSE A CRITICAL NODE HAS
                        01EA  ; CRASHED AND CANNOT BE REBOOTED.
                        01EA  ;
                        01EA  ; INPUTS:
                        01EA  ;
                        01EA  ;       NONE
                        01EA  ;
                        01EA  ; OUTPUTS:
                        01EA  ;
                        01EA  ;       NONE
                        01EA  ;
                        01EA  ;--
                        01EA          ASSUME  FKB$B_TYPE EQ FKB$W_SIZE+2
                        01EA          ASSUME  FKB$B_FIPL EQ FKB$W_SIZE+3
                        01EA  FKB_INIT:                               ;INITIALIZATION FOR FORK BLOCK
               0018     01EA          .WORD   FKB$K_LENGTH            ; FKB$W_SIZE
                 08     01EC          .BYTE   DYN$C_FRK               ; FKB$B_TYPE
                 08     01ED          .BYTE   IPL$_SYNCH              ; FKB$B_FIPL
                        01EE
                        01EE  IPC_FORK_QUORUM:
                        01EE  ;
                        01EE  ; ALLOCATE A FORK BLOCK FROM NONPAGED POOL. INITIALIZE IT.
                        01EE  ;
     51    18    D0     01EE          MOVL    #FKB$K_LENGTH,R1        ;LENGTH OF FORK BLOCK
00000000'GF      16     01F1          JSB     G^EXE$ALONONPAGED       ;GET THE BLOCK
        26 50    E9     01F7          BLBC    R0,10$                  ;IF LBC, DIDN'T GET IT.
     55    52    D0     01FA          MOVL    R2,R5                   ;PUT IT WHERE IT BELONGS
```

```
  08 A5    EA AF    D0  01FD             MOVL    FKB_INIT,FKB$W_SIZE(R5)          ;INIT SIZE, TYPE, FIPL
                        0202       ;
                        0202       ; FORK DOWN TO IPL$_SYNCH ...
                        0202       ;
                        0202             FORK                                     ;QUEUE FORK TO IPL$_SYNCH
                        0208                                                      ;AND RETURN TO CALLER
                        0208       ;
                        0208       ; AT THIS STAGE, WE ARE EXECUTING IN THE FORK AT IPL SYNCH.
                        0208       ;
                 55     DD  0208   5$:   PUSHL   R5                               ;SAVE FORK BLOCK POINTER
                 51     D4  020A         CLRL    R1                               ;QUORUM=0 MEANS YOU FIGURE IT OUT.
          00000000'GF   16  020C         JSB     G^CNX$CHANGE_QUORUM              ;CALL CONNECTION MANAGER
                 55   8ED0  0212         POPL    R5                               ;RESTORE FORK BLOCK POINTER
                        0215       ;
                        0215       ; IF WE GET ANY ERROR RETURNED, WAIT ONE SECOND AND RETRY.
                        0215       ; CONNECTION MANAGER CANNOT ALWAYS DO THIS THE FIRST TIME IT TRIES.
                        0215       ;
              08 50     E8  0215         BLBS    R0,10$                           ;LBS MEANS IT WORKED
                        0218             FORK_WAIT                                ;WAIT ONE SECOND
                 E8     11  021E         BRB     5$                               ;TRY AGAIN.
                        0220                                                      ;EVENTUALLY IT WILL WORK (SEZ HERE)
                        0220
           50    55     D0  0220   10$:  MOVL    R5,R0                            ;COPY FORK BLOCK ADDRESS
          00000000'GF   16  0223         JSB     G^EXE$DEANONPAGED                ;DEALLOCATE THE FORK BLOCK
                        05  0229         RSB                                      ;DONE
```

```
     022A   565          .END
```

```
CLU$GL_CLUB                      ********   X    03
CLUB$L_LOCAL_CSB               = 00000010
CNX$CHANGE_QUORUM                ********   X    03
CON$OWNCTY                       ********   X    03
CON$RELEASECTY                   ********   X    03
CSB$L_STATUS                   = 00000060
CSB$V_MEMBER                   = 00000001
DC$_DISK                       = 00000001
DYN$C_FRK                      = 00000008
EXE$ALONONPAGED                  ********   X    03
EXE$DEANONPAGED                  ********   X    03
EXE$FORK                         ********   X    03
EXE$FORK_WAIT                    ********   X    03
EXE$IPCONTROL                    00000000 RG     03
FKB$B_FIPL                     = 0000000B
FKB$B_TYPE                     = 0000000A
FKB$K_LENGTH                   = 00000018
FKB$W_SIZE                     = 00000008
FKB_INIT                         000001EA R      03
INI$BRK                          ********   X    03
IOC$M_ANY                      = 00000040
IOC$M_PHY                      = 00000001
IOC$PARSDEVNAM                   ********   X    03
IOC$SEARCHINT                    ********   X    03
IPC_B_FLAGS                      00000015
IPC_B_PAD                        00000022
IPC_CANCEL                       000000A0 R      03
IPC_CRLF                         000001DB R      03
IPC_CRLFSTR                      00000061 R      02
IPC_C_BELL                     = 00000007
IPC_C_CARRETN                  = 0000000D
IPC_C_CHAR_C                   = 00000043
IPC_C_CHAR_Q                   = 00000051
IPC_C_CHAR_X                   = 00000058
IPC_C_CNTRLU                   = 00000015
IPC_C_CNTRLZ                   = 0000001A
IPC_C_COLON                    = 0000003A
IPC_C_DELETE                   = 0000007F
IPC_C_LENGTH                     00000024
IPC_C_LINFEED                  = 0000000A
IPC_C_LOWERA                   = 00000061
IPC_C_MAXLINSIZ                = 0000000B
IPC_C_SPACE                    = 00000020
IPC_C_START                      00000000
IPC_C_UNDER                    = 0000005F
IPC_ENABLE                       0000008D R      03
IPC_FORK_QUORUM                  000001EE R      03
IPC_GETLINE                      00000112 R      03
IPC_ILLCOMMAND                   00000000 R      02
IPC_ILLDEVICE                    00000012 R      02
IPC_L_OLDGETNXT                  00000016
IPC_L_SAVRXCS                    0000001A
IPC_L_SAVTXCS                    0000001E
IPC_M_CNTRLZ                   = 00000004
IPC_M_NEWLINE                  = 00000002
IPC_NOCLUSTER                    00000048 R      02
IPC_NOMNTVER                     00000023 R      02
IPC_OUTCHR                       000001B6 R      03
IPC_OUTPUTZ                      000001E0 R      03
IPC_PROMPT                       00000064 R      02
IPC_Q_DEVICE                     0000000D
IPC_T_INPLINE                    00000002
IPC_V_ARROW                      0000007A R      02
IPC_V_CNTRLZ                   = 00000002
IPC_V_DELETE                   = 00000000
IPC_V_LEGAL                      0000006A R      02
IPC_V_NEWLINE                  = 00000001
IPC_W_INPSIZE                    00000000
IPL$_SYNCH                     = 00000008
PR$_RXCS                       = 00000020
PR$_RXDB                       = 00000021
PR$_TXCS                       = 00000022
PR$_TXDB                       = 00000023
SIZ...                         = 00000001
UCB$B_DEVCLASS                 = 00000040
UCB$L_VCB                      = 00000034
UCB$V_MNTVERIP                 = 0000000E
UCB$W_STS                      = 00000064
VCB$B_STATUS2                  = 00000053
VCB$V_MOUNTVER                 = 00000002
```

```
                                        +----------------+
                                        ! Psect synopsis !
                                        +----------------+

PSECT name                     Allocation          PSECT No.  Attributes
----------                     ----------          ---------  ----------
.  ABS  .                      00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                          00000024  (    36.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
WMOUNTVERMSG                   0000008A  (   138.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
WIONONPAGED                    0000022A  (   554.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG

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

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                     29    00:00:00.06    00:00:01.07
Command processing                105    00:00:00.56    00:00:03.90
Pass 1                            391    00:00:14.00    00:00:43.52
Symbol table sort                   0    00:00:02.27    00:00:08.11
Pass 2                            112    00:00:02.72    00:00:10.66
Symbol table output                 9    00:00:00.10    00:00:00.23
Psect synopsis output               3    00:00:00.02    00:00:00.03
Cross-reference output              0    00:00:00.00    00:00:00.00
Assembler run totals              651    00:00:19.73    00:01:07.67

The working set limit was 1350 pages.
81107 bytes (159 pages) of virtual memory were used to buffer the intermediate c  .
There were 80 pages of symbol table space allocated to hold 1539 non-local and   local symbols.
565 source lines were read in Pass 1, producing 16 object records in Pass 2.
27 pages of virtual memory were used to define 26 macros.

                                    +----------------------------+
                                    ! Macro library statistics !
                                    +----------------------------+

Macro library name                         Macros defined
------------------                         --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                       12
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     8
TOTALS (all libraries)                               20

1604 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:IPCONTROL/OBJ=OBJ$:IPCONTROL MSRC$:IPCONTROL/UPDATE=(ENH$:IPCONTROL)+EXECML$/LIB
```

0376 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY