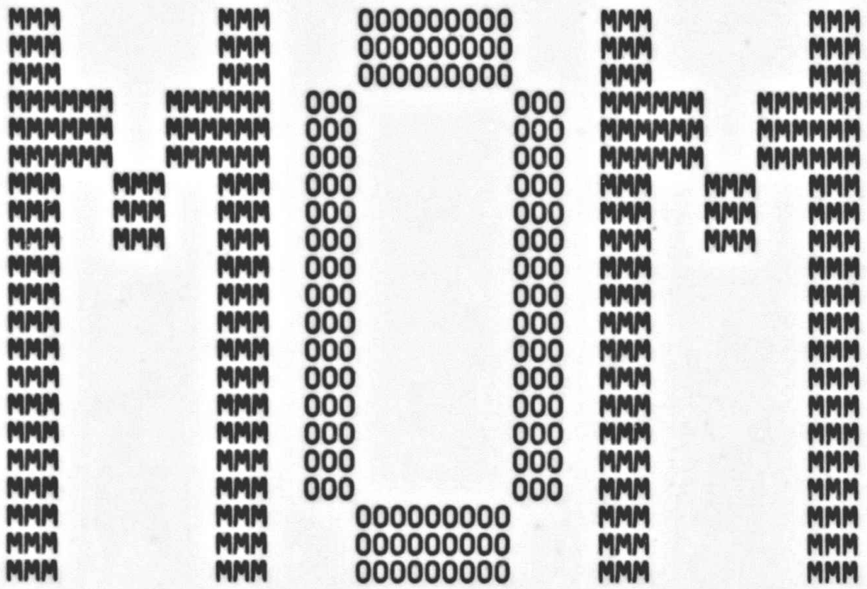
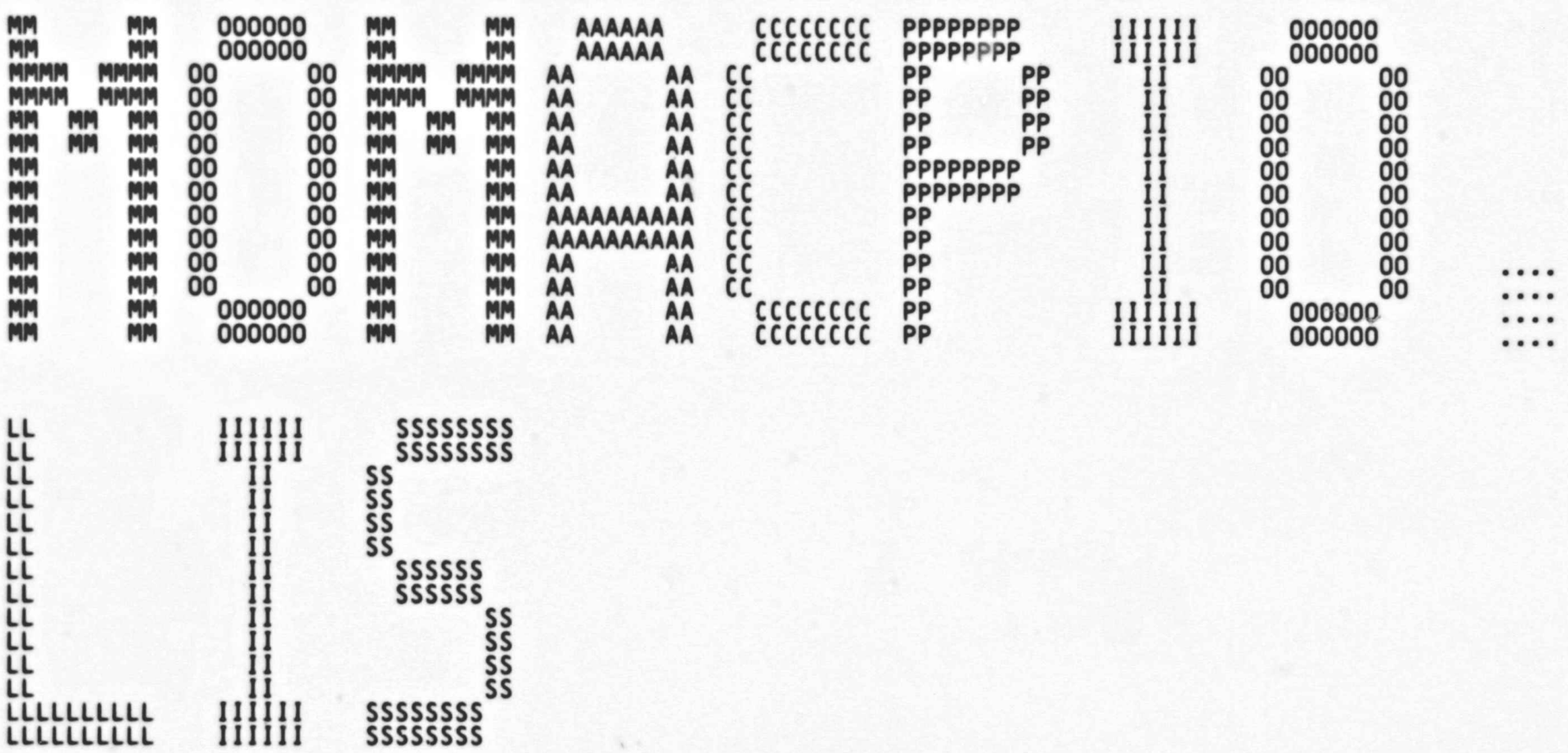

MMM MMM 000000000 MMM MMM
MMM MMM 000000000 MMM MMM
MMM MMM 000000000 MMM MMM
MMMMMM MMMMMM 000 000 MMMMMM MMMMMM
MMMMMM MMMMMM 000 000 MMMMMM MMMMMM
MMMMMM MMMMMM 000 000 MMMMMM MMMMMM
MMM MMM MMM 000 000 MMM MMM MMM
MMM MMM MMM 000 000 MMM MMM MMM
MMM MMM MMM 000 000 MMM MMM MMM
MMM MMM 000 000 MMM MMM
MMM MMM 000 000 MMM MMM
MMM MMM 000 000 MMM MMM
MMM MMM 000 000 MMM MMM
MMM MMM 000 000 MMM MMM
MMM MMM 000 000 MMM MMM
MMM MMM 000 000 MMM MMM
MMM MMM 000 000 MMM MMM
MMM MMM 000 000 MMM MMM
MMM MMM 000000000 MMM MMM
MMM MMM 000000000 MMM MMM
MMM MMM 000000000 MMM MMM


MOMACPIO

```
;  1          0001  0 %TITLE 'MOM Network I/O module'
;  2          0002  0 MODULE MOMACPIO (
;  3          0003  0                 LANGUAGE (BLISS32),
;  4          0004  0                 ADDRESSING_MODE (NONEXTERNAL=GENERAL),
;  5          0005  0                 ADDRESSING_MODE (EXTERNAL=GENERAL),
;  6          0006  0                 IDENT = 'V04-000'
;  7          0007  0                 ) =
;  8          0008  1 BEGIN
;  9          0009  1 !
; 10          0010  1 !*****************************************************************************
; 11          0011  1 !*                                                                           *
; 12          0012  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 13          0013  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 14          0014  1 !*  ALL RIGHTS RESERVED.                                                     *
; 15          0015  1 !*                                                                           *
; 16          0016  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 17          0017  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 18          0018  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 19          0019  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 20          0020  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 21          0021  1 !*  TRANSFERRED.                                                             *
; 22          0022  1 !*                                                                           *
; 23          0023  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 24          0024  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 25          0025  1 !*  CORPORATION.                                                             *
; 26          0026  1 !*                                                                           *
; 27          0027  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 28          0028  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 29          0029  1 !*                                                                           *
; 30          0030  1 !*                                                                           *
; 31          0031  1 !*****************************************************************************
; 32          0032  1
; 33          0033  1 !++
; 34          0034  1 ! FACILITY:  DECnet-VAX Network Management Maintenance Operations Module (MOM)
; 35          0035  1 !
; 36          0036  1 ! ABSTRACT:
; 37          0037  1 !       This module contains routines to handle all network I/O
; 38          0038  1 !       with NETACP (NETDRIVER).
; 39          0039  1 !
; 40          0040  1 ! ENVIRONMENT:  VAX/VMS Operating System
; 41          0041  1 !
; 42          0042  1 ! AUTHOR:  Kathy Perko
; 43          0043  1 !
; 44          0044  1 ! CREATION DATE:  11-Jan-1983
; 45          0045  1 !
; 46          0046  1 ! MODIFIED BY:
; 47          0047  1 !       V03-001 MKP0001         Kathy Perko             2-June-1983
; 48          0048  1 !               Fix code that maps parameter IDs from NETUSR values
; 49          0049  1 !               to NICE values.
; 50          0050  1 !
; 51          0051  1 !--
; 52          0052  1
```

```
;    54          0053  1 %SBTTL 'Declarations'
;    55          0054  1
;    56          0055  1 !
;    57          0056  1 ! TABLE OF CONTENTS:
;    58          0057  1 !
;    59          0058  1
;    60          0059  1 FORWARD ROUTINE
;    61          0060  1     mom$build_p2: NOVALUE,
;    62          0061  1     mom$netacp_qio,
;    63          0062  1     mom$mapqioerror,
;    64          0063  1     mom_mapentity,
;    65          0064  1     mom_mapparamid;
;    66          0065  1
;    67          0066  1 !
;    68          0067  1 ! INCLUDE FILES:
;    69          0068  1 !
;    70          0069  1
;    71          0070  1 LIBRARY 'LIB$:MOMLIB';                  ! Facility-wide definitions
;    72          0071  1 LIBRARY 'SHRLIB$:NMALIBRY';             ! NICE definitions
;    73          0072  1 LIBRARY 'SHRLIB$:NET';                  ! NETACP QIO interface
;    74          0073  1 LIBRARY 'SYS$LIBRARY:STARLET';          ! VMS common definitions
;    75          0074  1
;    76          0075  1 !
;    77          0076  1 ! OWN STORAGE:
;    78          0077  1 !
;    79          0078  1
;    80          0079  1 !
;    81          0080  1 ! EXTERNAL REFERENCES:
;    82          0081  1 !
;    83          0082  1
;    84          0083  1 $mom_externals;
;    85          0084  1
;    86          0085  1 EXTERNAL
;    87          0086  1     mom$gq_netnamdsc,
;    88          0087  1     mom$gw_acp_chan;
;    89          0088  1
;    90          0089  1 EXTERNAL LITERAL
;    91          0090  1     mom$_qiobfovf;
;    92          0091  1
;    93          0092  1 EXTERNAL ROUTINE
;    94          0093  1     mom$debug_qio,
;    95          0094  1     mom$bld_reply;
;    96          0095  1
```

```
;    98          0096  1 %SBTTL 'mom$build_p2      Build P2 buffer and descriptor'
;    99          0097  1 GLOBAL ROUTINE mom$build_p2 (len1, adr1, len2, adr2, p2dsc, resdsc) : NOVALUE =
;   100          0098  1
;   101          0099  1 !++
;   102          0100  1 ! FUNCTIONAL DESCRIPTION:
;   103          0101  1 !
;   104          0102  1 !       This routine builds the P2 buffer and descriptor for operations to
;   105          0103  1 !       NETACP.  The buffer contains one or two search key values.  These
;   106          0104  1 !       values are compared to fields in NETACPs database entries (the database
;   107          0105  1 !       and the fields (search keys) are specified in the NFB buffer).  When
;   108          0106  1 !       an entry is found in which the specified field(s) match the search
;   109          0107  1 !       key value(s), NETACP performs the operation requested by the rest of
;   110          0108  1 !       the NFB (set, snow, etc.)
;   111          0109  1 !
;   112          0110  1 ! FORMAL PARAMETERS:
;   113          0111  1 !
;   114          0112  1 !       LEN1            First search key length.  If LEN1 is:
;   115          0113  1 !                           - zero then ADR1 contains a longword search key.
;   116          0114  1 !                           - >0 it contains the length of a string which
;   117          0115  1 !                             ADR1 points to.
;   118          0116  1 !                           - -1 then search key ID is a wildcard, and nothing
;   119          0117  1 !                             needs to be put into the P2 buffer for it.
;   120          0118  1 !                           - -2 then ADR1 contains a word search key.
;   121          0119  1 !       ADR1            First search key address.  If LEN1 is zero then this
;   122          0120  1 !                       is the longword value of the search key.  If LEN1 is -1 then
;   123          0121  1 !                       the search key is omitted.
;   124          0122  1 !       LEN2            Second search  key length.  Same rules apply as for
;   125          0123  1 !                       LEN1.
;   126          0124  1 !       ADR2            Second search key address. Same rules apply as for
;   127          0125  1 !                       ADR1.
;   128          0126  1 !       P2DSC           Address of P2 descriptor.  This routine assumes that
;   129          0127  1 !                       the buffer is large enough to handle the result.
;   130          0128  1 !                       The maximum P2 buffer required by NML is 36 bytes.
;   131          0129  1 !       RESDSC          Address of descriptor to hold resulting P2.
;   132          0130  1 !
;   133          0131  1 ! IMPLICIT OUTPUTS:
;   134          0132  1 !       The buffer described by P2DSC contains the search key and
;   135          0133  1 !       start key information.
;   136          0134  1 !
;   137          0135  1 !--
;   138          0136  1
;   139          0137  2 BEGIN
;   140          0138  2
;   141          0139  2 MAP
;   142          0140  2     p2dsc  : REF VECTOR,
;   143          0141  2     resdsc : REF VECTOR;
;   144          0142  2
;   145          0143  2 OWN
;   146          0144  2     collate_start_value: VECTOR [nfb$c_ctx_size, BYTE]
;   147          0145  2                          INITIAL ( REP nfb$c_ctx_size OF BYTE (0));
;   148          0146  2
;   149          0147  2 LOCAL
;   150          0148  2     msgsize,
;   151          0149  2     count,                                  ! P2 buffer length
;   152          0150  2     ptr;                                    ! P2 buffer pointer
;   153          0151  2
;   154          0152  2 !
```

```
;  155     0153  2 ! Calculate the length of the resulting P2 buffer, and signal if
;  156     0154  2 ! the buffer supplied isn't big enough.
;  157     0155  2 !
;  158     0156  2 count = 4;                              ! Account for count at beginning of buffer.
;  159     0157  2 SELECTONE .len1 OF
;  160     0158  2     SET
;  161     0159  2     [-2]: count = .count + 2;                       ! It's a word
;  162     0160  2     [0]: count = .count + 4;                        ! It's a longword
;  163     0161  2     [1 to 255]: count = .count + .len1 + 2          ! It's a string.
;  164     0162  2     TES;
;  165     0163  2
;  166     0164  2 SELECTONE .len2 OF
;  167     0165  2     SET
;  168     0166  2     [-2]: count = .count + 2;                       ! It's a word
;  169     0167  2     [0]: count = .count + 4;                        ! It's a longword
;  170     0168  2     [1 to 255]: count = .count + .len2 + 2          ! It's a string.
;  171     0169  2     TES;
;  172     0170  2
;  173     0171  2 count = .count + nfb$c_ctx_size;
;  174     0172  2 IF .count GTR .p2dsc [0] THEN
;  175     0173  2     !
;  176     0174  2     ! The P2 buffer will overflow.  Signal a MOM error.
;  177     0175  2     !
;  178     0176  3     BEGIN
;  179     0177  3     mom$ab_msgblock [msb$l_flags] = msb$m_msg_fld;      ! Set message text flag.
;  180     0178  3     mom$ab_msgblock [msb$b_code] = nma$c_sts_mpr;
;  181     0179  3     mom$ab_msgblock [msb$l_text] = mom$_qiobfovf;
;  182     0180  3     mom$bld_reply (mom$ab_msgblock, msgsize);       ! Build message
;  183     0181  3     $signal_msg (mom$ab_nice_xmit_buf, .msgsize);   ! Signal it.
;  184     0182  2     END;
;  185     0183  2
;  186     0184  2 ptr = .p2dsc [1] + 4;   ! Skip over return count
;  187     0185  2
;  188     0186  2 !
;  189     0187  2 ! Add first search key value to the P2 buffer.
;  190     0188  2 !
;  191     0189  2 SELECTONE .len1 OF
;  192     0190  2     SET
;  193     0191  2     [-2]: ptr = CH$MOVE (2, adr1, .ptr);            ! It's a word
;  194     0192  2     [0]: ptr = CH$MOVE (4, adr1, .ptr);             ! It's a longword
;  195     0193  2     [1 TO 255]:                                     ! It's a string.
;  196     0194  3         BEGIN
;  197     0195  3         CH$WCHAR_A (.len1<0,8>, ptr);
;  198     0196  3         CH$WCHAR_A (.len1<8,8>, ptr);
;  199     0197  3         ptr = CH$MOVE (.len1, .adr1, .ptr);
;  200     0198  3         END
;  201     0199  2     TES;
;  202     0200  2
;  203     0201  2 !
;  204     0202  2 ! Add search key two to buffer.
;  205     0203  2 !
;  206     0204  2 SELECTONE .len2 OF
;  207     0205  2     SET
;  208     0206  2     [-2]: ptr = CH$MOVE (2, adr2, .ptr);            ! It's a word
;  209     0207  2     [0]: ptr = CH$MOVE (4, adr2, .ptr);             ! It's a longword
;  210     0208  2     [1 TO 255]:                                     ! It's a string.
;  211     0209  3         BEGIN
```

```
;  212     0210  3          CH$WCHAR_A (.len2<0,8>, ptr);
;  213     0211  3          CH$WCHAR_A (.len2<8,8>, ptr);
;  214     0212  3          ptr = CH$MOVE (.len2, .adr2, .ptr);
;  215     0213  3          END
;  216     0214  2      TES;
;  217     0215  2
;  218     0216  2  !
;  219     0217  2  ! Set up a context area of a string of nulls.  NETACP will
;  220     0218  2  ! replace the null string with a start value of the last database
;  221     0219  2  ! entry matched by the search key.
;  222     0220  2  !
;  223     0221  2  ptr = CH$MOVE ( nfb$c_ctx_size, collate_start_value, .ptr);
;  224     0222  2  !
;  225     0223  2  ! Set up resulting descriptor for return.
;  226     0224  2  !
;  227     0225  2  resdsc [0] = .ptr - .p2dsc [1];
;  228     0226  2  resdsc [1] = .p2dsc [1];
;  229     0227  2
;  230     0228  1  END;                                   ! End of mom$build_p2


                                                        .TITLE  MOMACPIO MOM Network I/O module
                                                        .IDENT  \V04-000\

                                                        .PSECT  $OWN$,NOEXE,2

                                                00# 00000 COLLATE_START_VALUE:
                                                        .BYTE   0[64]                                   ;

                                                        .EXTRN  MOM$GL_LOGMASK, MOM$GL_SVD_INDEX
                                                        .EXTRN  MOM$AB_SERVICE_DATA
                                                        .EXTRN  MOM$GB_FUNCTION
                                                        .EXTRN  MOM$GB_OPTION_BYTE
                                                        .EXTRN  MOM$GB_ENTITY_CODE
                                                        .EXTRN  MOM$AB_ENTITY_BUF
                                                        .EXTRN  MOM$GQ_ENTITY_BUF_DSC
                                                        .EXTRN  MOM$GL_SERVICE_FLAGS
                                                        .EXTRN  MOM$AB_NPARSE_BLK
                                                        .EXTRN  MOM$AB_NICE_RCV_BUF
                                                        .EXTRN  MOM$AB_NICE_XMIT_BUF
                                                        .EXTRN  MOM$GQ_NICE_RCV_BUF_DSC
                                                        .EXTRN  MOM$GL_NICE_RCV_MSG_LEN
                                                        .EXTRN  MOM$GQ_NICE_XMIT_BUF_DSC
                                                        .EXTRN  MOM$AB_MSGBLOCK
                                                        .EXTRN  MOM$AB_ACPQIO_BUFFER
                                                        .EXTRN  MOM$GQ_ACPQIO_BUF_DSC
                                                        .EXTRN  MOM$AB_CIB, MOM$AB_LOOP_CIB
                                                        .EXTRN  MOM$AB_TRIGGER_CIB
                                                        .EXTRN  MOM$AB_MOP_XMIT_BUF
                                                        .EXTRN  MOM$GQ_MOP_XMIT_BUF_DSC
                                                        .EXTRN  MOM$AB_MOP_RCV_BUF
                                                        .EXTRN  MOM$GQ_MOP_RCV_BUF_DSC
                                                        .EXTRN  MOM$AB_MOP_MSG, MOM$GQ_MOP_MSG_DSC
                                                        .EXTRN  MOM$GW_EVT_CODE
                                                        .EXTRN  MOM$GB_EVT_POPR
                                                        .EXTRN  MOM$GB_EVT_PRSN
                                                        .EXTRN  MOM$GB_EVT_PSER
```

```
                                                        .EXTRN  SVD$GK_PCNO_ADD
                                                        .EXTRN  SVD$GK_PCNO_SDV
                                                        .EXTRN  SVD$GK_PCNO_CPU
                                                        .EXTRN  SVD$GK_PCNO_STY
                                                        .EXTRN  SVD$GK_PCNO_DAD
                                                        .EXTRN  SVD$GK_PCNO_DCT
                                                        .EXTRN  SVD$GK_PCNO_IHO
                                                        .EXTRN  SVD$GK_PCNO_NNA
                                                        .EXTRN  SVD$GK_PCNO_SLI
                                                        .EXTRN  SVD$GK_PCNO_SPA
                                                        .EXTRN  SVD$GK_PCNO_HWA
                                                        .EXTRN  SVD$GK_PCNO_SNV
                                                        .EXTRN  SVD$GK_PCNO_LOA
                                                        .EXTRN  SVD$GK_PCNO_SLO
                                                        .EXTRN  SVD$GK_PCNO_TLO
                                                        .EXTRN  SVD$GK_PCNO_DFL
                                                        .EXTRN  SVD$GK_PCNO_SID
                                                        .EXTRN  SVD$GK_PCNO_DUM
                                                        .EXTRN  SVD$GK_PCNO_SDU
                                                        .EXTRN  SVD$GK_PCNO_$HNA
                                                        .EXTRN  SVD$GK_PCNO_$HHW
                                                        .EXTRN  SVD$GK_PCNO_$FTY
                                                        .EXTRN  SVD$GK_PCNO_PHA
                                                        .EXTRN  SVD$GK_PCNO_$DA
                                                        .EXTRN  SVD$GK_PCNO_LPC
                                                        .EXTRN  SVD$GK_PCNO_LPL
                                                        .EXTRN  SVD$GK_PCNO_LPD
                                                        .EXTRN  SVD$GK_PCNO_LPH
                                                        .EXTRN  SVD$GK_PCNO_LPA
                                                        .EXTRN  SVD$GK_PCNO_LPN
                                                        .EXTRN  SVD$GK_PCNO_$LNA
                                                        .EXTRN  SVD$GK_PCNO_$LNH
                                                        .EXTRN  SVD$GK_PCNO_LAN
                                                        .EXTRN  SVD$GK_PCNO_$LNN
                                                        .EXTRN  SVD$GK_PCNO_$LAH
                                                        .EXTRN  SVD$GK_PCLI_STI
                                                        .EXTRN  SVD$C_ENTRY_COUNT
                                                        .EXTRN  MOM$GQ_NETNAMDSC
                                                        .EXTRN  MOM$GW_ACP_CHAN
                                                        .EXTRN  MOM$_QIOBFOVF, MOM$DEBUG_QIO
                                                        .EXTRN  MOM$BLD_REPLY

                                                        .PSECT  $CODE$,NOWRT,2

                                  01FC 00000            .ENTRY  MOM$BUILD_P2, Save R2,R3,R4,R5,R6,R7,R8     ; 0097
                58 00000000G   00   9E 00002            MOVAB   MOM$AB_MSGBLOCK, R8                         ;
                5E             04   C2 00009            SUBL2   #4, SP                                      ;
                50             04   D0 0000C            MOVL    #4, COUNT                                   ; 0156
                52       04    AC   D0 0000F            MOVL    LEN1, R2                                    ; 0157
FFFFFFFE        8F             52   D1 00013            CMPL    R2, #-2                                     ; 0159
                               05   12 0001A            BNEQ    1$                                          ;
                50             02   C0 0001C            ADDL2   #2, COUNT                                   ;
                               19   11 0001F            BRB     3$                                          ;
                               52   D5 00021  1$:       TSTL    R2                                          ; 0160
                               05   12 00023            BNEQ    2$                                          ;
                50             04   C0 00025            ADDL2   #4, COUNT                                   ;
                               10   11 00028            BRB     3$                                          ;
```

```
                               0E   15 0002A 2$:   BLEQ    3$                                         ; 0161
      000000FF   8F            52   D1 0002C       CMPL    R2, #255                                   ;
                               05   14 00033       BGTR    3$                                         ;
            50        02 A240       9E 00035       MOVAB   2(R2)[COUNT], COUNT                        ;
            57        0C   AC       D0 0003A 3$:   MOVL    LEN2, R7                                   ; 0164
      FFFFFFFE   8F            57   D1 0003E       CMPL    R7, #-2                                    ; 0166
                               05   12 00045       BNEQ    4$                                         ;
            50                 02   C0 00047       ADDL2   #2, COUNT                                  ;
                               19   11 0004A       BRB     6$                                         ;
                               57   D5 0004C 4$:   TSTL    R7                                         ; 0167
                               05   12 0004E       BNEQ    5$                                         ;
            50                 04   C0 00050       ADDL2   #4, COUNT                                  ;
                               10   11 00053       BRB     6$                                         ;
                               0E   15 00055 5$:   BLEQ    6$                                         ; 0168
      000000FF   8F            57   D1 00057       CMPL    R7, #255                                   ;
                               05   14 0005E       BGTR    6$                                         ;
            50        02 A740       9E 00060       MOVAB   2(R7)[COUNT], COUNT                        ;
            50        40   A0       9E 00065 6$:   MOVAB   64(R0), COUNT                              ; 0171
            56        14   AC       D0 00069       MOVL    P2DSC, R6                                  ; 0172
            66                 50   D1 0006D       CMPL    COUNT, (R6)                                ;
                               2F   15 00070       BLEQ    7$                                         ;
            68                 04   D0 00072       MOVL    #4, MOM$AB_MSGBLOCK                        ; 0177
         04   A8               05   8E 00075       MNEGB   #5, MOM$AB_MSGBLOCK+4                      ; 0178
         0C   A8  00000000G    8F   D0 00079       MOVL    #MOM$_QIOBFOVF, MOM$AB_MSGBLOCK+12         ; 0179
                      4100     8F   BB 00081       PUSHR   #^M<R8,SP>                                 ; 0180
      00000000G  00            02   FB 00085       CALLS   #2, MOM$BLD_REPLY                          ;
                               6E   DD 0008C       PUSHL   MSGSIZE                                    ; 0181
                 00000000G 00       9F 0008E       PUSHAB  MOM$AB_NICE_XMIT_BUF                       ;
                 02070000  8F       DD 00094       PUSHL   #34013184                                  ;
      00000000G  00            03   FB 0009A       CALLS   #3, LIB$SIGNAL                             ;
  53     04   A6               04   C1 000A1 7$:   ADDL3   #4, 4(R6), PTR                             ; 0184
      FFFFFFFE   8F            52   D1 000A6       CMPL    R2, #-2                                    ; 0191
                               06   12 000AD       BNEQ    8$                                         ;
            83        08   AC       B0 000AF       MOVW    ADR1, (PTR)+                               ;
                               1E   11 000B3       BRB     10$                                        ;
                               52   D5 000B5 8$:   TSTL    R2                                         ; 0192
                               06   12 000B7       BNEQ    9$                                         ;
            83        08   AC       D0 000B9       MOVL    ADR1, (PTR)+                               ;
                               14   11 000BD       BRB     10$                                        ;
                               12   15 000BF 9$:   BLEQ    10$                                        ; 0193
      000000FF   8F            52   D1 000C1       CMPL    R2, #255                                   ;
                               09   14 000C8       BGTR    10$                                        ;
            83        04   AC       B0 000CA       MOVW    LEN1, (PTR)+                               ; 0195
  63     08   BC               52   28 000CE       MOVC3   R2, @ADR1, (PTR)                           ; 0197
      FFFFFFFE   8F            57   D1 000D3 10$:  CMPL    R7, #-2                                    ; 0206
                               06   12 000DA       BNEQ    11$                                        ;
            83        10   AC       B0 000DC       MOVW    ADR2, (PTR)+                               ;
                               1E   11 000E0       BRB     13$                                        ;
                               57   D5 000E2 11$:  TSTL    R7                                         ; 0207
                               06   12 000E4       BNEQ    12$                                        ;
            83        10   AC       D0 000E6       MOVL    ADR2, (PTR)+                               ;
                               14   11 000EA       BRB     13$                                        ;
                               12   15 000EC 12$:  BLEQ    13$                                        ; 0208
      000000FF   8F            57   D1 000EE       CMPL    R7, #255                                   ;
                               09   14 000F5       BGTR    13$                                        ;
            83        0C   AC       B0 000F7       MOVW    LEN2, (PTR)+                               ; 0210
  63     10   BC               57   28 000FB       MOVC3   R7, @ADR2, (PTR)                           ; 0212
```

```
                63 00000000'  00     0040   8F  28 00100 13$:    MOVC3   #64, COLLATE_START_VALUE, (PTR)          ; 0221
                              50       18   AC  D0 0010A         MOVL    RESDSC, R0                             ; 0225
                60            53       04   A6  C3 0010E         SUBL3   4(R6), PTR, (R0)                       ;
                        04    A0       04   A6  D0 00113         MOVL    4(R6), 4(R0)                           ; 0226
                                                 04 00118         RET                                            ; 0228

; Routine Size:  281 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  232          0229  1 %SBTTL 'mom$netacp_qio  General network QIO routine'
;  233          0230  1 GLOBAL ROUTINE mom$netacp_qio (nfbdsc, p2, p3, bufdsc) =
;  234          0231  1
;  235          0232  1 !++
;  236          0233  1 ! FUNCTIONAL DESCRIPTION:
;  237          0234  1 !
;  238          0235  1 !       This routine issues QIO function requests to NETACP to perform
;  239          0236  1 !       volatile data base operations.
;  240          0237  1 !
;  241          0238  1 ! FORMAL PARAMETERS:
;  242          0239  1 !
;  243          0240  1 !       NFBDSC          Descriptor of NFB data.
;  244          0241  1 !       P2              Descriptor of P2 data.
;  245          0242  1 !       P3              Address of word to contain resulting length.
;  246          0243  1 !       BUFDSC          Descriptor of data buffer
;  247          0244  1 !
;  248          0245  1 ! IMPLICIT INPUTS:
;  249          0246  1 !
;  250          0247  1 !       MOM$GW_ACP_CHAN Channel assigned to the command process link.
;  251          0248  1 !
;  252          0249  1 ! IMPLICIT OUTPUTS:
;  253          0250  1 !
;  254          0251  1 !       NONE
;  255          0252  1 !
;  256          0253  1 ! ROUTINE VALUE:
;  257          0254  1 ! COMPLETION CODES:
;  258          0255  1 !
;  259          0256  1 !       This routine returns an MOM status code that has been mapped from
;  260          0257  1 !       the QIO status code.
;  261          0258  1 !
;  262          0259  1 ! SIDE EFFECTS:
;  263          0260  1 !
;  264          0261  1 !       NONE
;  265          0262  1 !
;  266          0263  1 !--
;  267          0264  1
;  268          0265  2 BEGIN
;  269          0266  2
;  270          0267  2 MAP
;  271          0268  2     nfbdsc : REF VECTOR,
;  272          0269  2     bufdsc : REF VECTOR;
;  273          0270  2
;  274          0271  2 LOCAL
;  275          0272  2     iosb     : $iosb,                   ! I/O status block
;  276          0273  2     database,                           ! Database ID
;  277          0274  2     status;                             ! Temporary return status
;  278          0275  2 !
;  279          0276  2 ! If it hasn't already been done, establish channel to NETACP for QIO control
;  280          0277  2 ! functions. The channel is to NET:, the pseudo device to which volatile
;  281          0278  2 ! database commands are issued.  Doing the assing here allows NCP commands to
;  282          0279  2 ! the permanent data base to be processed even if NETACP is not mounted.
;  283          0280  2 !
;  284          0281  2 status = ss$_normal;
;  285          0282  2 IF .mom$gw_acp_chan EQL 0 THEN
;  286        P 0283  2     status = $ASSIGN(DEVNAM = mom$gq_netnamdsc,
;  287          0284  2                      CHAN = mom$gw_acp_chan);
;  288          0285  2 IF .status THEN
```

```
;  289          0286  3     BEGIN
;  290          0287  3     !
;  291          0288  3     ! Issue the QIO.
;  292          0289  3     !
;  293        P 0290  3     status = $QIOW (CHAN = .mom$gw_acp_chan,     ! Channel
;  294        P 0291  3                     FUNC = io$_acpcontrol,       ! Function
;  295        P 0292  3                     IOSB = iosb,                 ! I/O status block
;  296        P 0293  3                     P1   = .nfbdsc,              ! P1 descriptor (NFB)
;  297        P 0294  3                     P2   = .p2,                  ! P2 descriptor (component id)
;  298        P 0295  3                     P3   = .p3,                  ! Address for resulting length
;  299          0296  3                     P4   = .bufdsc);             ! P4 (data buffer) descriptor
;  300          0297  3     !
;  301          0298  3     ! Log the QIO function.
;  302          0299  3     !
;  303          0300  3     mom$debug_qio (dbg$c_acpqio,                 ! Log type code
;  304          0301  3                    .status,                      ! QIO status value
;  305          0302  3                    iosb,                         ! Address of I/O status block
;  306          0303  3                    .nfbdsc,                      ! NFB descriptor
;  307          0304  3                    .p2,                          ! P2 descriptor
;  308          0305  3                    .p3,                          ! Address of P3 word
;  309          0306  3                    .bufdsc,                      ! Data buffer descriptor
;  310          0307  3                    $ASCID('SET, SHOW, or CLEAR NETACPs database'));
;  311          0308  3     !
;  312          0309  3     ! Map the operation status into an MOM code.
;  313          0310  3     !
;  314          0311  3     database = .bblock [.nfbdsc [1], nfb$b_database];
;  315          0312  2     END;
;  316          0313  2 status = mom$mapqioerror (.database, .status, iosb);
;  317          0314  2 !
;  318          0315  2 ! Return the mapped status code.
;  319          0316  2 !
;  320          0317  2 RETURN .status
;  321          0318  2
;  322          0319  1 END;                                    ! End of mom$netqio


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

43 20 72 6F 20 2C 57 4F 48 53 20 2C 54 45 53 00000 P.AAB:    .ASCII  \SET, SHOW, or CLEAR NETACPs database\      ;
61 64 20 73 50 43 41 54 45 4E 20 52 41 45 4C 0000F                                                              ;
                           65 73 61 62 61 74 0001E                                                              ;
                                          00000024  00024 P.AAA:    .LONG   36                                  ;
                                          00000000' 00028           .ADDRESS P.AAB                              ;

                                                                .EXTRN  SYS$ASSIGN, SYS$QIOW

                                                                .PSECT  $CODE$,NOWRT,2

                                             001C 00000         .ENTRY  MOM$NETACP_QIO, Save R2,R3,R4           ; 0230
                    54 00000000G   00    9E 00002         MOVAB   MOM$GW_ACP_CHAN, R4                         ;
                    5E             08    C2 00009         SUBL2   #8, SP                                      ;
                    53             01    D0 0000C         MOVL    #1, STATUS                                  ; 0281
                                   64    D5 0000F         TSTL    MOM$GW_ACP_CHAN                             ; 0282
                                   14    12 00011         BNEQ    1$                                          ;
                                   7E    7C 00013         CLRQ    -(SP)                                       ; 0284
                                   54    DD 00015         PUSHL   R4                                          ;
```

```
                    00000000G   00  9F  00017          PUSHAB  MOM$GQ_NETNAMDSC                                ;
          00000000G   00            04  FB  0001D          CALLS   #4, SYS$ASSIGN                              ;
                      53            50  D0  00024          MOVL    R0, STATUS                                  ;
                      49            53  E9  00027 1$:      BLBC    STATUS, 2$                                  ; 0285
                                    7E  7C  0002A          CLRQ    -(SP)                                       ; 0296
                      7E      0C    AC  7D  0002C          MOVQ    P3, -(SP)                                   ;
                              08    AC  DD  00030          PUSHL   P2                                          ;
                      52      04    AC  D0  00033          MOVL    NFBDSC, R2                                  ;
                                    52  DD  00037          PUSHL   R2                                          ;
                                    7E  7C  00039          CLRQ    -(SP)                                       ;
                              20    AE  9F  0003B          PUSHAB  IOSB                                        ;
                                    38  DD  0003E          PUSHL   #56                                         ;
                                    64  DD  00040          PUSHL   MOM$GW_ACP_CHAN                             ;
                                    7E  D4  00042          CLRL    -(SP)                                       ;
          00000000G   00            0C  FB  00044          CALLS   #12, SYS$QIOW                               ;
                      53            50  D0  0004B          MOVL    R0, STATUS                                  ;
                    00000000'   00  9F  0004E          PUSHAB  P.AAA                                       ; 0307
                      7E      0C    AC  7D  00054          MOVQ    P3, -(SP)                                   ; 0305
                              08    AC  DD  00058          PUSHL   P2                                          ; 0304
                                    52  DD  0005B          PUSHL   R2                                          ; 0303
                              14    AE  9F  0005D          PUSHAB  IOSB                                        ; 0300
                                    53  DD  00060          PUSHL   STATUS                                      ; 0301
                                    04  DD  00062          PUSHL   #4                                          ; 0300
          00000000G   00            08  FB  00064          CALLS   #8, MOM$DEBUG_QIO                           ;
                      50      04    A2  D0  0006B          MOVL    4(R2), R0                                   ; 0311
                      50      02    A0  9A  0006F          MOVZBL  2(R0), DATABASE                             ;
                            4009    8F  BB  00073 2$:      PUSHR   #^M<R0,R3,SP>                               ; 0313
          00000000V   00            03  FB  00077          CALLS   #3, MOM$MAPQIOERROR                         ;
                      53            50  D0  0007E          MOVL    R0, STATUS                                  ;
                                        04  00081          RET                                                 ; 0319

; Routine Size:  130 bytes,    Routine Base:  $CODE$ + 0119
```

```
;   324     0320  1 %SBTTL 'mom$mapqioerror      Map QIO error to MOM error'
;   325     0321  1 GLOBAL ROUTINE mom$mapqioerror (database, qiostatus, iosb) =
;   326     0322  1
;   327     0323  1 !++
;   328     0324  1 ! FUNCTIONAL DESCRIPTION:
;   329     0325  1 !
;   330     0326  1 !       This routine translates QIO errors into network management
;   331     0327  1 !       errors and makes the appropriate entries in the message
;   332     0328  1 !       block.
;   333     0329  1 !
;   334     0330  1 ! FORMAL PARAMETERS:
;   335     0331  1 !
;   336     0332  1 !       DATABASE        Database ID
;   337     0333  1 !       QIOSTATUS       QIO status return.
;   338     0334  1 !       IOSB            Address of I/O status block.
;   339     0335  1 !
;   340     0336  1 ! IMPLICIT INPUTS:
;   341     0337  1 !
;   342     0338  1 !       NONE
;   343     0339  1 !
;   344     0340  1 ! IMPLICIT OUTPUTS:
;   345     0341  1 !
;   346     0342  1 !       MOM$AB_MSGBLOCK contains the appropriate error code and detail
;   347     0343  1 !       if applicable.
;   348     0344  1 !
;   349     0345  1 ! ROUTINE VALUE:
;   350     0346  1 ! COMPLETION CODES:
;   351     0347  1 !
;   352     0348  1 !       The return status is the MOM error that corresponds to the QIO error.
;   353     0349  1 !
;   354     0350  1 ! SIDE EFFECTS:
;   355     0351  1 !
;   356     0352  1 !       NONE
;   357     0353  1 !
;   358     0354  1 !--
;   359     0355  1
;   360     0356  2 BEGIN
;   361     0357  2
;   362     0358  2 MAP
;   363     0359  2     iosb : REF $iosb;
;   364     0360  2
;   365     0361  2 LOCAL
;   366     0362  2     code   : BYTE,                          ! NICE status code
;   367     0363  2     detail : WORD,                          ! NICE detail code
;   368     0364  2     flags,                                  ! Message flags
;   369     0365  2     status,                                 ! Return status
;   370     0366  2     text;                                   ! Optional text code
;   371     0367  2 !
;   372     0368  2 ! Set up the default message information.
;   373     0369  2 !
;   374     0370  2 code = nma$c_sts_ope;                       ! Management program error
;   375     0371  2 detail = -1;                                ! No detail
;   376     0372  2 flags = msb$m_det_fld;                      ! Detail flag
;   377     0373  2 !
;   378     0374  2 ! Check the QIO status and the I/O status block.
;   379     0375  2 !
;   380     0376  2 IF NOT .qiostatus THEN
```

```
;  381          0377  3     BEGIN
;  382          0378  !
;  383          0379  ! The QIO was in error.  This indicates a program or a system error.
;  384          0380  !
;  385          0381        text = .qiostatus;                  ! Use system message as optional text
;  386          0382        flags = .flags OR                   ! Default flags
;  387          0383                msb$m_msg_fld;              !  and optional text
;  388          0384        status = failure;                   ! Return status
;  389          0385
;  390          0386        END
;  391          0387  ELSE
;  392          0388        BEGIN
;  393          0389        !
;  394          0390        ! The QIO status was successful so check the I/O status block.
;  395          0391        ! If it indicates success the just return.  Otherwise, attempt to map
;  396          0392        ! the error code to an MOM error code.
;  397          0393        !
;  398          0394        IF .iosb [ios$w_status] THEN
;  399          0395            RETURN success;
;  400          0396
;  401          0397        SELECTONE .iosb [ios$w_status] OF
;  402          0398            SET
;  403          0399            [ss$_insfarg]:                          ! Missing parameter
;  404          0400  4             BEGIN
;  405          0401  4             code = nma$c_sts_pms;
;  406          0402  4             detail = mom_mapparamid (.iosb [ios$l_info]);
;  407          0403  4             status = failure;
;  408          0404  4             END;
;  409          0405
;  410          0406            [ss$_badparam,                          ! Parameter value error
;  411          0407            ss$_devactive]:
;  412          0408  4             BEGIN
;  413          0409  4             code = nma$c_sts_pva;
;  414          0410  4             detail = mom_mapparamid (.iosb [ios$l_info]);
;  415          0411  4             status = failure;
;  416          0412  3             END;
;  417          0413  3
;  418          0414  3         [ss$_writlck]:                          ! Component in wrong state
;  419          0415  4             BEGIN
;  420          0416  4             code = nma$c_sts_sta;
;  421          0417  4             detail = mom_mapentity (.database);
;  422          0418  4             status = failure;
;  423          0419  3             END;
;  424          0420  3
;  425          0421  3         [ss$_insfmem]:                          ! No room for new entry
;  426          0422  4             BEGIN
;  427          0423  4             code = nma$c_sts_roo;
;  428          0424  4             status = failure;
;  429          0425  3             END;
;  430          0426  3
;  431          0427  3         [ss$_endoffile]:                        ! Unrecognized component
;  432          0428  4             BEGIN
;  433          0429  4             code = nma$c_sts_cmp;
;  434          0430  4             detail = mom_mapentity (.database);
;  435          0431  4             status = nma$c_sts_cmp;
;  436          0432  3             END;
;  437          0433  3
```

```
;   438         0434  3          [ss$_nopriv]:                               ! Privilege violation
;   439         0435  4              BEGIN
;   440         0436  4              code = nma$c_sts_pri;
;   441         0437  4              status = failure;
;   442         0438  3              END;
;   443         0439  3
;   444         0440  3          [ss$_nosuchdev]:                            ! No such device
;   445         0441  4              BEGIN
;   446         0442  4              code = nma$c_sts_cmp;
;   447         0443  4              detail = mom_mapentity (.database);
;   448         0444  4              text = .iosb [ios$w_status];
;   449         0445  4              flags = .flags OR msb$m_msg_fld;
;   450         0446  4              status = failure;
;   451         0447  3              END;
;   452         0448  3
;   453         0449  3          [ss$_devinact]:                             ! Device inactive
;   454         0450  4              BEGIN
;   455         0451  4              code = nma$c_sts_sta;
;   456         0452  4              detail = mom_mapparamid (.iosb [ios$l_info]);
;   457         0453  4              text = .iosb [ios$w_status];
;   458         0454  4              flags = .flags OR msb$m_msg_fld;
;   459         0455  4              status = failure;
;   460         0456  3              END;
;   461         0457  3
;   462         0458  3          [ss$_ivdevnam]:                             ! Invalid device name.
;   463         0459  4              BEGIN
;   464         0460  4              code = nma$c_sts_ide;
;   465         0461  4              detail = mom_mapentity (.database);
;   466         0462  4              text = .iosb [ios$w_status];
;   467         0463  4              flags = .flags OR msb$m_msg_fld;
;   468         0464  4              status = failure;
;   469         0465  3              END;
;   470         0466  3
;   471         0467  3          [ss$_nolicense]:                            ! Customer doesn't have a
;   472         0468  4              BEGIN
;   473         0469  4              code = nma$c_sts_ope;
;   474         0470  4              text = .iosb [ios$w_status];
;   475         0471  4              flags = .flags OR msb$m_msg_fld;
;   476         0472  4              status = failure;
;   477         0473  3              END;
;   478         0474  3
;   479         0475  3          [OTHERWISE]:                                ! Operation failure
;   480         0476  4              BEGIN
;   481         0477  4              code = nma$c_sts_ope;
;   482         0478  4              detail = .iosb [ios$l_info];
;   483         0479  4              text = .iosb [ios$w_status];
;   484         0480  4              flags = .flags OR
;   485         0481  4                      msb$m_msg_fld;
;   486         0482  4              status = failure;
;   487         0483  3              END;
;   488         0484  3
;   489         0485  3          TES;
;   490         0486  2      END;
;   491         0487  2  !
;   492         0488  2  ! Set up the message information.
;   493         0489  2  !
;   494         0490  2  mom$ab_msgblock [msb$l_flags]  = .flags;
```

```
;  495          0491  2 mom$ab_msgblock [msb$b_code]   = .code;
;  496          0492  2 mom$ab_msgblock [msb$w_detail] = .detail;
;  497          0493  2 mom$ab_msgblock [msb$l_text]   = .text;
;  498          0494  2 !
;  499          0495  2 ! Return the mapped status.
;  500          0496  2 !
;  501          0497  2 RETURN .status
;  502          0498  2
;  503          0499  1 END;                                    ! End of mom$mapqioerror


                                        07FC 00000          .ENTRY  MOM$MAPQIOERROR, Save R2,R3,R4,R5,R6,R7,R8,-; 0321
                                                                    R9,R10
                                5A 00000000V   00    9E 00002       MOVAB   MOM_MAPPARAMID, R10                     ;
                                59 00000000V   00    9E 00009       MOVAB   MOM_MAPENTITY, R9                       ;
                                58 00000000G   00    9E 00010       MOVAB   MOM$AB_MSGBLOCK, R8                     ;
                                54                   19  8E 00017       MNEGB   #25, CODE                           ; 0370
                                57                   01  AE 0001A       MNEGW   #1, DETAIL                          ; 0371
                                55                   02  D0 0001D       MOVL    #2, FLAGS                           ; 0372
                                07             08    AC  E8 00020       BLBS    QIOSTATUS, 1$                       ; 0376
                                56             08    AC  D0 00024       MOVL    QIOSTATUS, TEXT                     ; 0381
                                                 00BA  31  00028       BRW     19$                                 ; 0382
                                52             0C    AC  D0 0002B 1$:    MOVL    IOSB, R2                            ; 0394
                                04                   62  E9 0002F       BLBC    (R2), 2$                            ;
                                50                   01  D0 00032       MOVL    #1, R0                              ; 0395
                                                         04  00035       RET                                         ;
                          0114  8F                   62  B1 00036 2$:    CMPW    (R2), #276                          ; 0399
                                                     05  12 0003B       BNEQ    3$                                  ;
                                54                   1D  8E 0003D       MNEGB   #29, CODE                           ; 0401
                                                     0F  11 00040       BRB     5$                                  ; 0402
                                14                   62  B1 00042 3$:    CMPW    (R2), #20                           ; 0406
                                                     07  13 00045       BEQL    4$                                  ;
                          02C4  8F                   62  B1 00047       CMPW    (R2), #708                          ;
                                                     0B  12 0004C       BNEQ    6$                                  ;
                                54                   10  8E 0004E 4$:    MNEGB   #16, CODE                           ; 0409
                                             04    A2  DD 00051 5$:    PUSHL   4(R2)                               ; 0410
                                6A                   01  FB 00054       CALLS   #1, MOM_MAPPARAMID                  ;
                                                     10  11 00057       BRB     7$                                  ;
                          025C  8F                   62  B1 00059 6$:    CMPW    (R2), #604                          ; 0414
                                                     0E  12 0005E       BNEQ    8$                                  ;
                                54                   0B  8E 00060       MNEGB   #11, CODE                           ; 0416
                                             04    AC  DD 00063       PUSHL   DATABASE                            ; 0417
                                69                   01  FB 00066       CALLS   #1, MOM_MAPENTITY                   ;
                                57                   50  B0 00069 7$:    MOVW    R0, DETAIL                          ;
                                                     7A  11 0006C       BRB     20$                                 ; 0418
                          0124  8F                   62  B1 0006E 8$:    CMPW    (R2), #292                          ; 0421
                                                     05  12 00073       BNEQ    9$                                  ;
                                54                   14  8E 00075       MNEGB   #20, CODE                           ; 0423
                                                     6E  11 00078       BRB     20$                                 ; 0424
                          0870  8F                   62  B1 0007A 9$:    CMPW    (R2), #2160                         ; 0427
                                                     11  12 0007F       BNEQ    10$                                 ;
                                54                   08  8E 00081       MNEGB   #8, CODE                            ; 0429
                                             04    AC  DD 00084       PUSHL   DATABASE                            ; 0430
                                69                   01  FB 00087       CALLS   #1, MOM_MAPENTITY                   ;
```

```
                                57                      50  B0 0008A          MOVW    R0, DETAIL                              ;
                                53                      08  CE 0008D          MNEGL   #8, STATUS                              ;  0431
                                                        58  11 00090          BRB     21$                                     ;  0397
                                24                      62  B1 00092 10$:     CMPW    (R2), #36                               ;  0434
                                                        05  12 00095          BNEQ    11$                                     ;
                                54                      03  8E 00097          MNEGB   #3, CODE                                ;  0436
                                                        4C  11 0009A          BRB     20$                                     ;  0437
                            0908    8F                  62  B1 0009C 11$:     CMPW    (R2), #2312                             ;  0440
                                                        05  12 000A1          BNEQ    12$                                     ;
                                54                      08  8E 000A3          MNEGB   #8, CODE                                ;  0442
                                                        1C  11 000A6          BRB     14$                                     ;  0443
                            20D4    8F                  62  B1 000A8 12$:     CMPW    (R2), #8404                             ;  0449
                                                        0B  12 000AD          BNEQ    13$                                     ;
                                54                      0B  8E 000AF          MNEGB   #11, CODE                               ;  0451
                                            04          A2  DD 000B2          PUSHL   4(R2)                                   ;  0452
                                6A                      01  FB 000B5          CALLS   #1, MOM_MAPPARAMID                      ;
                                                        10  11 000B8          BRB     15$                                     ;
                            0144    8F                  62  B1 000BA 13$:     CMPW    (R2), #324                              ;  0458
                                                        0E  12 000BF          BNEQ    16$                                     ;
                                54                      09  8E 000C1          MNEGB   #9, CODE                                ;  0460
                                            04          AC  DD 000C4 14$:     PUSHL   DATABASE                                ;  0461
                                69                      01  FB 000C7          CALLS   #1, MOM_MAPENTITY                       ;
                                57                      50  B0 000CA 15$:     MOVW    R0, DETAIL                              ;
                                                        13  11 000CD          BRB     18$                                     ;  0462
                            2194    8F                  62  B1 000CF 16$:     CMPW    (R2), #8596                             ;  0467
                                                        05  12 000D4          BNEQ    17$                                     ;
                                54                      19  8E 000D6          MNEGB   #25, CODE                               ;  0469
                                                        07  11 000D9          BRB     18$                                     ;  0470
                                54                      19  8E 000DB 17$:     MNEGB   #25, CODE                               ;  0477
                                57              04      A2  B0 000DE          MOVW    4(R2), DETAIL                           ;  0478
                                56                      62  3C 000E2 18$:     MOVZWL  (R2), TEXT                              ;  0479
                                55                      04  88 000E5 19$:     BISB2   #4, FLAGS                               ;  0480
                                                        53  D4 000E8 20$:     CLRL    STATUS                                  ;  0482
                                68                      55  D0 000EA 21$:     MOVL    FLAGS, MOM$AB_MSGBLOCK                  ;  0490
                            04  A8                      54  90 000ED          MOVB    CODE, MOM$AB_MSGBLOCK+4                 ;  0491
                            08  A8                      57  B0 000F1          MOVW    DETAIL, MOM$AB_MSGBLOCK+8               ;  0492
                            0C  A8                      56  D0 000F5          MOVL    TEXT, MOM$AB_MSGBLOCK+12                ;  0493
                                50                      53  D0 000F9          MOVL    STATUS, R0                              ;  0497
                                                            04 000FC          RET                                             ;  0499

; Routine Size:  253 bytes,    Routine Base:  $CODE$ + 019B
```

```
;   505          0500  1 %SBTTL 'mom_mapentity      Map NETACP database ID into entity type'
;   506          0501  1 ROUTINE mom_mapentity (database) =
;   507          0502  1
;   508          0503  1 !++
;   509          0504  1 ! FUNCTIONAL DESCRIPTION:
;   510          0505  1 !
;   511          0506  1 !       This routine translates the QIO database ID into a network
;   512          0507  1 !       management entity detail code.
;   513          0508  1 !
;   514          0509  1 ! INPUTS:
;   515          0510  1 !
;   516          0511  1 !       DATABASE        NETACP database ID
;   517          0512  1 !
;   518          0513  1 ! OUTPUTS:
;   519          0514  1 !
;   520          0515  1 !       The return value is the detail code.
;   521          0516  1 !--
;   522          0517  1
;   523          0518  2 BEGIN
;   524          0519  2
;   525          0520  2 LOCAL
;   526          0521  2     detail : WORD;
;   527          0522  2
;   528          0523  2 detail = (
;   529          0524  2     SELECTONE .database OF
;   530          0525  2         SET
;   531          0526  2         [nfb$c_db_pli]: nma$c_ent_lin;
;   532          0527  2
;   533          0528  2         [nfb$c_db_lni,
;   534          0529  2         nfb$c_db_ndi]: nma$c_ent_nod;
;   535          0530  2
;   536          0531  2
;   537          0532  2         [nfb$c_db_cri]: nma$c_ent_cir;
;   538          0533  2
;   539          0534  2         [nfb$c_db_aji]: nma$c_ent_nod;
;   540          0535  2
;   541          0536  2         [OTHERWISE]:    -1;
;   542          0537  2
;   543          0538  2         TES);
;   544          0539  2
;   545          0540  2 RETURN .detail
;   546          0541  2
;   547          0542  1 END;                                ! End of mom_mapentity


                                0000 00000 MOM_MAPENTITY:
                                           .WORD   Save nothing                    ; 0501
                     50    04   AC  D0 00002         MOVL    DATABASE, R0        ; 0524
                     05         50  D1 00006         CMPL    R0, #5              ; 0526
                                05  12 00009         BNEQ    1$                  ;
                     50         01  D0 0000B         MOVL    #1, R0              ;
                                1F  11 0000E         BRB     6$                  ;
                                50  D5 00010 1$:     TSTL    R0                  ; 0528
                                05  15 00012         BLEQ    2$                  ;
```

```
                        02          50  D1 00014          CMPL    R0, #2                                  ;
                                    0F  15 00017          BLEQ    4$                                      ;
                        04          50  D1 00019 2$:      CMPL    R0, #4                                  ; 0532
                                    05  12 0001C          BNEQ    3$                                      ;
                        50          03  D0 0001E          MOVL    #3, R0                                  ;
                                    0C  11 00021          BRB     6$                                      ;
                        13          50  D1 00023 3$:      CMPL    R0, #19                                 ; 0534
                                    04  12 00026          BNEQ    5$                                      ;
                                    50  D4 00028 4$:      CLRL    R0                                      ;
                                    03  11 0002A          BRB     6$                                      ;
                        50          01  CE 0002C 5$:      MNEGL   #1, R0                                  ; 0536
                        51          50  B0 0002F 6$:      MOVW    R0, DETAIL                              ; 0523
                        50          51  3C 00032          MOVZWL  DETAIL, R0                              ; 0540
                                    04  00035            RET                                             ; 0542

; Routine Size:  54 bytes,    Routine Base:  $CODE$ + 0298
```

```
1 %SBTTL 'mom_mapparamid        Map QIO parameter ID into management code'
1 ROUTINE mom_mapparamid (netacp_param_id) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine translates the NETACP QIO parameter ID code into network
1 !       management parameter code.
1 !
1 ! INPUTS:
1 !       NETACP_PARAM_ID         NETACP NFB ID for parameter.  Returned in
1 !                               second longword of IOSB for some ACP QIO
1 !                               errors.
1 !
1 ! OUTPUTS:
1 !
1 !       The return value is the detail code.
1 !--
1
2 BEGIN
2
2 MAP
2     netacp_param_id: BBLOCK;
2
2 INCR svd_index FROM 0 TO svd$c_entry_count DO
3     BEGIN
3     IF .mom$ab_service_data [.svd_index, svd$l_nfb_id]
3                                           EQL .netacp_param_id THEN
3         RETURN .mom$ab_service_data [.svd_index, svd$w_nice_id];
2     END;
2
2 !
2 ! The parameter doesn't map to any NICE parameter.  Simply return
2 ! a null parameter value.
2 !
2 RETURN -1;
2
1 END;                                   ! End of mom_mapparamid


                              0000 00000 MOM_MAPPARAMID:
                                               .WORD    Save nothing                            ; 0544
          50                   01  CE 00002    MNEGL    #1, SVD_INDEX                           ; 0570
                               20  11 00005    BRB      2$
    51    50 00000089          8F  C5 00007 1$: MULL3   #137, SVD_INDEX, R1                     ; 0569
             00000000G0041     9F 0000F        PUSHAB   MOM$AB_SERVICE_DATA[R1]                 ; 0570
       04 AC                   9E  D1 00016    CMPL     @(SP)+, NETACP_PARAM_ID                 ;
                               0B  12 0001A    BNEQ     2$                                      ;
             00000000G0041     9F 0001C        PUSHAB   MOM$AB_SERVICE_DATA+4[R1]               ; 0571
          50                   9E  3C 00023    MOVZWL   @(SP)+, R0                              ;
                                   04 00026    RET                                              ;
    D8    50 00000000G         8F  F3 00027 2$: AOBLEQ  #SVD$C_ENTRY_COUNT, SVD_INDEX, 1$       ; 0567
          50                   01  CE 0002F    MNEGL    #1, R0                                  ; 0578
                                   04 00032    RET                                              ; 0580
```

```
; Routine Size:  51 bytes,    Routine Base:  $CODE$ + 02CE
```

```
;  588          0581  1 END                                          ! End of module
;  589          0582  1
;  590          0583  0 ELUDOM



                                                                    .EXTRN  LIB$SIGNAL

;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $OWN$                          64  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        769  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                         44  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                 Library Statistics
;
;                                 --------- Symbols ---------      Pages       Processing
;       File                      Total     Loaded    Percent      Mapped      Time
;
;  _$255$DUA28:[MOM.OBJ]MOMLIB.L32;1          194         25          12          21        00:00.1
;  _$255$DUA28:[SHRLIB]NMALIBRY.L32;1         887         12           1          47        00:00.2
;  _$255$DUA28:[SHRLIB]NET.L32;1             1279          7           0          63        00:00.3
;  _$255$DUA28:[SYSLIB]STARLET.L32;1         9776         17           0         581        00:03.1



;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:MOMACPIO/OBJ=OBJ$:MOMACPIO MSRC$:MOMACPIO/UPDATE=(ENH$:MOMACPIO)

; Size:          769 code + 108 data bytes
; Run Time:          00:18.6
; Elapsed Time:      00:45.1
; Lines/CPU Min:      1878
; Lexemes/CPU-Min:  10881
; Memory Used:  157 pages
; Compilation Complete
```

MOMBLDMSG
LIS

MOMINISTA
LIS

MOMLOAD
LIS

MOMACPIO
LIS

MOMDAT
LIS

NMAHEAD
B32

NPADEF
SDL

MOMDEF
SDL

NMATAIL
B32

MOMDDL
B32

NPAMAC
MAR

MOMLIB
LIS

MOMFILTO
LIS