N 15

```
MMM         MMM     OOOOOOOOO     MMM         MMM
MMM         MMM     OOOOOOOOO     MMM         MMM
MMM         MMM     OOOOOOOOO     MMM         MMM
MMMMMM   MMMMMM  OOO         OOO  MMMMMM   MMMMMM
MMMMMM   MMMMMM  OOO         OOO  MMMMMM   MMMMMM
MMMMMM   MMMMMM  OOO         OOO  MMMMMM   MMMMMM
MMM   MMM   MMM  OOO         OOO  MMM   MMM   MMM
MMM   MMM   MMM  OOO         OOO  MMM   MMM   MMM
MMM   MMM   MMM  OOO         OOO  MMM   MMM   MMM
MMM         MMM  OOO         OOO  MMM         MMM
MMM         MMM  OOO         OOO  MMM         MMM
MMM         MMM  OOO         OOO  MMM         MMM
MMM         MMM  OOO         OOO  MMM         MMM
MMM         MMM  OOO         OOO  MMM         MMM
MMM         MMM  OOO         OOO  MMM         MMM
MMM         MMM  OOO         OOO  MMM         MMM
MMM         MMM  OOO         OOO  MMM         MMM
MMM         MMM  OOO         OOO  MMM         MMM
MMM         MMM     OOOOOOOOO     MMM         MMM
MMM         MMM     OOOOOOOOO     MMM         MMM
MMM         MMM     OOOOOOOOO     MMM         MMM
```

B C D E F G H I J K L M N B C D E F G H I J K L M N B C D E F G H I J K L M N B C D E F G H I J K L M N B C D E F G H I

```
**FILE**ID**MOMBLDMSG                                  C  6





MM      MM    OOOOOO    MM      MM  BBBBBBBB    LL          DDDDDDDD    MM      MM    SSSSSSSS    GGGGGGGG
MM      MM    OOOOOO    MM      MM  BBBBBBBB    LL          DDDDDDDD    MM      MM    SSSSSSSS    GGGGGGGG
MMMM  MMMM  OO      OO  MMMM  MMMM  BB      BB  LL          DD      DD  MMMM  MMMM  SS          GG
MMMM  MMMM  OO      OO  MMMM  MMMM  BB      BB  LL          DD      DD  MMMM  MMMM  SS          GG
MM  MM  MM  OO      OO  MM  MM  MM  BB      BB  LL          DD      DD  MM  MM  MM  SS          GG
MM  MM  MM  OO      OO  MM  MM  MM  BB      BB  LL          DD      DD  MM  MM  MM  SS          GG
MM      MM  OO      OO  MM      MM  BBBBBBBB    LL          DD      DD  MM      MM    SSSSSS    GG
MM      MM  OO      OO  MM      MM  BBBBBBBB    LL          DD      DD  MM      MM    SSSSSS    GG
MM      MM  OO      OO  MM      MM  BB      BB  LL          DD      DD  MM      MM          SS  GG  GGGGGG
MM      MM  OO      OO  MM      MM  BB      BB  LL          DD      DD  MM      MM          SS  GG  GGGGGG
MM      MM  OO      OO  MM      MM  BB      BB  LL          DD      DD  MM      MM          SS  GG      GG
MM      MM  OO      OO  MM      MM  BB      BB  LL          DD      DD  MM      MM          SS  GG      GG    ....
MM      MM    OOOOOO    MM      MM  BBBBBBBB    LLLLLLLLLL  DDDDDDDD    MM      MM  SSSSSSSS      GGGGGG      ....
MM      MM    OOOOOO    MM      MM  BBBBBBBB    LLLLLLLLLL  DDDDDDDD    MM      MM  SSSSSSSS      GGGGGG      ....
                                                                                                              ....

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
```

```
0
0 %TITLE 'MOM Network message builder module'
0 MODULE MOMBLDMSG (
0               LANGUAGE (BLISS32),
0               ADDRESSING_MODE (NONEXTERNAL=GENERAL),
0               ADDRESSING_MODE (EXTERNAL=GENERAL),
0               IDENT = 'V04-000'
0               ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:  DECnet-VAX Network Management Maintenance Operations Module (MOM)
1 !
1 ! ABSTRACT:
1 !       This module contains routines to build NICE response messages
1 !       and miscellaneous routines for debugging.
1 !
1 ! ENVIRONMENT:  VAX/VMS Operating System
1 !
1 ! AUTHOR:  Kathy Perko
1 !
1 ! CREATION DATE:  9-Jan-1982
1 !
1 ! MODIFIED BY:
1 !       V03-001 MKP0001         Kathy Perko             29-Jan-1984
1 !               Fix number of bytes returned to NCP for error messages.
1 !
1 !--
1
```

```
1 %SBTTL 'Declarations'
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     mom$bld_reply,
1     mom$getmsg  : NOVALUE,
1     mom$error              : NOVALUE,
1     mom$debug_txt          : NOVALUE,
1     mom$debug_msg          : NOVALUE,
1     mom$debug_qio          : NOVALUE,
1     mom$dump_qio_bufs      : NOVALUE,
1     mom$trnlognum;
1
1 !
1 ! INCLUDE FILES:
1 !
1
1 LIBRARY 'LIB$:MOMLIB.L32';
1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
1 LIBRARY 'SYS$LIBRARY:STARLET.L32';
1
1 !
1 ! EXTERNAL REFERENCES:
1 !
1
1 $mom_externals;
1
1 EXTERNAL
1     mom$gq_proprvmsk : BBLOCK [8];         ! Process privilege mask
1
1 EXTERNAL ROUTINE
1     LIB$CVT_HTB            : ADDRESSING_MODE (GENERAL),
1     LIB$PUT_OUTPUT         : ADDRESSING_MODE (GENERAL);
1
```

```
1 %SBTTL 'mom$bld_reply          Build NICE response message'
1 GLOBAL ROUTINE mom$bld_reply (msgblk, msglen) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine builds a NICE response message based on the
1 !       message segment block.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       MSGBLK          Address of the message segment block (MSB).
1 !       MSGLEN          Address of longword to return the total size of
1 !                       the message that was built.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       MOM$AB_NICE_XMIT_BUF contains the NICE reply message built as described in
1 !       the message segment block.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       The NICE response message is in MOM$AB_NICE_XMIT_BUF.
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     msgblk : REF BBLOCK;
2
2 LOCAL
2     bufcnt  : SIGNED,                               ! Message length counter
2     len     : BYTE,                                 ! Temporary string length
2     in_ptr,                                         ! Input text pointer
2     out_ptr;                                        ! Output message pointer
2 !
2 ! The MSB longword mask determines the message fields that are
2 ! described in the following longwords.  The status code is always
2 ! required.
2 !
2 bufcnt = 0;                                         ! Initialize buffer count
2 out_ptr = mom$ab_nice_xmit_buf;                     ! Get output buffer pointer
2 CH$WCHAR_A (.msgblk [msb$b_code], out_ptr);         ! Add return code
2 bufcnt = .bufcnt + 1;                               ! Increment message count
2 !
2 ! Check for detail field.
2 !
2 IF .msgblk [msb$v_det_fld] THEN
3     BEGIN
3     !
3     ! Move the detail word into the message buffer.
3     !
3     (.out_ptr)<0,16> = .msgblk [msb$w_detail];
3     out_ptr = .out_ptr + 2;
3     END
2 ELSE
```

```
3           BEGIN
3           !
3           ! No detail field is specified so add a minus one to the message.
3           !
3           (.out_ptr)<0,16> = -1;
3           out_ptr = .out_ptr + 2;
2           END;
2
2       bufcnt = .bufcnt + 2;                        ! Add detail length to count
2       !
2       ! Check for message field if there is room in the buffer.
2       !
2       IF .bufcnt LSS mom$k_nice_buf_len THEN
2           IF .msgblk [msb$v_msg_fld] THEN
3               BEGIN
3               mom$getmsg (.msgblk [msb$l_text],
3                           len,
3                           in_ptr);             ! Get system message text
3               !
3               ! If message will not fit in the buffer move the maximum.
3               !
3               IF (.bufcnt + .len) GTR mom$k_nice_buf_len THEN
3                   len = mom$k_nice_buf_len - .bufcnt - 1;
3               !
3               ! Move the count and the entire message into the buffer and the
3               ! length to the total.
3               !
3               CH$WCHAR_A (.len, out_ptr);
3               out_ptr = CH$MOVE (.len,
3                                  .in_ptr,
3                                  .out_ptr);
3               bufcnt = .bufcnt + .len + 1;
3               !
3               ! If a secondary status message is requested, then append a CR/LF
3               ! and the second line of message text to the ASCIC text string in
3               ! the NICE response.
3               !
3               IF .msgblk [msb$v_msg2_fld] THEN          ! If secondary message supplied,
4                   BEGIN
4                   local ascic_count;                    ! Pointer to count byte of string
4                   ascic_count = .out_ptr - .len - 1;
4                   mom$getmsg(.msgblk [msb$l_text2], len, in_ptr);
4                   out_ptr = CH$COPY(2, UPLIT BYTE(13, 10),
4                                  .len, .in_ptr,
4                                  0, mom$k_nice_buf_len - .bufcnt - 1, .out_ptr);
4                   bufcnt = .bufcnt + .len + 2;          ! Increment buffer space used
4                   CH$WCHAR(CH$RCHAR(.ascic_count)+.len+2,
4                               .ascic_count);            ! Increment ASCIC string length
3                   END;
3               END
2           ELSE
3               BEGIN
3               !
3               ! No message field is present so insert zero length.
3               !
3               CH$WCHAR_A (0, out_ptr);
3               bufcnt = .bufcnt + 1;
```

```
2               END;
2           !
2           ! If there is room in the buffer check for the data field.
2           !
2           IF .bufcnt LSS mom$k_nice_buf_len THEN
2               IF .msgblk [msb$v_data_fld]
2               AND (.msgblk [msb$a_data] NEQA 0) THEN
3                   BEGIN
3                   !
3                   ! Data field is ASCID string.
3                   !
3                   BIND
3                       datadsc = msgblk [msb$a_data] : REF VECTOR;
3
3                   in_ptr = .datadsc [1]; ! Get data pointer
3                   len = .datadsc [0];    ! Get length
3                   !
3                   ! If message will not fit in the buffer move the maximum.
3                   !
3                   IF (.bufcnt + .len) LEQ mom$k_nice_buf_len THEN
4                       BEGIN
4                       !
4                       ! Move the data string into the buffer and add length to
4                       ! total.
4                       !
4                       out_ptr = CH$MOVE (.len,
4                                          .in_ptr,
4                                          .out_ptr);
4                       bufcnt = .bufcnt + .len;
3                       END;
2                   END;
2
2 .msglen = .bufcnt;                                              ! Return total message size
2
2 RETURN success                                                  ! Return success
2
1 END;                                                   ! End of mom$bld_reply

                                                                .TITLE  MOMBLDMSG MOM Network message builder module
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                      0A  0D  00000 P.AAA:      .BYTE   13, 10                                        ;

                                                                .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$GQ_PROPRVMSK
                .EXTRN  LIB$CVT_HTB, LIB$PUT_OUTPUT
```

```
                                                                     .PSECT  $CODE$,NOWRT,2

                                             0FFC 00000              .ENTRY  MOM$BLD_REPLY, Save R2,R3,R4,R5,R6,R7,R8,-    ; 0092
                                                                             R9,R10,R11
                      5E               08    C2 00002                SUBL2   #8, SP
                                       56    D4 00005                CLRL    BUFCNT                                        ; 0132
                      52 00000000G     00    9E 00007                MOVAB   MOM$AB_NICE_XMIT_BUF, OUT_PTR                 ; 0133
                      58         04    AC    D0 0000E                MOVL    MSGBLK, R8                                    ; 0134
                      82         04    A8    90 00012                MOVB    4(R8), (OUT_PTR)+
                                       56    D6 00016                INCL    BUFCNT                                        ; 0135
              06      68               01    E1 00018                BBC     #1, (R8), 1$                                  ; 0139
                      62         08    A8    B0 0001C                MOVW    8(R8), (OUT_PTR)                              ; 0144
                                       03    11 00020                BRB     2$                                            ; 0139
                      62               01    AE 00022 1$:            MNEGW   #1, (OUT_PTR)                                 ; 0152
                      52               02    C0 00025 2$:            ADDL2   #2, OUT_PTR                                   ; 0145
                      56               02    C0 00028                ADDL2   #2, BUFCNT                                    ; 0156
            000000C5  8F               56    D1 0002B                CMPL    BUFCNT, #197                                  ; 0160
                                       03    19 00032                BLSS    3$
                                     009C    31 00034                BRW     8$
              03      68               02    E0 00037 3$:            BBS     #2, (R8), 4$                                  ; 0161
                                     0091    31 0003B                BRW     7$
                                       5E    DD 0003E 4$:            PUSHL   SP                                            ; 0163
                                 08    AE    9F 00040                PUSHAB  LEN
                                 0C    A8    DD 00043                PUSHL   12(R8)
            00000000V 00               03    FB 00046                CALLS   #3, MOM$GETMSG
                      59         04    AE    9A 0004D                MOVZBL  LEN, R9                                       ; 0169
                      59               56    C0 00051                ADDL2   BUFCNT, R9
            000000C5  8F               59    D1 00054                CMPL    R9, #197
                                       06    15 0005B                BLEQ    5$
         04   AE         C4  8F        56    83 0005D                SUBB3   BUFCNT, #196, LEN                             ; 0170
                      57         04    AE    9A 00063 5$:            MOVZBL  LEN, R7                                       ; 0175
                      82               57    90 00067                MOVB    R7, (OUT_PTR)+
              62         00   BE       57    28 0006A                MOVC3   R7, @IN_PTR, (OUT_PTR)                        ; 0178
                      52               53    D0 0006F                MOVL    R3, OUT_PTR
                      56    01 A746    9E    00072                   MOVAB   1(R7)[BUFCNT], BUFCNT                         ; 0179
              58      68               03    E1 00077                BBC     #3, (R8), 8$                                  ; 0185
              53      52               57    C3 0007B                SUBL3   R7, OUT_PTR, R3                               ; 0188
                      5A         FF    A3    9E 0007F                MOVAB   -1(R3), ASCIC_COUNT
                                       5E    DD 00083                PUSHL   SP                                            ; 0189
                                 08    AE    9F 00085                PUSHAB  LEN
                                 10    A8    DD 00088                PUSHL   16(R8)
            00000000V 00               03    FB 0008B                CALLS   #3, MOM$GETMSG
                      57         04    AE    9A 00092                MOVZBL  LEN, R7                                       ; 0191
              59 000000C4  8F          56    C3 00096                SUBL3   BUFCNT, #196, R9                              ; 0192
                      5B               52    D0 0009E                MOVL    OUT_PTR, R11
      59      00 00000000'  00         02    2C 000A1                MOVC5   #2, P.AAA, #0, R9, (R11)
                      6B                        000AA
                      0D               18       000AB                BGEQ    6$
                      5B               02    C0 000AD                ADDL2   #2, R11
                      59               02    C2 000B0                SUBL2   #2, R9
      59      00         00   BE       57    2C 000B3                MOVC5   R7, @IN_PTR, #0, R9, (R11)
                      6B                        000B9
                      52               53    D0 000BA 6$:            MOVL    R3, OUT_PTR
                      56    02 A746    9E    000BD                   MOVAB   2(R7)[BUFCNT], BUFCNT                         ; 0193
                      50               6A    9A 000C2                MOVZBL  (ASCIC_COUNT), R0                             ; 0194
                      51    02 A740    9E    000C5                   MOVAB   2(R7)[R0], R1
                      6A               51    90 000CA                MOVB    R1, (ASCIC_COUNT)
```

```
                                                                 K  6
MOMBLDMSG        MOM Network message builder module              16-Sep-1984 02:00:34     VAX-11 Bliss-32 V4.0-742              Page   8
V04-000          mom$bld_reply         Build NICE response message 14-Sep-1984 12:44:29   [MOM.SRC]MOMBLDMSG.B32;1                    (3)

                                          04  11 000CD          BRB     8$                                                     ; 0161
                                          82  94 000CF  7$:     CLRB    (OUT_PTR)+                                             ; 0203
                                          56  D6 000D1          INCL    BUFCNT                                                 ; 0204
                   000000C5    8F         56  D1 000D3  8$:     CMPL    BUFCNT, #197                                           ; 0209
                                          38  18 000DA          BGEQ    9$
              34               68         05  E1 000DC          BBC     #5, (R8), 9$                                           ; 0210
                                    18    A8  D5 000E0          TSTL    24(R8)                                                 ; 0211
                                          2F  13 000E3          BEQL    9$
                               50   18    A8  D0 000E5          MOVL    24(R8), R0                                             ; 0219
                               6E   04    A0  D0 000E9          MOVL    4(R0), IN_PTR
                         04    AE         60  90 000ED          MOVB    (R0), LEN                                              ; 0220
                               59   04    AE  9A 000F1          MOVZBL  LEN, R9                                                ; 0224
                               59         56  C0 000F5          ADDL2   BUFCNT, R9
                   000000C5    8F         59  D1 000F8          CMPL    R9, #197
                                          13  14 000FF          BGTR    9$
                               50   04    AE  9A 00101          MOVZBL  LEN, R0                                                ; 0230
              62         00    BE         50  28 00105          MOVC3   R0, @IN_PTR, (OUT_PTR)                                 ; 0232
                               52         53  D0 0010A          MOVL    R3, OUT_PTR
                               50   04    AE  9A 0010D          MOVZBL  LEN, R0                                                ; 0233
                               56         50  C0 00111          ADDL2   R0, BUFCNT
                         08    BC         56  D0 00114  9$:     MOVL    BUFCNT, @MSGLEN                                        ; 0237
                               50         01  D0 00118          MOVL    #1, R0                                                 ; 0239
                                              04 0011B          RET                                                            ; 0241

; Routine Size:  284 bytes,    Routine Base:  $CODE$ + 0000


;  244          0242  1
```

```
  1 %SBTTL 'mom$getmsg      Get message text from message file'
  1 GLOBAL ROUTINE mom$getmsg (cod, len, ptr) : NOVALUE =
  1
  1 !++
  1 ! FUNCTIONAL DESCRIPTION:
  1 !
  1 !       This routine performs a $GETMSG system service to retrieve the
  1 !       message text for the specified status code from either the system
  1 !       message file, or MOM's message file.
  1 !
  1 ! FORMAL PARAMETERS:
  1 !
  1 !       COD      System error code.
  1 !       LEN      Length of standard message text.
  1 !       PTR      Address of text.
  1 !
  1 ! IMPLICIT OUTPUTS:
  1 !
  1 !       The message text is contained in MSGBUF.  The information
  1 !       in MSGBUF must be copied before a subsequent call to this routine.
  1 !
  1 !--
  1
  2 BEGIN
  2
  2 OWN
  2     msgbuf : BBLOCK [255];                ! Buffer for message text
  2                                           !  (Must be OWN because the text
  2                                           !   has to stay around after the
  2                                           !   return from this routine.)
  2
  2 LOCAL
  2     bufdsc : VECTOR [2],                  ! Message buffer descriptor
  2     reslen : WORD;                        ! Length of text
  2
  2 .len = 0;
  2
  2 bufdsc [0] = 255;       ! Initialize buffer descriptor
  2 bufdsc [1] = msgbuf;
  2 !
  2 ! Retrieve the message text for the specified error code.
  2 !
P 2 $GETMSG (MSGID  = .cod,
P 2         MSGLEN = reslen,
  2         BUFADR = bufdsc);
  2 !
  2 ! Set up return values.
  2 !
  2 .len = .reslen;
  2 .ptr = msgbuf;
  2
  1 END;                                      ! End of MOM$GETMSG


                                                              .PSECT  $OWN$,NOEXE,2
```

```
                                                00000 MSGBUF: .BLKB    255

                                                              .EXTRN   SYS$GETMSG

                                                              .PSECT   $CODE$,NOWRT,2

                                          0004 00000          .ENTRY   MOM$GETMSG, Save R2                     ; 0244
                 52 00000000'    00    9E 00002               MOVAB    MSGBUF, R2                              ;
                 5E              0C    C2 00009               SUBL2    #12, SP                                 ;
                           08    BC    D4 0000C               CLRL     @LEN                                    ; 0278
          04     AE        FF    8F    9A 0000F               MOVZBL   #255, BUFDSC                            ; 0280
          08     AE              62    9E 00014               MOVAB    MSGBUF, BUFDSC+4                        ; 0281
                 7E              0F    7D 00018               MOVQ     #15, -(SP)                              ; 0287
                           0C    AE    9F 0001B               PUSHAB   BUFDSC                                  ;
                           0C    AE    9F 0001E               PUSHAB   RESLEN                                  ;
                           04    AC    DD 00021               PUSHL    COD                                     ;
       00000000G 00              05    FB 00024               CALLS    #5, SYS$GETMSG                          ;
          08     BC              6E    3C 0002B               MOVZWL   RESLEN, @LEN                            ; 0291
          0C     BC              62    9E 0002F               MOVAB    MSGBUF, @PTR                            ; 0292
                                       04 00033               RET                                              ; 0294

; Routine Size:  52 bytes,    Routine Base:  $CODE$ + 011C
```

```
0295  1 %SBTTL 'mom$error          Signal an error message with detail field'
0296  1 GLOBAL ROUTINE mom$error (err, det) : NOVALUE =
0297  1
0298  1 !++
0299  1 ! FUNCTIONAL DESCRIPTION:
0300  1 !        This routine moves an error or status code into the output buffer
0301  1 !        followed by the detail word.
0302  1 !
0303  1 ! FORMAL PARAMETERS:
0304  1 !        ERR                NICE status code to be transmitted (NMA$C_STS_xxx).
0305  1 !        DET                NICE error detail code.
0306  1 !
0307  1 ! SIDE EFFECTS:
0308  1 !
0309  1 !        An error message is signalled to be sent by the condition handler.
0310  1 !
0311  1 !--
0312  1
0313  2 BEGIN
0314  2
0315  2 BUILTIN
0316  2     AP;
0317  2
0318  2 LOCAL
0319  2     count;
0320  2 !
0321  2 ! Move the error code and the detail code into the buffer.
0322  2 !
0323  2 (mom$ab_nice_xmit_buf)<0,8> = .err;
0324  2 IF ..AP GTR 1 THEN
0325  3     BEGIN
0326  3     (mom$ab_nice_xmit_buf + 1)<0,16> = .det;
0327  3     count = 3;
0328  3     END
0329  2 ELSE
0330  2     count = 1;
0331  2 !
0332  2 ! Signal the message.
0333  2 !
0334  2 $signal_msg (mom$ab_nice_xmit_buf, .count);
0335  2
0336  1 END;                                  ! End of mom$error



                            0004 00000           .ENTRY  MOM$ERROR, Save R2                          ; 0296
          52 00000000G  00  9E 00002           MOVAB   MOM$AB_NICE_XMIT_BUF, R2
          62         04 AC  90 00009           MOVB    ERR, MOM$AB_NICE_XMIT_BUF                  ; 0323
          01            6C  D1 0000D           CMPL    (AP), #1                                   ; 0324
                        0A  15 00010           BLEQ    1$
 01       A2         08 AC  B0 00012           MOVW    DET, MOM$AB_NICE_XMIT_BUF+1                ; 0326
          50            03  D0 00017           MOVL    #3, COUNT                                  ; 0327
                        03  11 0001A           BRB     2$                                         ; 0324
          50            01  D0 0001C 1$:       MOVL    #1, COUNT                                  ; 0330
                        50  DD 0001F 2$:       PUSHL   COUNT                                      ; 0334
```

```
                                                          B  7
MOMBLDMSG       MOM Network message builder module       16-Sep-1984 02:00:34    VAX-11 Bliss-32 V4.0-742              Page  12
V04-000         mom$error         Signal an error message with de 14-Sep-1984 12:44:29    [MOM.SRC]MOMBLDMSG.B32;1             (5)

                                                52  DD 00021           PUSHL    R2                                         ;
                                    02070000    8F  DD 00023           PUSHL    #34013184                                  ;
                         00000000G  00          03  FB 00029           CALLS    #3, LIB$SIGNAL                             ;
                                                    04 00030           RET                                                 ; 0336

; Routine Size:  49 bytes,    Routine Base:  $CODE$ + 0150
```

```
  1 %SBTTL 'mom$debug_txt  Print text message'
  1 GLOBAL ROUTINE mom$debug_txt (bitnum, txtdsc) : NOVALUE =
  1
  1 !++
  1 ! FUNCTIONAL DESCRIPTION:
  1 !
  1 !       This routine prints the specified message text to SYS$OUTPUT if
  1 !       the appropriate logging flags are set.
  1 !
  1 ! FORMAL PARAMETERS:
  1 !
  1 !       BITNUM          Bit number of the logging flag.
  1 !       TXTDSC          Descriptor of ASCII text string.
  1 !
  1 ! IMPLICIT INPUTS:
  1 !
  1 !       MOM$GL_LOGMASK  Values of current logging flags.
  1 !
  1 !--
  1
  2 BEGIN
  2
  2 MAP
  2     txtdsc : REF VECTOR;
  2
  2 LITERAL
  2     faosize = 132;
  2
  2 LOCAL
  2     faoprm,
  2     outdsc : VECTOR [2],
  2     faobuf : BBLOCK [faosize];
  2
  2 !
  2 ! If the correct logging flag is set then output the text string.
  2 !
  2 IF .mom$gl_logmask [.bitnum]
  2 THEN
  3     BEGIN
  3     faoprm  = .txtdsc;
  3     outdsc [0] = faosize;
  3     outdsc [1] = faobuf;
P 3     $FAOL (CTRSTR = $ASCID ('*** !AS'),
P 3             OUTLEN = outdsc [0],
P 3             OUTBUF = outdsc,
  3             PRMLST = faoprm);
  3     LIB$PUT_OUTPUT (outdsc);
  3     END;
  2
  1 END;                                ! End of mom$debug_txt


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                 53  41  21  20  2A  2A  2A  00002 P.AAC:  .ASCII  \*** !AS\                      ;
                                             00009         .BLKB   3
```

```
                                     00000007  0000C P.AAB:  .LONG   7                                          :
                                     00000000' 00010         .ADDRESS P.AAC                                     :

                                                             .EXTRN  SYS$FAOL

                                                             .PSECT  $CODE$,NOWRT,2

                                          0000 00000         .ENTRY  MOM$DEBUG_TXT, Save nothing                : 0338
                    5E       FF70  CE   9E 00002             MOVAB   -144(SP), SP                               :
        2D 00000000G  00       04  AC   E1 00007             BBC     BITNUM, MOM$GL_LOGMASK, 1$                 : 0373
                    6E       08    AC   D0 00010             MOVL    TXTDSC, FAOPRM                             : 0376
              F8    AD       84    8F   9A 00014             MOVZBL  #132, OUTDSC                               : 0377
              FC    AD       04    AE   9E 00019             MOVAB   FAOBUF, OUTDSC+4                           : 0378
                                   5E   DD 0001E             PUSHL   SP                                         : 0382
                             F8    AD   9F 00020             PUSHAB  OUTDSC                                     :
                             F8    AD   9F 00023             PUSHAB  OUTDSC                                     :
                       00000000'   00   9F 00026             PUSHAB  P.AAB                                      :
           00000000G  00           04   FB 0002C             CALLS   #4, SYS$FAOL                               :
                             F8    AD   9F 00033             PUSHAB  OUTDSC                                     : 0383
           00000000G  00           01   FB 00036             CALLS   #1, LIB$PUT_OUTPUT                         :
                                        04 0003D 1$:         RET                                                : 0386

; Routine Size:  62 bytes,    Routine Base:  $CODE$ + 0181
```

```
1 %SBTTL 'mom$debug_msg        Print binary message'
1 GLOBAL ROUTINE mom$debug_msg (bitnum, buffer_adr,
1                                      buffer_len, txtdsc) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine dumps binary messages to SYS$OUTPUT.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       BITNUM          Number of the logging flag bit.
1 !       BUFFER_ADR      Address of the message buffer.
1 !       BUFFER_LEN      Length of the message in bytes.
1 !       TXTDSC          Descriptor of text string.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       MOM$GL_LOGMASK  Values of current logging flags.
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     txtdsc : REF VECTOR;
2
2 LITERAL
2     faosiz = 256,                         ! The print buffer.
2     faolst_size = 10,                     ! Size of FAO parameter vector
2     dump_buffer_size = 2000;
2
2 LOCAL
2     faobuf   : VECTOR [faosiz, BYTE],! Print buffer
2     faolst   : VECTOR [faolst_size],      ! List of args to $FAOL
2     outdsc   : VECTOR [2],                ! Descriptor of the output line
2     bytes,                                ! Counter for bytes written
2     ptr:         REF BBLOCK,
2     i,                                    ! index
2     buffer_end,                           ! Address of end of message buffer.
2     dump_buffer :                         ! Buffer from which the data is dumped.
2         BBLOCK [dump_buffer_size];
2
2 !
2 ! If the correct logging flag is not set then just return.
2 !
2 IF NOT .mom$gl_logmask [.bitnum] THEN
2     RETURN;
2 !
2 ! If it's a MOP message, only log it if logging is on for that particular type
2 ! of MOP message.
2 !
2 IF .bitnum EQL dbg$c_mopio THEN
3     BEGIN
3     SELECTONEU .(.buffer_adr)<0,8> OF
3         SET
3         [mop$_fct_mld]:  IF NOT .mom$gl_logmask [dbg$c_mop_mld] THEN RETURN;
```

```
  3             [mop$_fct_rml]:  IF NOT .mom$gl_logmask [dbg$c_mop_rml] THEN RETURN;
  3             [mop$_fct_rmd]:  IF NOT .mom$gl_logmask [dbg$c_mop_rmd] THEN RETURN;
  3             [mop$_fct_mdd]:  IF NOT .mom$gl_logmask [dbg$c_mop_mdd] THEN RETURN;
  3             TES;
  2         END;
  2 !
  2 ! If the string length is nonzero then print it.
  2 !
  2 IF .txtdsc NEQA 0 THEN
  3         BEGIN
  3
  3         outdsc [0] = faosiz;
  3         outdsc [1] = faobuf;
  3
  3         faolst [0] = .txtdsc [0];
  3         faolst [1] = .txtdsc [1];
  3         faolst [2] = .buffer_len;
  3
P 3         $FAOL (CTRSTR = $ASCID (' !AD  (length = !UL bytes)'),
P 3                     OUTLEN = outdsc [0],
P 3                     OUTBUF = outdsc,
  3                     PRMLST = faolst);
  3
  3         LIB$PUT_OUTPUT (outdsc);
  3
  2         END;
  2 !
  2 ! Dumping permanent data base records requires BYPASS privilege because the
  2 ! passwords are displayed.
  2 !
  3 IF (.bitnum EQL dbg$c_fileio)
  2         AND (NOT .mom$gq_proprvmsk [prv$v_bypass]) THEN
  2         RETURN;
  2
  2 !
  2 ! Move the data to be dumped into the dump buffer, filling it with zeros.
  2 ! This ensures that any information past the end of the buffer is printed
  2 ! as zeros.
  2 !
  2 CH$COPY (.buffer_len, .buffer_adr, 0, dump_buffer_size, dump_buffer);
  2 !
  2 ! Dump the buffer contents in hex and ASCII.
  2 !
  2 outdsc [1] = faobuf;
  2 ptr = dump_buffer;
  2 buffer_end = dump_buffer + .buffer_len;
  2 WHILE .ptr LSS .buffer_end DO
  3         BEGIN
  3         outdsc [0] = faosiz;
  3         faolst [0] = .ptr [12,0,32,0];
  3         faolst [1] = .ptr [8,0,32,0];
  3         faolst [2] = .ptr [4,0,32,0];
  3         faolst [3] = .ptr [0,0,32,0];
  3         faolst [4] = 16;
  3         faolst [5] = .ptr;
P 3         $FAOL (CTRSTR = $ASCID ('!XL !XL !XL !XL !_!AF'),
P 3                 OUTLEN = outdsc [0],
;  507        P 0501  3              OUTBUF = outdsc,
;  508          0502  3              PRMLST = faolst);
;  509          0503  3        LIB$PUT_OUTPUT (outdsc);
;  510          0504  3        ptr = .ptr + 16;
;  511          0505  3        END;
;  512          0506  2   !
;  513          0507  2   ! Add a new line.
;  514          0508  2   !
;  515          0509  2   LIB$PUT_OUTPUT ($ASCID (''));
;  516          0510  2
;  517          0511  1   END;                                                ! End of mom$debug_msg


                                                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

3D  20  68  74  67  6E  65  6C  28  20  20  44  41  21  20  00014 P.AAE:  .ASCII  \ !AD  (length = !UL bytes)\              ;
                29  73  65  74  79  62  20  4C  55  21  20  00023                                                          ;
                                                            0002E          .BLKB   2
                                                0000001A    00030 P.AAD:   .LONG   26                                         ;
                                                00000000'   00034          .ADDRESS P.AAE                                     ;
4C  58  21  20  4C  58  21  20  4C  58  21  20  4C  58  21  00038 P.AAG:   .ASCII  \!XL !XL !XL !XL !_!AF\                   ;
                                46  41  21  5F  21  20      00047                                                          ;
                                                            0004D          .BLKB   3
                                                00000015    00050 P.AAF:   .LONG   21                                         ;
                                                00000000'   00054          .ADDRESS P.AAG                                     ;
                                                            00058 P.AAI:   .BLKB   0                                          ;
                                                00000000    00058 P.AAH:   .LONG   0                                          ;
                                                00000000'   0005C          .ADDRESS P.AAI                                     ;


                                                                           .PSECT  $CODE$,NOWRT,2

                                                    03FC  00000            .ENTRY  MOM$DEBUG_MSG, Save R2,R3,R4,R5,R6,R7,R8,R9  ; 0388
                       59 00000000G   00    9E  00002            MOVAB   SYS$FAOL, R9                               ;
                       58 00000000G   00    9E  00009            MOVAB   LIB$PUT_OUTPUT, R8                         ;
                       57 00000000'   00    9E  00010            MOVAB   P.AAD, R7                                  ;
                       56 00000000G   00    9E  00017            MOVAB   MOM$GL_LOGMASK, R6                         ;
                       5E      F700   CE    9E  0001E            MOVAB   -2304(SP), SP                              ;
             01        66        04   AC    E0  00023            BBS     BITNUM, MOM$GL_LOGMASK, 1$                 ; 0433
                                             04  00028            RET                                                ;
             05        04   AC    D1  00029 1$:         CMPL    BITNUM, #5                                 ; 0439
                                 30    12  0002D            BNEQ    5$                                         ;
             50        08   BC    9A  0002F            MOVZBL  @BUFFER_ADR, R0                            ; 0441
             02             50    91  00033            CMPB    R0, #2                                     ; 0443
                                 06    12  00036            BNEQ    2$                                         ;
    22       01        A6        01    E0  00038            BBS     #1, MOM$GL_LOGMASK+1, 5$                   ;
                                       04  0003D            RET                                                ;
    0A                      50    91  0003E 2$:         CMPB    R0, #10                                    ; 0444
                                 06    12  00041            BNEQ    3$                                         ;
    17       01        A6        02    E0  00043            BBS     #2, MOM$GL_LOGMASK+1, 5$                   ;
                                       04  00048            RET                                                ;
    04                      50    91  00049 3$:         CMPB    R0, #4                                     ; 0445
                                 06    12  0004C            BNEQ    4$                                         ;
    0C       01        A6        03    E0  0004E            BBS     #3, MOM$GL_LOGMASK+1, 5$                   ;
                                       04  00053            RET                                                ;
```

```
                                                              H  7
MOMBLDMSG          MOM Network message builder module        16-Sep-1984 02:00:34     VAX-11 Bliss-32 V4.0-742              Page  18
V04-000            mom$debug_msg        Print binary message 14-Sep-1984 12:44:29     [MOM.SRC]MOMBLDMSG.B32;1                   (7)

                                 0E            50  91 00054 4$:      CMPB    R0, #14                                     ; 0446
                                               06  12 00057          BNEQ    5$
                     01      01  A6            04  E0 00059          BBS     #4, MOM$GL_LOGMASK+1, 5$
                                                   04 0005E          RET
                                 50      10    AC  D0 0005F 5$:      MOVL    TXTDSC, R0                                  ; 0452
                                               31  13 00063          BEQL    6$
                          FED0   CD    0100    8F  3C 00065          MOVZWL  #256, OUTDSC                                ; 0455
                          FED4   CD    FF00    CD  9E 0006C          MOVAB   FAOBUF, OUTDSC+4                            ; 0456
                          FED8   CD            60  7D 00073          MOVQ    (R0), FAOLST                                ; 0458
                          FEE0   CD      0C    AC  D0 00078          MOVL    BUFFER_LEN, FAOLST+8                        ; 0460
                                       FED8    CD  9F 0007E          PUSHAB  FAOLST                                      ; 0465
                                       FED0    CD  9F 00082          PUSHAB  OUTDSC
                                       FED0    CD  9F 00086          PUSHAB  OUTDSC
                                               57  DD 0008A          PUSHL   R7
                                 69            04  FB 0008C          CALLS   #4, SYS$FAOL
                                       FED0    CD  9F 0008F          PUSHAB  OUTDSC                                      ; 0467
                                 68            01  FB 00093          CALLS   #1, LIB$PUT_OUTPUT
                                 01      04    AC  D1 00096 6$:      CMPL    BITNUM, #1                                  ; 0474
                                               08  12 0009A          BNEQ    7$
                         6D 00000000G  00      05  E1 0009C          BBC     #5, MOM$GQ_PROPRVMSK+3, 10$                 ; 0475
     07D0     8F         00      08    BC      0C  AC  2C 000A4 7$:  MOVC5   BUFFER_LEN, @BUFFER_ADR, #0, #2000, -       ; 0483
                                 6E                   000AD                  DUMP_BUFFER
                          FED4   CD    FF00    CD  9E 000AE          MOVAB   FAOBUF, OUTDSC+4                            ; 0487
                                 52            6E  9E 000B5          MOVAB   DUMP_BUFFER, PTR                            ; 0488
                                 50            6E  9E 000B8          MOVAB   DUMP_BUFFER, R0                             ; 0489
                     53          50      0C    AC  C1 000BB          ADDL3   BUFFER_LEN, R0, BUFFER_END
                                 53            52  D1 000C0 8$:      CMPL    PTR, BUFFER_END                             ; 0490
                                               46  18 000C3          BGEQ    9$
                          FED0   CD    0100    8F  3C 000C5          MOVZWL  #256, OUTDSC                                ; 0492
                          FED8   CD      0C    A2  D0 000CC          MOVL    12(PTR), FAOLST                             ; 0493
                          FEDC   CD      08    A2  D0 000D2          MOVL    8(PTR), FAOLST+4                            ; 0494
                          FEE0   CD      04    A2  D0 000D8          MOVL    4(PTR), FAOLST+8                            ; 0495
                          FEE4   CD            62  D0 000DE          MOVL    (PTR), FAOLST+12                            ; 0496
                          FEE8   CD            10  D0 000E3          MOVL    #16, FAOLST+16                              ; 0497
                          FEEC   CD            52  D0 000E8          MOVL    PTR, FAOLST+20                              ; 0498
                                       FED8    CD  9F 000ED          PUSHAB  FAOLST                                      ; 0502
                                       FED0    CD  9F 000F1          PUSHAB  OUTDSC
                                       FED0    CD  9F 000F5          PUSHAB  OUTDSC
                                         20    A7  9F 000F9          PUSHAB  P.AAF
                                 69            04  FB 000FC          CALLS   #4, SYS$FAOL
                                       FED0    CD  9F 000FF          PUSHAB  OUTDSC                                      ; 0503
                                 68            01  FB 00103          CALLS   #1, LIB$PUT_OUTPUT
                                 52            10  C0 00106          ADDL2   #16, PTR                                    ; 0504
                                               B5  11 00109          BRB     8$                                          ; 0490
                                         28    A7  9F 0010B 9$:      PUSHAB  P.AAH                                       ; 0509
                                 68            01  FB 0010E          CALLS   #1, LIB$PUT_OUTPUT
                                                   04 00111 10$:     RET                                                 ; 0511

; Routine Size:  274 bytes,    Routine Base:  $CODE$ + 01BF
```

```
   1 %SBTTL 'mom$debug_qio        Print NETACP QIO information'
   1 GLOBAL ROUTINE mom$debug_qio (bitnum, qios, iosb, p1dsc,
   1                                   p2dsc, p3adr, p4dsc, txtdsc) : NOVALUE =
   1
   1 !++
   1 ! FUNCTIONAL DESCRIPTION:
   1 !
   1 !       This routine dumps NETACP QIO information to SYS$OUTPUT.
   1 !
   1 ! FORMAL PARAMETERS:
   1 !
   1 !       BITNUM          Contains the number of the logging flag bit.
   1 !       QIOS            Status of QIO (R0).
   1 !       IOSB            Address of I/O status block.
   1 !       P1DSC           Address of P1 descriptor.
   1 !       P2DSC           Address of P2 descriptor.
   1 !       P3ADR           Address of P3 word.
   1 !       P4DSC           Address of P4 descriptor.
   1 !       TXTDSC          Descriptor of text string.
   1 !
   1 ! IMPLICIT INPUTS:
   1 !
   1 !       MOM$GL_LOGMASK  Values of current logging flags.
   1 !
   1 !--
   1
   2 BEGIN
   2
   2 MAP
   2         iosb   : REF $IOSB,
   2         p1dsc : REF VECTOR,
   2         p2dsc  : REF VECTOR,
   2         p4dsc  : REF VECTOR;
   2
   2 BIND
 P 2         faostr = $ASCID ('R0=!XL IOSB=!XL/!XL P1=!XW/!XL!/',
   2                          'P2=!XW/!XL P3=!XL (!XW) P4=!XW/!XL');
   2
   2 LITERAL
   2         faosiz = 256;                   ! The print buffer
   2
   2 LOCAL
   2         faobuf : VECTOR [faosiz, BYTE], ! Print buffer
   2         faolst : VECTOR [20],           ! List of args to $FAOL
   2         outdsc : VECTOR [2];            ! Descriptor of the output line
   2 !
   2 ! If the correct logging flag is not enabled then just return.
   2 !
   2 IF NOT .mom$gl_logmask [.bitnum]
   2 THEN
   2         RETURN;
   2
   2 !
   2 ! Print header message at beginning of QIO information.
   2 !
   2 IF .txtdsc NEQ 0 THEN
   2     mom$debug_txt (.bitnum, .txtdsc);
```

```
2
2 outdsc [0] = faosiz;
2 outdsc [1] = faobuf;
2 !
2 ! Log the QIO completion status, IOSB, and the values of the QIO
2 ! parameters.
2 !
2 faolst [0] = .qios;
2 IF .iosb NEQ 0 THEN
3     BEGIN
3     faolst [1] = .iosb [0,0,32,0];
3     faolst [2] = .iosb [4,0,32,0];
3     END
2 ELSE
3     BEGIN
3     faolst [1] = 0;
3     faolst [2] = 0;
2     END;
2
2 IF .p1dsc NEQA 0 THEN
3     BEGIN
3     faolst [3] = .p1dsc [0];
3     faolst [4] = .p1dsc [1];
3     END
2 ELSE
3     BEGIN
3     faolst [3] = 0;
3     faolst [4] = 0;
2     END;
2
2 IF .p2dsc NEQA 0
2 THEN
3         BEGIN
3         faolst [5] = .p2dsc [0];
3         faolst [6] = .p2dsc [1];
3         END
2 ELSE
3         BEGIN
3         faolst [5] = 0;
3         faolst [6] = 0;
2         END;
2
2 faolst [7] = .p3adr;
2 IF .p3adr NEQA 0
2 THEN
2         faolst [8] = .(.p3adr)<0,16>
2 ELSE
2         faolst [8] = 0;
2
2 IF .p4dsc NEQA 0
2 THEN
3         BEGIN
3         faolst [9] = .p4dsc [0];
3         faolst [10] = .p4dsc [1];
3         END
2 ELSE
3         BEGIN
```

```
;  633        0626 3            faolst [9] = 0;
;  634        0627 3            faolst [10] = 0;
;  635        0628 2            END;
;  636        0629 2
;  637      P 0630 2    $FAOL (CTRSTR = faostr,
;  638      P 0631 2            OUTLEN = outdsc [0],
;  639      P 0632 2            OUTBUF = outdsc,
;  640        0633 2            PRMLST = faolst);
;  641        0634 2
;  642        0635 2    LIB$PUT_OUTPUT (outdsc);                      ! Write to SYS$OUTPUT
;  643        0636 2
;  644        0637 2    IF NOT .qios
;  645        0638 2    THEN
;  646        0639 2            mom$getmsg (.qios, outdsc [0], outdsc [1])
;  647        0640 2    ELSE
;  648        0641 2        IF .iosb NEQ 0
;  649        0642 2        THEN
;  650        0643 2            mom$getmsg (.iosb [ios$w_status],
;  651        0644 2                        outdsc [0],
;  652        0645 2                        outdsc [1]);
;  653        0646 2
;  654        0647 2    LIB$PUT_OUTPUT (outdsc);                      ! Write to SYS$OUTPUT
;  655        0648 2    !
;  656        0649 2    !
;  657        0650 2    ! Dump the contents of the NFB, the P2 (Key) buffer, and the P4 (Value) buffer.
;  658        0651 2    !
;  659        0652 2    mom$dump_qio_bufs (.bitnum, .p1dsc, .p2dsc, .p4dsc, .p3adr);
;  660        0653 2
;  661        0654 1    END;                                  ! End of mom$debug_qio


                                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

4C 58 21 3D 42 53 4F 49 20 4C 58 21 3D 30 52  00060 P.AAK:  .ASCII  \R0=!XL IOSB=!XL/!XL P1=!XW/!XL!/P2=!XW/!\  ;
4C 58 21 2F 57 58 21 3D 31 50 20 4C 58 21 2F  0006F                                                             ;
               21 2F 57 58 21 3D 32 50 2F 21  0007E                                                             ;
29 57 58 21 28 20 4C 58 21 3D 33 50 20 4C 58  00088         .ASCII  \XL P3=!XL (!XW) P4=!XW/!XL\               ;
            4C 58 21 2F 57 58 21 3D 34 50 20  00097                                                             ;
                                              000A2         .BLKB   2
                                   00000042   000A4 P.AAJ:  .LONG   66                                          ;
                                   00000000'  000A8         .ADDRESS P.AAK                                      ;

                                                    FAOSTR=                 P.AAJ


                                                                          .PSECT  $CODE$,NOWRT,2

                                     00FC 00000         .ENTRY  MOM$DEBUG_QIO, Save R2,R3,R4,R5,R6,R7     ; 0513
                   57 00000000G  00   9E 00002         MOVAB   LIB$PUT_OUTPUT, R7                         ;
                   5E      FEA8  CE   9E 00009         MOVAB   -344(SP), SP                               ;
          01 00000000G  00   04  AC   E0 0000E         BBS     BITNUM, MOM$GL_LOGMASK, 1$                 ; 0560
                                      04 00017         RET                                                ;
                             20  AC   D5 00018 1$:     TSTL    TXTDSC                                     ; 0567
                                 0B   13 0001B         BEQL    2$                                         ;
                             20  AC   DD 0001D         PUSHL   TXTDSC                                     ; 0568
                             04  AC   DD 00020         PUSHL   BITNUM                                     ;
          FE88   CF            02   FB 00023           CALLS   #2, MOM$DEBUG_TXT                      ;
                 6E     0100   8F   3C 00028 2$:       MOVZWL  #256, OUTDSC                           ; 0570
            04   AE       58   AE   9E 0002D           MOVAB   FAOBUF, OUTDSC+4                       ; 0571
            08   AE       08   AC   D0 00032           MOVL    QIOS, FAOLST                           ; 0576
                 55       0C   AC   D0 00037           MOVL    IOSB, R5                               ; 0577
                               56   D4 0003B           CLRL    R6                                     ;
                               55   D5 0003D           TSTL    R5                                     ;
                               08   13 0003F           BEQL    3$                                     ;
                               56   D6 00041           INCL    R6                                     ;
            0C   AE                 65   7D 00043      MOVQ    (R5), FAOLST+4                         ; 0579
                               03   11 00047           BRB     4$                                     ; 0577
                          0C   AE   7C 00049 3$:       CLRQ    FAOLST+4                               ; 0584
                 54       10   AC   D0 0004C 4$:       MOVL    P1DSC, R4                              ; 0588
                               06   13 00050           BEQL    5$                                     ;
            14   AE            64   7D 00052           MOVQ    (R4), FAOLST+12                        ; 0590
                               03   11 00056           BRB     6$                                     ; 0588
                          14   AE   7C 00058 5$:       CLRQ    FAOLST+12                              ; 0595
                 53       14   AC   D0 0005B 6$:       MOVL    P2DSC, R3                              ; 0599
                               06   13 0005F           BEQL    7$                                     ;
            1C   AE            63   7D 00061           MOVQ    (R3), FAOLST+20                        ; 0602
                               03   11 00065           BRB     8$                                     ; 0599
                          1C   AE   7C 00067 7$:       CLRQ    FAOLST+20                              ; 0607
            24   AE       18   AC   D0 0006A 8$:       MOVL    P3ADR, FAOLST+28                       ; 0611
                               07   13 0006F           BEQL    9$                                     ; 0612
            28   AE       18   BC   3C 00071           MOVZWL  @P3ADR, FAOLST+32                      ; 0614
                               03   11 00076           BRB     10$                                    ;
                          28   AE   D4 00078 9$:       CLRL    FAOLST+32                              ; 0616
                 52       1C   AC   D0 0007B 10$:      MOVL    P4DSC, R2                              ; 0618
                               06   13 0007F           BEQL    11$                                    ;
            2C   AE            62   7D 00081           MOVQ    (R2), FAOLST+36                        ; 0621
                               03   11 00085           BRB     12$                                    ; 0618
                          2C   AE   7C 00087 11$:      CLRQ    FAOLST+36                              ; 0626
                          08   AE   9F 0008A 12$:      PUSHAB  FAOLST                                 ; 0633
                          04   AE   9F 0008D           PUSHAB  OUTDSC                                 ;
                          08   AE   9F 00090           PUSHAB  OUTDSC                                 ;
                    00000000'  00   9F 00093           PUSHAB  FAOSTR                                 ;
     00000000G   00            04   FB 00099           CALLS   #4, SYS$FAOL                           ;
                               5E   DD 000A0           PUSHL   SP                                     ; 0635
                 67            01   FB 000A2           CALLS   #1, LIB$PUT_OUTPUT                     ;
                 0B       08   AC   E8 000A5           BLBS    QIOS, 13$                              ; 0637
                          04   AE   9F 000A9           PUSHAB  OUTDSC+4                               ; 0639
                          04   AE   9F 000AC           PUSHAB  OUTDSC                                 ;
                          08   AC   DD 000AF           PUSHL   QIOS                                   ;
                               0C   11 000B2           BRB     14$                                    ;
                 0E            56   E9 000B4 13$:      BLBC    R6, 15$                                ; 0641
                          04   AE   9F 000B7           PUSHAB  OUTDSC+4                               ; 0645
                          04   AE   9F 000BA           PUSHAB  OUTDSC                                 ; 0644
                          7E   65   3C 000BD           MOVZWL  (R5), -(SP)                            ; 0643
          FD86   CF            03   FB 000C0 14$:      CALLS   #3, MOM$GETMSG                         ;
                               5E   DD 000C5 15$:      PUSHL   SP                                     ; 0647
                 67            01   FB 000C7           CALLS   #1, LIB$PUT_OUTPUT                     ;
                          18   AC   DD 000CA           PUSHL   P3ADR                                  ; 0652
                               52   DD 000CD           PUSHL   R2                                     ;
                               53   DD 000CF           PUSHL   R3                                     ;
                               54   DD 000D1           PUSHL   R4                                     ;
                          04   AC   DD 000D3           PUSHL   BITNUM                                 ;
     00000000V   00            05   FB 000D6           CALLS   #5, MOM$DUMP_QIO_BUFS                  ;
```

```
                                                  04 000DD          RET                                                ; 0654

; Routine Size:  222 bytes,    Routine Base:  $CODE$ + 02D1
```

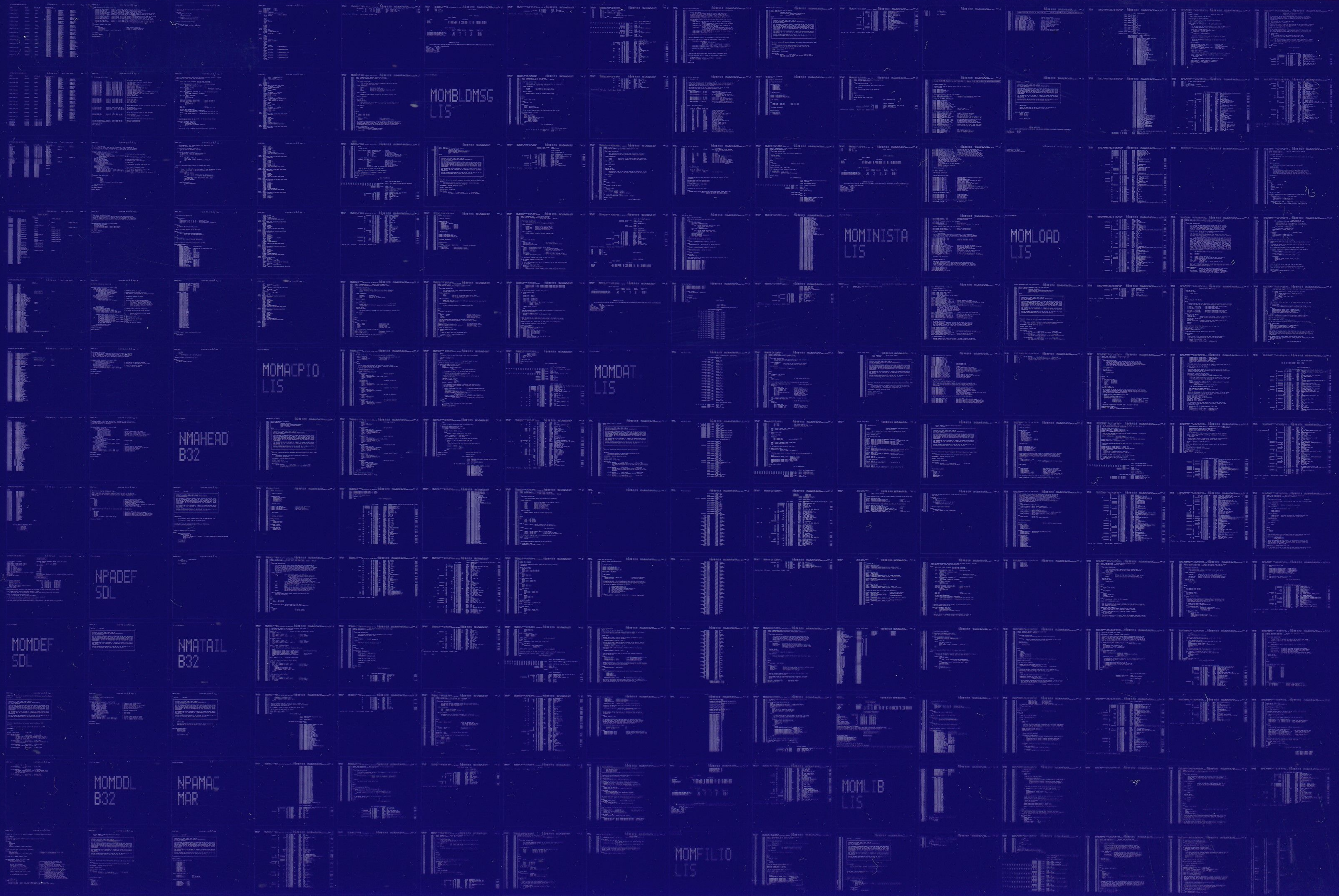
```
1 %SBTTL 'mom$dump_qio_bufs  Dump QIO buffers'
1 GLOBAL ROUTINE mom$dump_qio_bufs (bitnum, p1dsc, p2dsc, p4dsc, p3adr) :
1                                                        NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine dumps the contents of the buffers after a QIO to NETACP.
1 !       The buffers dumped are the NFB, the P2 (Key) buffer, and the
1 !       P4 (Value) buffer.
1 !
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       BITNUM          Contains the number of the logging flag bit.
1 !       P1DSC           Address of P1 descriptor.
1 !       P2DSC           Address of P2 descriptor.
1 !       P4DSC           Address of P4 descriptor.
1 !       P3ADR           Address of P3 word.
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       p4len;          ! Length of P4 buffer
2
2 MAP
2       p1dsc   : REF VECTOR,
2       p2dsc   : REF VECTOR,
2       p4dsc   : REF VECTOR;
2
2 IF .p1dsc NEQ 0 THEN
2     mom$debug_msg ( .bitnum,
2                     .p1dsc [1],
2                     .p1dsc [0],
2                     $ASCID('P1 buffer contents'));
2
2 IF .p2dsc NEQ 0
2 THEN
2     mom$debug_msg ( .bitnum,
2                     .p2dsc [1],
2                     .p2dsc [0],
2                     $ASCID ('P2 buffer contents'));
2
2 IF .p4dsc NEQ 0
2 THEN
3     BEGIN
3     !
3     ! Figure out how much of the P4 buffer to dump.  If it's a
3     ! show, the byte count was returned in P3.  If it's a set,
3     ! the byte count is in the P4 buffer descriptor.
3     !
3     IF .p3adr NEQ 0 THEN
3         IF .(.p3adr)<0,16> GTR mom$k_qio_buf_len THEN
3             p4len = 64
3         ELSE
```

```
;   720          0712  3                 p4len = .(.p3adr)<0,16>
;   721          0713  3          ELSE
;   722          0714  3              p4len = .p4dsc [0];
;   723          0715  3          mom$debug_msg ( .bitnum,
;   724          0716  3                          .p4dsc [1],
;   725          0717  3                          .p4len,
;   726          0718  3                          $ASCID ('P4 buffer contents'));
;   727          0719  2          END;
;   728          0720  1 END;        ! of  mom$dump_qio_bufs


                                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

65  74  6E  6F  63  20  72  65  66  66  75  62  20  31  50  000AC P.AAM:    .ASCII  \P1 buffer contents\                     ;
                                                73  74  6E  000BB
                                                            000BE           .BLKB   2
                                           00000012  000C0 P.AAL:    .LONG   18                                            ;
                                           00000000' 000C4           .ADDRESS P.AAM
65  74  6E  6F  63  20  72  65  66  66  75  62  20  32  50  000C8 P.AAO:    .ASCII  \P2 buffer contents\                     ;
                                                73  74  6E  000D7
                                                            000DA           .BLKB   2
                                           00000012  000DC P.AAN:    .LONG   18                                            ;
                                           00000000' 000E0           .ADDRESS P.AAO
65  74  6E  6F  63  20  72  65  66  66  75  62  20  34  50  000E4 P.AAQ:    .ASCII  \P4 buffer contents\                     ;
                                                73  74  6E  000F3
                                                            000F6           .BLKB   2
                                           00000012  000F8 P.AAP:    .LONG   18                                            ;
                                           00000000' 000FC           .ADDRESS P.AAQ


                                                                                   .PSECT  $CODE$,NOWRT,2

                                              000C 00000           .ENTRY  MOM$DUMP_QIO_BUFS, Save R2,R3                 ; 0656
                          53 00000000'   00   9E 00002           MOVAB   P.AAL, R3                                         ;
                          52       FE03  CF   9E 00009           MOVAB   MOM$DEBUG_MSG, R2                                 ;
                          50         08  AC   D0 0000E           MOVL    P1DSC, R0                                         ; 0687
                                         0D   13 00012           BEQL    1$                                                ;
                                         53   DD 00014           PUSHL   R3                                                ; 0691
                                         60   DD 00016           PUSHL   (R0)                                              ; 0690
                                   04    A0   DD 00018           PUSHL   4(R0)                                             ; 0689
                                   04    AC   DD 0001B           PUSHL   BITNUM                                            ; 0688
                          62             04   FB 0001E           CALLS   #4, MOM$DEBUG_MSG                                 ;
                          50         0C  AC   D0 00021 1$:       MOVL    P2DSC, R0                                         ; 0693
                                         0E   13 00025           BEQL    2$                                                ;
                                   1C    A3   9F 00027           PUSHAB  P.AAN                                             ; 0698
                                         60   DD 0002A           PUSHL   (R0)                                              ; 0697
                                   04    A0   DD 0002C           PUSHL   4(R0)                                             ; 0696
                                   04    AC   DD 0002F           PUSHL   BITNUM                                            ; 0695
                          62             04   FB 00032           CALLS   #4, MOM$DEBUG_MSG                                 ;
                          51         10  AC   D0 00035 2$:       MOVL    P4DSC, R1                                         ; 0700
                                         2A   13 00039           BEQL    6$                                                ;
                                   14    AC   D5 0003B           TSTL    P3ADR                                             ; 0708
                                         14   13 0003E           BEQL    4$                                                ;
                    0200    8F     14    BC   B1 00040           CMPW    @P3ADR, #512                                      ; 0709
                                         06   1B 00046           BLEQU   3$                                                ;
                    50      40   8F  9A 00048          MOVZBL  #64, P4LEN                             ; 0710
                                 09  11 0004C          BRB     5$                                     ;
                    50      14   BC  3C 0004E 3$:      MOVZWL  @P3ADR, P4LEN                          ; 0712
                                 03  11 00052          BRB     5$                                     ; 0709
                    50           61  D0 00054 4$:      MOVL    (R1), P4LEN                            ; 0714
                            38   A3  9F 00057 5$:      PUSHAB  P.AAP                                  ; 0718
                                 50  DD 0005A          PUSHL   P4LEN                                  ; 0717
                            04   A1  DD 0005C          PUSHL   4(R1)                                  ; 0716
                            04   AC  DD 0005F          PUSHL   BITNUM                                 ; 0715
                    62           04  FB 00062          CALLS   #4, MOM$DEBUG_MSG                      ;
                                     04 00065 6$:      RET                                            ; 0720

; Routine Size:  102 bytes,    Routine Base:  $CODE$ + 03AF
;   730        0721  1 %SBTTL 'mom$trnlognum       Translate numeric logical name'
;   731        0722  1 GLOBAL ROUTINE mom$trnlognum (lnmdsc, resadr) =
;   732        0723  1
;   733        0724  1 !++
;   734        0725  1 ! FUNCTIONAL DESCRIPTION:
;   735        0726  1 !
;   736        0727  1 !       This routine translates a logical name and returns the numeric
;   737        0728  1 !       representation of the ASCII hexadecimal number that results.
;   738        0729  1 !
;   739        0730  1 ! FORMAL PARAMETERS:
;   740        0731  1 !
;   741        0732  1 !       LNMDSC          Descriptor of the logical name to be translated.
;   742        0733  1 !       RESADR          Address of longword to contain the numeric value.
;   743        0734  1 !
;   744        0735  1 ! IMPLICIT INPUTS:
;   745        0736  1 !
;   746        0737  1 !       NONE
;   747        0738  1 !
;   748        0739  1 ! IMPLICIT OUTPUTS:
;   749        0740  1 !
;   750        0741  1 !       NONE
;   751        0742  1 !
;   752        0743  1 ! ROUTINE VALUE:
;   753        0744  1 ! COMPLETION CODES:
;   754        0745  1 !
;   755        0746  1 !       Returns error code if the logical name has no translation or the
;   756        0747  1 !       translation is invalid.  The result longword will be set to zero.
;   757        0748  1 !
;   758        0749  1 ! SIDE EFFECTS:
;   759        0750  1 !
;   760        0751  1 !       NONE
;   761        0752  1 !
;   762        0753  1 !--
;   763        0754  1
;   764        0755  2 BEGIN
;   765        0756  2
;   766        0757  2 MAP
;   767        0758  2     lnmdsc : vector;
;   768        0759  2
;   769        0760  2 OWN
;   770        0761  2     ascnum : VECTOR [8, BYTE];
;   771        0762  2
;   772        0763  2 LOCAL
;   773        0764  2     asclen : WORD,
;   774        0765  2     status;
;   775        0766  2
;   776      P 0767  2 status = $TRNLOG (LOGNAM = .lnmdsc,
;   777      P 0768  2                   RSLLEN = asclen,
;   778        0769  2                   RSLBUF = UPLIT (8, ascnum));
;   779        0770  2
;   780        0771  2 IF .status EQL ss$_normal THEN
;   781        0772  2     status = LIB$CVT_HTB (.asclen, ascnum, .resadr);
;   782        0773  2
;   783        0774  2 RETURN .status
;   784        0775  2
;   785        0776  1 END;                               ! End of mom$trnlognum
```

```
                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                          00000008  00100 P.AAR:        .LONG   8                                       ;
                                          00000000' 00104               .ADDRESS ASCNUM                                 ;

                                                                        .PSECT  $OWN$,NOEXE,2

                                                    000FF               .BLKB   1
                                                    00100 ASCNUM:       .BLKB   8

                                                                        .EXTRN  SYS$TRNLOG

                                                                        .PSECT  $CODE$,NOWRT,2

                                               0000 00000               .ENTRY  MOM$TRNLOGNUM, Save nothing             ; 0722
                          5E            04   C2 00002                   SUBL2   #4, SP
                                        7E   7C 00005                   CLRQ    -(SP)                                   ; 0769
                                        7E   D4 00007                   CLRL    -(SP)
                          00000000'     00   9F 00009                   PUSHAB  P.AAR
                                10      AE   9F 0000F                   PUSHAB  ASCLEN
                                04      AC   DD 00012                   PUSHL   LNMDSC
                 00000000G  00          06   FB 00015                   CALLS   #6, SYS$TRNLOG
                            01          50   D1 0001C                   CMPL    STATUS, #1                              ; 0771
                                        14   12 0001F                   BNEQ    1$
                                08      AC   DD 00021                   PUSHL   RESADR                                  ; 0772
                          00000000'     00   9F 00024                   PUSHAB  ASCNUM
                          7E    08      AE   3C 0002A                   MOVZWL  ASCLEN, -(SP)
                 00000000G  00          03   FB 0002E                   CALLS   #3, LIB$CVT_HTB
                                             04 00035 1$:               RET                                             ; 0776

; Routine Size:  54 bytes,    Routine Base:  $CODE$ + 0415


;   786          0777  1
;   787          0778  1
;   788          0779  1
;   789          0780  1 END                                            ! End of module
;   790          0781  1
;   791          0782  0 ELUDOM



                                                                        .EXTRN  LIB$SIGNAL

;                                         PSECT SUMMARY
;
;       Name                          Bytes                          Attributes
;
;   $PLIT$                                264  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                               1099  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                                 264  NOVEC,  WRT,  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       36        18            21         00:00.1
;  _$255$DUA28:[SHRLIB]NMALIBRY.L32;1       887        0         0            47         00:00.2
;  _$255$DUA28:[SYSLIB]STARLET.L32;1       9776        7         0           581         00:02.1




;                                    COMMAND QUALIFIERS

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

;  792          0783  0
; Size:         1099 code + 528 data bytes
; Run Time:        00:23.6
; Elapsed Time:    00:46.4
; Lines/CPU Min:     1987
; Lexemes/CPU-Min: 18274
; Memory Used:  149 pages
; Compilation Complete
```

0237 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



MOMBLDMSG
LIS

MOMINISTA
LIS

MOMLOAD
LIS

MOMACPIO
LIS

MOMDAT
LIS

NMAHEAD
B32

NPADEF
SDL

MOMDEF
SDL

NMATAIL
B32

MOMDDL
B32

NPAMAC
MAR

MOMLIB
LIS

MOMFILIO
LIS