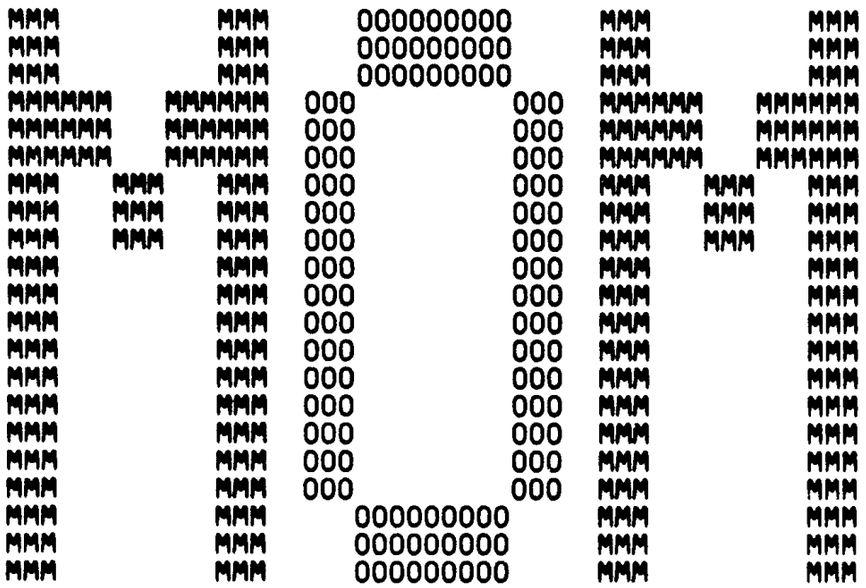

B C D E F G H I J K L M N B C D E F G H I J K L M N B C D E F G H I J K L M N B C D E F G H I

```
MM        MM    000000    MM        MM  FFFFFFFFFF     IIIIII    LL            IIIIII      000000
MM        MM    000000    MM        MM  FFFFFFFFFF     IIIIII    LL            IIIIII      000000
MMMM    MMMM  00      00  MMMM    MMMM  FF               II      LL              II      00      00
MMMM    MMMM  00      00  MMMM    MMMM  FF               II      LL              II      00      00
MM  MM  MM  00      00  MM  MM  MM  FF               II      LL              II      00      00
MM  MM  MM  00      00  MM  MM  MM  FF               II      LL              II      00      00
MM      MM  00      00  MM      MM  FFFFFFFF         II      LL              II      00      00
MM      MM  00      00  MM      MM  FFFFFFFF         II      LL              II      00      00
MM      MM  00      00  MM      MM  FF               II      LL              II      00      00
MM      MM  00      00  MM      MM  FF               II      LL              II      00      00
MM      MM  00      00  MM      MM  FF               II      LL              II      00      00
MM      MM  00      00  MM      MM  FF               II      LL              II      00      00
MM      MM    000000    MM      MM  FF             IIIIII    LLLLLLLLLL    IIIIII      000000
MM      MM    000000    MM      MM  FF             IIIIII    LLLLLLLLLL    IIIIII      000000


LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;  1          0001  0 %TITLE 'MOM Service file I/O modules'
;  2          0002  0 MODULE MOMFILEIO (
;  3          0003  0                 LANGUAGE (BLISS32),
;  4          0004  0                 ADDRESSING_MODE (NONEXTERNAL=LONG_RELATIVE,
;  5          0005  0                                  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 !++
; 35          0035  1 ! FACILITY:  DECnet-VAX Network Management Maintenance Operations Module (MOM)
; 36          0036  1 !
; 37          0037  1 ! ABSTRACT:
; 38          0038  1 !       This module contains general purpose file access routines
; 39          0039  1 !       necessary for manipulating the Load and Dump files for NML.
; 40          0040  1 !
; 41          0041  1 ! ENVIRONMENT:  VAX/VMS Operating System
; 42          0042  1 !
; 43          0043  1 ! AUTHOR:  Kathy Perko
; 44          0044  1 !
; 45          0045  1 ! CREATION DATE:  11-Jan-1983
; 46          0046  1 !
; 47          0047  1 ! MODIFIED BY:
; 48          0048  1 !       V03-001 MKP0001         Kathy Perko             12-April-1984
; 49          0049  1 !               Open load files for shared read so more than one MOM can
; 50          0050  1 !               load the file at a time.
; 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$SRVCLOSE,
;    62          0061  1     MOM$SRVOPEN,
;    63          0062  1     MOM$SRVREAD,
;    64          0063  1     MOM$SRVWRITE,
;    65          0064  1     MOM$SRVREWIND;
;    66          0065  1
;    67          0066  1 !
;    68          0067  1 ! INCLUDE FILES:
;    69          0068  1 !
;    70          0069  1
;    71          0070  1 LIBRARY 'LIB$:MOMLIB.L32';
;    72          0071  1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
;    73          0072  1 LIBRARY 'SYS$LIBRARY:STARLET.L32';
;    74          0073  1
;    75          0074  1 !
;    76          0075  1 ! OWN STORAGE:
;    77          0076  1 !
;    78          0077  1
;    79          0078  1 OWN
;    80          0079  1     MOM$T_FAB : $FAB_DECL,      ! Image file FAB
;    81          0080  1     MOM$T_RAB : $RAB_DECL;      ! Image file RAB
;    82          0081  1
;    83          0082  1 !
;    84          0083  1 ! EXTERNAL REFERENCES:
;    85          0084  1 !
;    86          0085  1
;    87          0086  1 $MOM_EXTERNALS;
;    88          0087  1
;    89          0088  1 EXTERNAL ROUTINE
;    90          0089  1     MOM$DEBUG_TXT,
;    91          0090  1     MOM$BLD_REPLY;
;    92          0091  1
```

```
;    94          0092  1 %SBTTL  'MOM$SRVCLOSE   Close currently opened file'
;    95          0093  1 GLOBAL ROUTINE MOM$SRVCLOSE =
;    96          0094  1 !++
;    97          0095  1 ! FUNCTIONAL DESCRIPTION:
;    98          0096  1 !
;    99          0097  1 !       This routine closes the currently opened file.
;   100          0098  1 !
;   101          0099  1 ! ROUTINE VALUE:
;   102          0100  1 ! COMPLETION CODE:
;   103          0101  1 !
;   104          0102  1 !       Signals errors.
;   105          0103  1 !
;   106          0104  1 !--
;   107          0105  2 BEGIN
;   108          0106  2
;   109          0107  2 LOCAL
;   110          0108  2     STATUS;                         ! Temporary return status
;   111          0109  2 !
;   112          0110  2 ! Close the file.
;   113          0111  2 !
;   114          0112  2 STATUS = $CLOSE (FAB = MOM$T_FAB);
;   115          0113  2 !
;   116          0114  2 ! If the operation was successful then log it.
;   117          0115  2 !
;   118          0116  2 IF .STATUS THEN
;   119          0117  2     MOM$DEBUG_TXT (DBG$C_SRVTRC,
;   120          0118  3                    $ASCID ('Image file closed.')
;   121          0119  2                    );
;   122          0120  2
;   123          0121  2 RETURN .STATUS
;   124          0122  2
;   125          0123  1 END;                                ! End of MOM$SRVCLOSE


                                                        .TITLE  MOMFILEIO MOM Service file I/O modules
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

73 6F 6C 63 20 65 6C 69 66 20 65 67 61 6D 49  00000 P.AAB:  .ASCII  \Image file closed.\
                                              2E 64 65  0000F
                                                        00012       .BLKB   2
                                          00000012      00014 P.AAA:  .LONG   18
                                          00000000'     00018       .ADDRESS P.AAB

                                                        .PSECT  $OWN$,NOEXE,2

                                                  00000 MOM$T_FAB:
                                                        .BLKB   80
                                                  00050 MOM$T_RAB:
                                                        .BLKB   68

                                                        .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$DEBUG_TXT, MOM$BLD_REPLY
                                                        .EXTRN  SYS$CLOSE

                                                        .PSECT  $CODE$,NOWRT,2

                                                0004 00000      .ENTRY  MOM$SRVCLOSE, Save R2                   ; 0093
                                    00000000'   EF   9F 00002   PUSHAB  MOM$T_FAB                               ; 0112
                        00000000G   00              01   FB 00008   CALLS   #1, SYS$CLOSE                        ;
                                    52              50   D0 0000F   MOVL    R0, STATUS                          ;
                                    0F              52   E9 00012   BLBC    STATUS, 1$                          ; 0116
                                    00000000'   EF   9F 00015   PUSHAB  P.AAA                                   ; 0118
                                                06   DD 0001B   PUSHL   #6                                     ; 0117
                        00000000G   00              02   FB 0001D   CALLS   #2, MOM$DEBUG_TXT                   ;
                                    50              52   D0 00024 1$:   MOVL    STATUS, R0                      ; 0121
                                                    04 00027       RET                                         ; 0123

; Routine Size:  40 bytes,    Routine Base:  $CODE$ + 0000

;  126          0124  1
```

```
;   128          0125  1 %SBTTL 'MOM$SRVOPEN     Open requested file'
;   129          0126  1 GLOBAL ROUTINE MOM$SRVOPEN (FILNAMDSC, ACCESS) =
;   130          0127  1
;   131          0128  1 !++
;   132          0129  1 ! FUNCTIONAL DESCRIPTION:
;   133          0130  1 !       This routine opens the specified file (if read access) or
;   134          0131  1 !       creates the file (if read/write access).
;   135          0132  1 !
;   136          0133  1 ! FORMAL PARAMETERS:
;   137          0134  1 !       FILNAMDSC       Descriptor of the filename to be opened.
;   138          0135  1 !       ACCESS          File access code (read or read/write).
;   139          0136  1 !
;   140          0137  1 ! SIDE EFFECTS:
;   141          0138  1 !
;   142          0139  1 !       Returns the status of the operation.
;   143          0140  1 !
;   144          0141  1 !--
;   145          0142  1
;   146          0143  2 BEGIN
;   147          0144  2
;   148          0145  2 MAP
;   149          0146  2     FILNAMDSC : REF VECTOR,
;   150          0147  2     ACCESS    : BYTE;
;   151          0148  2
;   152          0149  2 LOCAL
;   153          0150  2     STV,                                    ! To save Status Value field.
;   154          0151  2     MOM_L_STATUS;
;   155          0152  2
;   156          0153  2 !
;   157          0154  2 ! Fill in FAB and CREATE/OPEN the file depending on desired access
;   158          0155  2 !
;   159          0156  2 IF .ACCESS EQLU NMA$C_OPN_AC_RW THEN            ! Read/write=create file
;   160          0157  3     BEGIN
;   161        P 0158  3     $FAB_INIT
;   162        P 0159  3         (
;   163        P 0160  3         FAB = MOM$T_FAB,                    ! Pointer to FAB
;   164        P 0161  3         DNM = 'SYS$SYSTEM:.SYS',            ! Default file name string
;   165        P 0162  3         FNA = .FILNAMDSC [1],               ! Ptr to filname string
;   166        P 0163  3         FNS = .FILNAMDSC [0],               ! Length of filnam string
;   167        P 0164  3         FAC = PUT,                          ! Write access record I/O
;   168        P 0165  3         FOP = (CBT,MXV,SQO,SUP,TEF),        ! File open options
;   169        P 0166  3         MRS = 512,                          ! Maximum record size
;   170        P 0167  3         ORG = SEQ,                          ! Sequential file (create)
;   171        P 0168  3         RFM = FIX                           ! Fixed length records
;   172          0169  3         );
;   173          0170  3
;   174          0171  3     MOM_L_STATUS = $CREATE (FAB = MOM$T_FAB); ! Create the file
;   175          0172  3     STV = .MOM$T_FAB [FAB$L_STV];
;   176          0173  3     END
;   177          0174  2 ELSE                                        ! Assume RO (open not create)
;   178          0175  3     BEGIN
;   179        P 0176  3     $FAB_INIT
;   180        P 0177  3         (
;   181        P 0178  3         FAB = MOM$T_FAB,                    ! FAB address
;   182        P 0179  3         DNM = 'SYS$SYSTEM:.SYS',            ! Default file name string
;   183        P 0180  3         FNA = .FILNAMDSC [1],               ! Ptr to filname string
;   184        P 0181  3         FNS = .FILNAMDSC [0],               ! Length of filnam string
```

```
;   185        P 0182  3          FAC = GET,                      ! Read access, record I/O
;   186        P 0183  3          SHR = GET,                      ! Allow other MOMs to read it too.
;   187        P 0184  3          FOP = SQO                       ! Sequential processing only
;   188          0185  3          );
;   189          0186  3
;   190          0187  3      MOM_L_STATUS = $OPEN (FAB = MOM$T_FAB);
;   191          0188  3      STV = .MOM$T_FAB [FAB$L_STV];
;   192          0189  2      END;
;   193          0190  2
;   194          0191  2  IF .MOM_L_STATUS THEN
;   195          0192  3      BEGIN
;   196          0193  3      !
;   197          0194  3      ! Build the RAB.
;   198          0195  3      !
;   199        P 0196  3      $RAB_INIT
;   200        P 0197  3          (
;   201        P 0198  3          RAB = MOM$T_RAB,                ! Associated RAB
;   202        P 0199  3          FAB = MOM$T_FAB         ! Associated FAB
;   203          0200  3          );
;   204          0201  3
;   205          0202  3      MOM_L_STATUS = $CONNECT (RAB = MOM$T_RAB);
;   206          0203  3      STV = .MOM$T_RAB [RAB$L_STV];
;   207          0204  2      END;
;   208          0205  2  !
;   209          0206  2  ! If the operation was successful then log it.
;   210          0207  2  !
;   211          0208  2  IF .MOM_L_STATUS THEN
;   212          0209  2      MOM$DEBUG_TXT (DBG$C_SRVTRC,
;   213          0210  2                     $ASCID ('Image file opened.'))
;   214          0211  2  ELSE
;   215          0212  2      !
;   216          0213  2      ! If the create, open, or connect failed, set up the message block to
;   217          0214  2      ! log an error.
;   218          0215  2      !
;   219          0216  3      BEGIN
;   220          0217  3      MOM$AB_MSGBLOCK [MSB$L_FLAGS] = MSB$M_DET_FLD OR
;   221          0218  3                                      MSB$M_MSG_FLD OR
;   222          0219  3                                      MSB$M_MSG2_FLD;
;   223          0220  3      MOM$AB_MSGBLOCK [MSB$B_CODE] = NMA$C_STS_FOP;
;   224          0221  3      MOM$AB_MSGBLOCK [MSB$L_TEXT] = .MOM_L_STATUS;
;   225          0222  3      MOM$AB_MSGBLOCK [MSB$L_TEXT2] = .STV;
;   226          0223  2      END;
;   227          0224  2
;   228          0225  2  RETURN .MOM_L_STATUS;
;   229          0226  2
;   230          0227  1  END;                                   ! End of MOM$SRVOPEN


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

53  59  53  2E  3A  4D  45  54  53  59  53  24  53  59  53  0001C P.AAC:   .ASCII  \SYS$SYSTEM:.SYS\                          ;
53  59  53  2E  3A  4D  45  54  53  59  53  24  53  59  53  0002B P.AAD:   .ASCII  \SYS$SYSTEM:.SYS\                          ;
6E  65  70  6F  20  65  6C  69  66  20  65  67  61  6D  49  0003A P.AAF:   .ASCII  \Image file opened.\                       ;
                                                    2E  64  65  00049                                                       ;
                                                  00000012  0004C P.AAE:   .LONG   18                                       ;
                                                  00000000'  00050         .ADDRESS P.AAF                                 ;
```

```
                                        $RMS_PTR=           MOM$T_FAB
                                        $RMS_PTR=           MOM$T_FAB
                                        $RMS_PTR=           MOM$T_RAB
                                                .EXTRN  SYS$CREATE, SYS$OPEN
                                                .EXTRN  SYS$CONNECT

                                                .PSECT  $CODE$,NOWRT,2

                                    07FC 00000          .ENTRY  MOM$SRVOPEN, Save R2,R3,R4,R5,R6,R7,R8,R9,-      ; 0126
                                                                R10
                                5A 00000000G    00  9E 00002         MOVAB   MOM$AB_MSGBLOCK, R10
                                59 00000000'    EF  9E 00009         MOVAB   P.AAC, R9
                                58 00000000'    EF  9E 00010         MOVAB   $RMS_PTR, R8
                                56       04     AC  D0 00017         MOVL    FILNAMDSC, R6                        ; 0169
                                01       08     AC  91 0001B         CMPB    ACCESS, #1                           ; 0156
                                                43  12 0001F         BNEQ    1$
    0050    8F          00          6E          00  2C 00021         MOVC5   #0, (SP), #0, #80, $RMS_PTR          ; 0169
                                                68      00028
                                68       5003   8F  B0 00029         MOVW    #20483, $RMS_PTR
                          04    A8 10200046     8F  D0 0002E         MOVL    #270532678, $RMS_PTR+4
                          16    A8              01  90 00035         MOVB    #1, $RMS_PTR+22
                                            1D  A8  94 00038         CLRB    $RMS_PTR+29
                          1F    A8              01  90 0003D         MOVB    #1, $RMS_PTR+31
                          2C    A8        04    A6  D0 00041         MOVL    4(R6), $RMS_PTR+44
                          30    A8              69  9E 00046         MOVAB   P.AAC, $RMS_PTR+48
                          34    A8        04    BC  90 0004A         MOVB    @FILNAMDSC, $RMS_PTR+52
                          35    A8              0F  90 0004F         MOVB    #15, $RMS_PTR+53
                          36    A8      0200    8F  B0 00053         MOVW    #512, $RMS_PTR+54
                                                58  DD 00059         PUSHL   R8                                   ; 0171
                    00000000G   00              01  FB 0005B         CALLS   #1, SYS$CREATE
                                                38  11 00062         BRB     2$
    0050    8F          00          6E          00  2C 00064 1$:     MOVC5   #0, (SP), #0, #80, $RMS_PTR          ; 0185
                                                68      0006B
                                68       5003   8F  B0 0006C         MOVW    #20483, $RMS_PTR
                          04    A8         40   8F  9A 00071         MOVZBL  #64, $RMS_PTR+4
                          16    A8       0202   8F  B0 00076         MOVW    #514, $RMS_PTR+22
                          1F    A8              02  90 0007C         MOVB    #2, $RMS_PTR+31
                          2C    A8        04    A6  D0 00080         MOVL    4(R6), $RMS_PTR+44
                          30    A8        0F    A9  9E 00085         MOVAB   P.AAD, $RMS_PTR+48
                          34    A8        04    BC  90 0008A         MOVB    @FILNAMDSC, $RMS_PTR+52
                          35    A8              0F  90 0008F         MOVB    #15, $RMS_PTR+53
                                                58  DD 00093         PUSHL   R8                                   ; 0187
                    00000000G   00              01  FB 00095         CALLS   #1, SYS$OPEN
                                56              50  D0 0009C 2$:     MOVL    R0, MOM$L_STATUS
                                57       0C     A8  D0 0009F         MOVL    MOM$T_FAB+12, STV                    ; 0172
                                36              56  E9 000A3         BLBC    MOM$L_STATUS, 3$                     ; 0191
    0044    8F          00          6E          00  2C 000A6         MOVC5   #0, (SP), #0, #68, $RMS_PTR          ; 0200
                                            50  A8      000AD
                          50    A8       4401   8F  B0 000AF         MOVW    #17409, $RMS_PTR
                        008C    C8              68  9E 000B5         MOVAB   MOM$T_FAB, $RMS_PTR+60
                                            50  A8  9F 000BA         PUSHAB  MOM$T_RAB                            ; 0202
                    00000000G   00              01  FB 000BD         CALLS   #1, SYS$CONNECT
                                56              50  D0 000C4         MOVL    R0, MOM$L_STATUS
                                57       5C     A8  D0 000C7         MOVL    MOM$T_RAB+12, STV                    ; 0203
                                0E              56  E9 000CB         BLBC    MOM$L_STATUS, 3$                     ; 0208
                                            30  A9  9F 000CE         PUSHAB  P.AAE                                ; 0210
```

```
                                 06  DD 000D1          PUSHL   #6                                              ; 0209
                    00000000G    00  02  FB 000D3      CALLS   #2, MOM$DEBUG_TXT                               ;
                                     0B  11 000DA      BRB     4$                                              ;
                                 6A  0E  D0 000DC 3$:   MOVL    #14, MOM$AB_MSGBLOCK                            ; 0218
                              04 AA  0D  BE 000DF      MNEGB   #13, MOM$AB_MSGBLOCK+4                          ; 0220
                              0C AA  56  7D 000E3      MOVQ    MOM_L_STATUS, MOM$AB_MSGBLOCK+12                ; 0221
                                 50  56  D0 000E7 4$:   MOVL    MOM_L_STATUS, R0                                ; 0225
                                     04 000EA          RET                                                     ; 0227

; Routine Size:  235 bytes,    Routine Base:  $CODE$ + 0028
```

```
;   232     0228  1 %SBTTL 'MOM$SRVREAD Read a record from the opened file'
;   233     0229  1 GLOBAL ROUTINE MOM$SRVREAD (BUFFERDSC, BYTES_IN_BUF, LOAD_FILE) =
;   234     0230  1
;   235     0231  1 !++
;   236     0232  1 ! FUNCTIONAL DESCRIPTION:
;   237     0233  1 !
;   238     0234  1 !       This routine is called while doing down line load to get the
;   239     0235  1 !       contents of the load file (from disk or another node) to send
;   240     0236  1 !       to the node being loaded.  It reads records from the currently
;   241     0237  1 !       opened load file into the specified buffer and return in the
;   242     0238  1 !       BYTES_IN_BUF parameter the number of bytes actually read.
;   243     0239  1 !
;   244     0240  1 ! FORMAL PARAMETERS:
;   245     0241  1 !
;   246     0242  1 !       BUFFERDSC       Descriptor of the receive buffer.
;   247     0243  1 !       BYTES_IN_BUF    Word to receive the actual number of bytes read.
;   248     0244  1 !       LOAD_FILE       Identification of which load file is being loaded
;   249     0245  1 !                       (secondary, tertiary, or operating system).  Used
;   250     0246  1 !                       for signalling errors.
;   251     0247  1 !
;   252     0248  1 ! ROUTINE VALUE:
;   253     0249  1 ! COMPLETION CODES:
;   254     0250  1 !
;   255     0251  1 !       Returns the status of the operation.
;   256     0252  1 !
;   257     0253  1 ! SIDE EFFECTS:
;   258     0254  1 !
;   259     0255  1 !       Fills in the formal parameter BYTES_IN_BUF with the size of data
;   260     0256  1 !       actually received.
;   261     0257  1 !
;   262     0258  1 !--
;   263     0259  1
;   264     0260  2 BEGIN
;   265     0261  2
;   266     0262  2 MAP
;   267     0263  2     BUFFERDSC : REF VECTOR;
;   268     0264  2
;   269     0265  2 LOCAL
;   270     0266  2     MOM_L_STATUS,
;   271     0267  2     BUF_PTR,            ! Pointer into read buffer to place to put next record.
;   272     0268  2     BUF_LEFT,           ! Bytes left in read buffer.
;   273     0269  2     BYTES_IN_REC,       ! Size of last record read into buffer.
;   274     0270  2     MSGSIZE;            ! For signalled error message.
;   275     0271  2
;   276     0272  2
;   277     0273  2 !
;   278     0274  2 ! Read in as many complete records as will fit into the buffer.  The
;   279     0275  2 ! file has fixed length records.
;   280     0276  2 !
;   281     0277  2 ! Set up RAB fields for the beginning of the read buffer, the read
;   282     0278  2 ! buffer size, and the size of the record just read (zero to begin
;   283     0279  2 ! with).
;   284     0280  2 !
;   285     0281  2 BUF_PTR = .BUFFERDSC [1];       ! Read buffer address
;   286     0282  2 BUF_LEFT = .BUFFERDSC [0];      ! Read buffer size
;   287     0283  2 BYTES_IN_REC = 0;
;   288     0284  2 !
```

```
;  289          0285  2 ! If there is room in the buffer for another complete record, then
;  290          0286  2 ! read it in.
;  291          0287  2 !
;  292          0288  2 WHILE .BUF_LEFT GEQ .BYTES_IN_REC DO
;  293          0289  3     BEGIN
;  294          0290  3     MOM$T_RAB [RAB$L_UBF] = .BUF_PTR;   ! Read buffer address
;  295          0291  3     MOM$T_RAB [RAB$W_USZ] = .BUF_LEFT;  ! Read buffer size
;  296          0292  3     MOM_L_STATUS = $GET (RAB = MOM$T_RAB);
;  297          0293  3     IF .MOM_L_STATUS
;  298          0294  3     THEN
;  299          0295  4         BEGIN
;  300          0296  4         BYTES_IN_REC = .MOM$T_RAB [RAB$W_RSZ];
;  301          0297  4         BUF_LEFT = .BUF_LEFT - .BYTES_IN_REC;
;  302          0298  4         BUF_PTR = .BUF_PTR + .BYTES_IN_REC;
;  303          0299  4         END
;  304          0300  3     ELSE
;  305          0301  3         !
;  306          0302  3         ! If there was an error check to see if it was EOF.  If it was,
;  307          0303  3         ! and there is data in the read buffer, return the data to the
;  308          0304  3         ! caller.  Otherwise, signal an error and quit the load.
;  309          0305  3         !
;  310          0306  4         BEGIN
;  311          0307  4         IF (.MOM_L_STATUS EQL RMS$_EOF) AND
;  312          0308  4            (.BUF_PTR GTR .BUFFERDSC [1]) THEN
;  313          0309  5             BEGIN
;  314          0310  5             MOM_L_STATUS = SUCCESS;
;  315          0311  5             EXITLOOP;
;  316          0312  5             END
;  317          0313  4         ELSE
;  318          0314  5             BEGIN
;  319          0315  5             MOM$AB_MSGBLOCK [MSB$L_FLAGS]  = MSB$M_DET_FLD OR
;  320          0316  5                                              MSB$M_MSG_FLD;
;  321          0317  5             MOM$AB_MSGBLOCK [MSB$B_CODE]   = NMA$C_STS_FIO;
;  322          0318  5             MOM$AB_MSGBLOCK [MSB$W_DETAIL] = .LOAD_FILE;
;  323          0319  5             MOM$AB_MSGBLOCK [MSB$L_TEXT]   = .MOM_L_STATUS;
;  324          0320  5             MOM$AB_MSGBLOCK [MSB$L_TEXT2] = .MOM$T_RAB [RAB$L_STV];
;  325          0321  5             MOM$BLD_REPLY (MOM$AB_MSGBLOCK, MSGSIZE);
;  326          0322  5             $SIGNAL_MSG (MOM$AB_NICE_XMIT_BUF, .MSGSIZE);
;  327          0323  4             END;
;  328          0324  3         END;
;  329          0325  2     END;
;  330          0326  2 !
;  331          0327  2 ! Set up actual number of bytes read into read buffer.
;  332          0328  2 !
;  333          0329  2 (.BYTES_IN_BUF)<0,16> = .BUF_PTR - .BUFFERDSC [1];
;  334          0330  2 RETURN .MOM_L_STATUS
;  335          0331  1 END;                                ! End of MOM$SRVREAD


                                                            .EXTRN  SYS$GET

                                                 01FC 00000 .ENTRY  MOM$SRVREAD, Save R2,R3,R4,R5,R6,R7,R8       ; 0229
                    58 00000000'   EF   9E 00002 MOVAB   MOM$T_RAB+36, R8                                       ;
                    57 00000000G   00   9E 00009 MOVAB   MOM$AB_MSGBLOCK, R7                                     ;
                    5E             04   C2 00010 SUBL2   #4, SP                                                 ;
                    52        04   AC   D0 00013 MOVL    BUFFERDSC, R2                                          ; 0281
```

```
                                54        04    A2  D0 00017          MOVL    4(R2), BUF_PTR                                  ;
                                56              62  D0 0001B          MOVL    (R2), BUF_LEFT                                  ; 0282
                                                53  D4 0001E          CLRL    BYTES_IN_REC                                    ; 0283
                                53              56  D1 00020 1$:      CMPL    BUF_LEFT, BYTES_IN_REC                          ; 0288
                                                6E  19 00023          BLSS    4$                                              ;
                                68              54  D0 00025          MOVL    BUF_PTR, MOM$T_RAB+36                           ; 0290
                             FC   A8            56  B0 00028          MOVW    BUF_LEFT, MOM$T_RAB+32                          ; 0291
                                          DC    A8  9F 0002C          PUSHAB  MOM$T_RAB                                       ; 0292
                   00000000G  00              01  FB 0002F          CALLS   #1, SYS$GET                                     ;
                                55              50  D0 00036          MOVL    R0, MOM_L_STATUS                                ;
                                0C              55  E9 00039          BLBC    MOM_L_STATUS, 2$                                ; 0293
                                53        FE    A8  3C 0003C          MOVZWL  MOM$T_RAB+34, BYTES_IN_REC                      ; 0296
                                56              53  C2 00040          SUBL2   BYTES_IN_REC, BUF_LEFT                          ; 0297
                                54              53  C0 00043          ADDL2   BYTES_IN_REC, BUF_PTR                           ; 0298
                                                D8  11 00046          BRB     1$                                              ; 0293
                   0001827A     8F              55  D1 00048 2$:      CMPL    MOM_L_STATUS, #98938                            ; 0307
                                                0B  12 0004F          BNEQ    3$                                              ;
                         04     A2              54  D1 00051          CMPL    BUF_PTR, 4(R2)                                  ; 0308
                                                05  15 00055          BLEQ    3$                                              ;
                                55              01  D0 00057          MOVL    #1, MOM_L_STATUS                                ; 0310
                                                37  11 0005A          BRB     4$                                              ; 0309
                                67              06  D0 0005C 3$:      MOVL    #6, MOM$AB_MSGBLOCK                             ; 0315
                         04     A7              12  8E 0005F          MNEGB   #18, MOM$AB_MSGBLOCK+4                          ; 0317
                         08     A7        0C    AC  B0 00063          MOVW    LOAD_FILE, MOM$AB_MSGBLOCK+8                    ; 0318
                         0C     A7              55  D0 00068          MOVL    MOM_L_STATUS, MOM$AB_MSGBLOCK+12                ; 0319
                         10     A7        E8    A8  D0 0006C          MOVL    MOM$T_RAB+12, MOM$AB_MSGBLOCK+16                ; 0320
                                      4080      8F  BB 00071          PUSHR   #^M<R7,SP>                                      ; 0321
                   00000000G  00              02  FB 00075          CALLS   #2, MOM$BLD_REPLY                               ;
                                                6E  DD 0007C          PUSHL   MSGSIZE                                         ; 0322
                                00000000G  00  9F  DD 0007E          PUSHAB  MOM$AB_NICE_XMIT_BUF                            ;
                                02070000   8F  DD 00084          PUSHL   #34013184                                       ;
                   00000000G  00              03  FB 0008A          CALLS   #3, LIB$SIGNAL                                  ;
                                                8D  11 00091          BRB     1$                                              ; 0288
                      08   BC              54        04    A2  A3 00093 4$:      SUBW3   4(R2), BUF_PTR, @BYTES_IN_BUF           ; 0329
                                50              55  D0 00099          MOVL    MOM_L_STATUS, R0                                ; 0330
                                                    04 0009C          RET                                                     ; 0331

; Routine Size:  157 bytes,    Routine Base:  $CODE$ + 0113
```

```
;   337          0332  1 %SBTTL 'MOM$SRVWRITE  Write record in currently opened file'
;   338          0333  1 GLOBAL ROUTINE MOM$SRVWRITE (BUFADR, BUFLEN) =
;   339          0334  1
;   340          0335  1 !++
;   341          0336  1 ! FUNCTIONAL DESCRIPTION:
;   342          0337  1 !
;   343          0338  1 !       This routine writes out the buffer specified to the currently
;   344          0339  1 !       opened file.
;   345          0340  1 !
;   346          0341  1 ! FORMAL PARAMETERS:
;   347          0342  1 !
;   348          0343  1 !       BUFADR          The address of buffer to be written
;   349          0344  1 !       BUFLEN          The length of buffer to be written
;   350          0345  1 !
;   351          0346  1 ! SIDE EFFECTS:
;   352          0347  1 !
;   353          0348  1 !       Returns the status of the operation.
;   354          0349  1 !
;   355          0350  1 !--
;   356          0351  1
;   357          0352  2 BEGIN
;   358          0353  2
;   359          0354  2 MAP
;   360          0355  2     BUFLEN : WORD;
;   361          0356  2
;   362          0357  2 LOCAL
;   363          0358  2     STATUS,
;   364          0359  2     MSGSIZE;
;   365          0360  2 !
;   366          0361  2 ! Fill in the user buffer address and size
;   367          0362  2 !
;   368          0363  2 MOM$T_RAB [RAB$L_RBF] = .BUFADR;
;   369          0364  2 MOM$T_RAB [RAB$W_RSZ] = .BUFLEN;
;   370          0365  2 !
;   371          0366  2 ! Write the record and return the status.
;   372          0367  2 !
;   373          0368  2 STATUS = $PUT (RAB = MOM$T_RAB);
;   374          0369  2 IF NOT .STATUS THEN
;   375          0370  2     !
;   376          0371  2     ! If there was an error, then signal it and quit the load.
;   377          0372  2     !
;   378          0373  3     BEGIN
;   379          0374  3     MOM$AB_MSGBLOCK [MSB$L_FLAGS]  = MSB$M_DET_FLD OR
;   380          0375  3                                      MSB$M_MSG_FLD;
;   381          0376  3     MOM$AB_MSGBLOCK [MSB$B_CODE]   = NMA$C_STS_FIO;
;   382          0377  3     MOM$AB_MSGBLOCK [MSB$L_TEXT]   = .STATUS;
;   383          0378  3     MOM$AB_MSGBLOCK [MSB$L_TEXT2] = .MOM$T_RAB [RAB$L_STV];
;   384          0379  3     MOM$BLD_REPLY (MOM$AB_MSGBLOCK, MSGSIZE);
;   385          0380  3     $SIGNAL_MSG (MOM$AB_NICE_XMIT_BUF, .MSGSIZE);
;   386          0381  2     END;
;   387          0382  2 RETURN .STATUS;
;   388          0383  1 END;                                   ! End of MOM$SRVWRITE


                                                                .EXTRN  SYS$PUT
```

```
                                                 001C 00000          .ENTRY  MOM$SRVWRITE, Save R2,R3,R4                             ; 0333
                                          54 00000000'   EF   9E 00002       MOVAB   MOM$T_RAB+40, R4                                ;
                                          53 00000000G   00   9E 00009       MOVAB   MOM$AB_MSGBLOCK, R3                             ;
                                          5E                  04   C2 00010  SUBL2   #4, SP                                         ;
                                          64                  04   AC   D0 00013  MOVL    BUFADR, MOM$T_RAB+40                      ; 0363
                                       FA A4                  08   AC   B0 00017  MOVW    BUFLEN, MOM$T_RAB+34                      ; 0364
                                          D8                  A4   9F 0001C  PUSHAB  MOM$T_RAB                                       ; 0368
                                00000000G 00                  01   FB 0001F  CALLS   #1, SYS$PUT                                   ;
                                          52                  50   D0 00026  MOVL    R0, STATUS                                    ;
                                          30                  52   E8 00029  BLBS    STATUS, 1$                                    ; 0369
                                          63                  06   D0 0002C  MOVL    #6, MOM$AB_MSGBLOCK                           ; 0374
                                       04 A3                  12   8E 0002F  MNEGB   #18, MOM$AB_MSGBLOCK+4                       ; 0376
                                       0C A3                  52   D0 00033  MOVL    STATUS, MOM$AB_MSGBLOCK+12                    ; 0377
                                       10 A3              E4  A4   D0 00037  MOVL    MOM$T_RAB+12, MOM$AB_MSGBLOCK+16              ; 0378
                                                    4008  8F   BB 0003C  PUSHR   #^M<R3,SP>                                      ; 0379
                                00000000G 00                  02   FB 00040  CALLS   #2, MOM$BLD_REPLY                           ;
                                                              6E   DD 00047  PUSHL   MSGSIZE                                       ; 0380
                                             00000000G        00   9F 00049  PUSHAB  MOM$AB_NICE_XMIT_BUF                          ;
                                             02070000         8F   DD 0004F  PUSHL   #34013184                                     ;
                                00000000G 00                  03   FB 00055  CALLS   #3, LIB$SIGNAL                             ;
                                          50                  52   D0 0005C 1$:  MOVL    STATUS, R0                                    ; 0382
                                                              04   05F       RET                                                   ; 0383

; Routine Size:  96 bytes,    Routine Base:  $CODE$ + 01B0
```

```
;  390          0384  1 %SBTTL 'MOM$SRVREWIND  Rewind currently open file'
;  391          0385  1 GLOBAL ROUTINE MOM$SRVREWIND =
;  392          0386  1
;  393          0387  1 !++
;  394          0388  1 ! FUNCTIONAL DESCRIPTION:
;  395          0389  1 !
;  396          0390  1 !       This routine rewinds the currently open file.
;  397          0391  1 !
;  398          0392  1 ! SIDE EFFECTS:
;  399          0393  1 !
;  400          0394  1 !       Returns the status of the operation.
;  401          0395  1 !
;  402          0396  1 !--
;  403          0397  1
;  404          0398  2 BEGIN
;  405          0399  2
;  406          0400  2 !
;  407          0401  2 ! Rewind the file
;  408          0402  2 !
;  409          0403  3 RETURN $REWIND (RAB = MOM$T_RAB)
;  410          0404  3
;  411          0405  1 END;                                    ! End of MOM$SRVREWIND


                                                                .EXTRN  SYS$REWIND

                                                                .ENTRY  MOM$SRVREWIND, Save nothing                     ; 0385
                                        0000 00000              .PUSHAB MOM$T_RAB                                       ; 0403
                                00000000'  EF   9F 00002        CALLS   #1, SYS$REWIND                                  ;
                     00000000G  00         01   FB 00008        RET                                                     ; 0405
                                           04 0000F

; Routine Size:  16 bytes,    Routine Base:  $CODE$ + 0210
```

```
;  413          0406  1 END                                   ! End of module
;  414          0407  1
;  415          0408  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY

;       Name                    Bytes                   Attributes

;  $OWN$                          148  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                          84  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         544  NOVEC,NOWRT,  RD ,  EXE,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        19         9          21       00:00.1
;  _$255$DUA28:[SHRLIB]NMALIBRY.L32;1      887         3         0          47       00:00.2
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776        81         0         581       00:02.1


;                                  COMMAND QUALIFIERS

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

; Size:         544 code + 232 data bytes
; Run Time:       00:16.7
; Elapsed Time:   00:43.6
; Lines/CPU Min:   1466
; Lexemes/CPU-Min: 29227
; Memory Used:  153 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