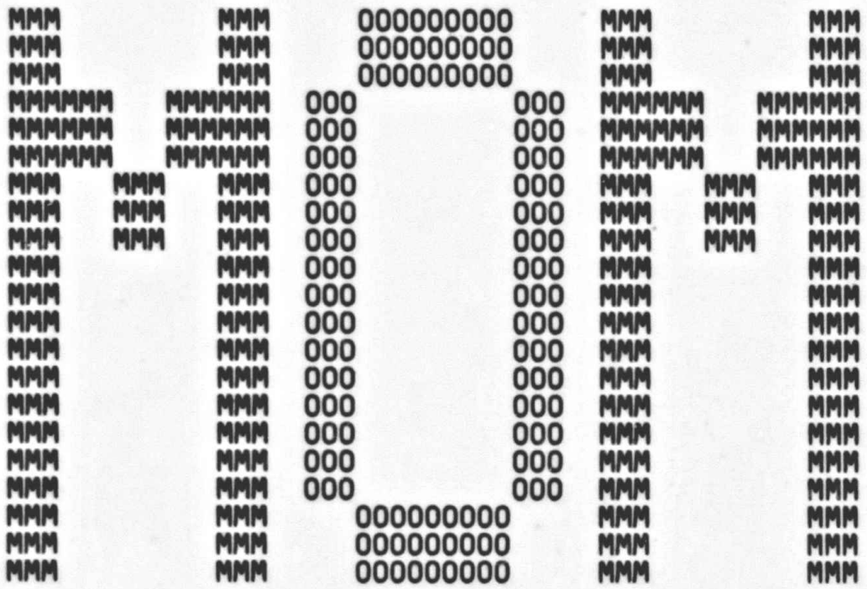
MMM MMM 000000000 MMM MMM
MMM MMM 000000000 MMM MMM
MMM MMM 000000000 MMM MMM
MMMMMM MMMMMM 000 000 MMMMMM MMMMMM
MMMMMM MMMMMM 000 000 MMMMMM MMMMMM
MMMMMM MMMMMM 000 000 MMMMMM MMMMMM
MMM MMM MMM 000 000 MMM MMM MMM
MMM MMM MMM 000 000 MMM MMM MMM
MMM MMM MMM 000 000 MMM MMM MMM
MMM MMM 000 000 MMM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

FILEID**MOMDDL

```
MM      MM    OOOOOO    MM      MM  DDDDDDDD    DDDDDDDD    LL
MM      MM    OOOOOO    MM      MM  DDDDDDDD    DDDDDDDD    LL
MMMM  MMMM  OO      OO  MMMM  MMMM  DD      DD  DD      DD  LL
MMMM  MMMM  OO      OO  MMMM  MMMM  DD      DD  DD      DD  LL
MM  MM  MM  OO      OO  MM  MM  MM  DD      DD  DD      DD  LL
MM  MM  MM  OO      OO  MM  MM  MM  DD      DD  DD      DD  LL
MM      MM  OO      OO  MM      MM  DD      DD  DD      DD  LL
MM      MM  OO      OO  MM      MM  DD      DD  DD      DD  LL
MM      MM  OO      OO  MM      MM  DD      DD  DD      DD  LL
MM      MM  OO      OO  MM      MM  DD      DD  DD      DD  LL
MM      MM  OO      OO  MM      MM  DD      DD  DD      DD  LL
MM      MM  OO      OO  MM      MM  DD      DD  DD      DD  LL
MM      MM  OO      OO  MM      MM  DD      DD  DD      DD  LL
MM      MM    OOOOOO    MM      MM  DDDDDDDD    DDDDDDDD    LLLLLLLLLL
MM      MM    OOOOOO    MM      MM  DDDDDDDD    DDDDDDDD    LLLLLLLLLL
```

```
BBBBBBBB      333333      222222
BBBBBBBB    333333      222222
BB      BB  33      33  22      22
BB      BB  33      33  22      22
BB      BB          33          22
BB      BB          33          22
BBBBBBBB        33          22
BBBBBBBB        33          22
BB      BB          33      22
BB      BB          33      22
BB      BB  33      33    22
BB      BB  33      33    22
BBBBBBBB      333333    2222222222
BBBBBBBB      333333    2222222222
```

```
!
!       %TITLE 'MOMDDL - MOM Data Definition Library'
!       IDENT = 'V04-000'
!
!****************************************************************************
!*                                                                          *
!*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
!*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
!*  ALL RIGHTS RESERVED.                                                    *
!*                                                                          *
!*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
!*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE          *
!*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER     *
!*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
!*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY       *
!*  TRANSFERRED.                                                            *
!*                                                                          *
!*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE    *
!*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT        *
!*  CORPORATION.                                                            *
!*                                                                          *
!*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS     *
!*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
!*                                                                          *
!****************************************************************************
!
!++
! FACILITY:  DECnet-VAX Network Management Maintenance Operations Module (MOM)
!
! ABSTRACT:
!       Contains structure definitions and global macros used by MOM.
!
! ENVIRONMENT:  VAX/VMS Operating System
!
! AUTHOR:  Kathy Perko
!
! CREATION DATE:  9-Jan-1982
!
! MODIFIED BY:
!       V03-001 MKP0001         Kathy Perko             20-Jan-1984
!               Add SERVICE NODE VERSION parameter.
!
!--

!
! Miscellaneous symbols
!
LITERAL
    FAILURE = 0,
    SUCCESS = 1,
    FALSE = 0,
    TRUE  = 1,
    MOM$NOSIGNAL = 0,
    MOM$SIGNAL = 1;
```

```
!
! Structure declarations used for system defined structures to
! save typing.
!
STRUCTURE
    BBLOCK [O, P, S, E; N] =
        [N]
        (BBLOCK+O)<P,S,E>,

    BBLOCKVECTOR [I, O, P, S, E; N, BS] =
        [N*BS]
        ((BBLOCKVECTOR+I*BS)+O)<P,S,E>;

!
! Macro to signal status message
!
MACRO
    $SIGNAL_MSG [] =
        SIGNAL (MOM$K_SIG_CODE, %REMAINING)
    %;
```

```
!
! Macro to generate Network ACP Control QIO (NFB) P1 buffer contents.  The NFB
! describes SET, SHOW, CLEAR, and ZERO operations.
!
MACRO
        $NFB (FUNC, FLAGS, DATABASE, SRCH_KEY_ONE, OPER_ONE,
                                     SRCH_KEY_TWO, OPER_TWO) =

        BYTE ( %IF %IDENTICAL (FUNC, 0)          ! QIO function code.
               %THEN 0
               %ELSE %NAME ('NFB$C_FC_',FUNC)
               %FI),
        BYTE ( %IF %NULL (FLAGS)                 ! Error Update and Process
               %THEN 0                           !       Multiple Entries flags.
               %ELSE FLAGS
               %FI),
        BYTE ( %IF %IDENTICAL (DATABASE, 0)      ! ACP database to update.
               %THEN 0
               %ELSE %NAME ('NFB$C_DB_',DATABASE)
               %FI),
        BYTE (%IF %NULL (OPER_ONE)               ! Oper1
               %THEN 0
               %ELSE OPER_ONE
               %FI
               ),
        $SRCH_KEY (DATABASE, SRCH_KEY_ONE),      ! Search key one ID
        $SRCH_KEY (DATABASE, SRCH_KEY_TWO),      ! Search key two ID
        BYTE (%IF %NULL (OPER_TWO)               ! Oper2
               %THEN 0
               %ELSE OPER_TWO
               %FI
               ),
        BYTE (0),                                ! Spare
        WORD (0),                                ! variable cell size

        %IF NOT %NULL(%REMAINING)
        %THEN $FIELD_ID_LIST (DATABASE, %REMAINING)
               ,LONG (NFB$C_ENDOFLIST) ! End delimiter for field ID list.
        %ELSE
               LONG (NFB$C_ENDOFLIST)  ! End delimiter for field ID list.
        %FI

        %,

!
! Generate a Search Key ID for an NFB.  If the Search key is null,
! use a wildcard search key ID.
!
$SRCH_KEY (DATABASE, SRCH_ID) =
        LONG ( %IF %NULL (SRCH_ID)
               %THEN NFB$C_WILDCARD
               %ELSE $FIELD_ID (DATABASE, SRCH_ID)
               %FI )
        %,
!
! Generate a list of longwords containing the NETACP field IDs for
```

```
    ! the parameters.  This iterative macro will generate as many
    ! field IDs as are supplied.
    !
    $FIELD_ID_LIST (DATABASE) [FIELD_ID] =
        LONG (%$FIELD_ID (DATABASE, FIELD_ID))
        %,

    $FIELD_ID (DATABASE, FIELD_ID) =
        %IF %IDENTICAL (FIELD_ID, NFB$C_WILDCARD) OR
            %IDENTICAL (FIELD_ID, NFB$C_COLLATE)
        %THEN
                FIELD_ID
        %ELSE
                %IF %NULL (FIELD_ID)
                %THEN   0
                %ELSE %NAME ('NFB$C_',DATABASE,'_',FIELD_ID)
                %FI
        %FI
        %;

!
! Macros to generate Network Control I/O request descriptors.
!
MACRO
        !
        ! Declare the NFB buffer (use the number of input parameters to figure
        ! out how big to make it) and set up a descriptor for it.
        !
        $NFBDSC (NAM) =
        SWITCHES UNAMES;
        OWN
            _NFB : VECTOR [$NFB_ALLOCATION (%REMAINING)]
                                INITIAL ($NFB (%REMAINING));
        BIND
            %NAME(NAM) = UPLIT (%ALLOCATION(_NFB), _NFB);
        UNDECLARE _NFB;
        SWITCHES NOUNAMES
        %,

    $NFB_ALLOCATION [] =
        5+(MAX(0,%LENGTH-6))
        %;
```

```
!
! I/O Status Block definition
!
FIELD
    IOSB_FIELDS =
        SET
        IOS$W_STATUS = [0, 0, 16, 0],    ! Status field
        IOS$W_COUNT  = [2, 0, 16, 0],    ! Byte count field
        IOS$L_INFO   = [4, 0, 32, 0]     ! Device dependent information
        TES;

MACRO
    $IOSB =
        BBLOCK [8]  FIELD (IOSB_FIELDS)
    %;
!
! Macro to create constant string descriptor
!
MACRO
    $ASCID [] =
        (UPLIT (%CHARCOUNT(%STRING(%REMAINING)),
                UPLIT BYTE (%STRING(%REMAINING))))
    %;

MACRO
    $ASCIC [] =
        UPLIT BYTE (%ASCIC %STRING (%REMAINING))
    %;

!
! Macro to declare frequently used externals in MOM
!
MACRO $MOM_EXTERNALS =
    EXTERNAL
        MOM$GL_LOGMASK:          BITVECTOR [32],
        MOM$GL_SVD_INDEX,
        MOM$AB_SERVICE_DATA:     BBLOCKVECTOR [0,SVD$K_ENTRY_LEN],
        MOM$GB_FUNCTION:         BYTE,
        MOM$GB_OPTION_BYTE:      BYTE,
        MOM$GB_ENTITY_CODE:      BYTE,
        MOM$AB_ENTITY_BUF:       BBLOCK [0],
        MOM$GQ_ENTITY_BUF_DSC:   VECTOR [0],
        MOM$GL_SERVICE_FLAGS:    BLOCK [1],
        MOM$AB_NPARSE_BLK:       $NPA_BLKDEF,
        MOM$AB_NICE_RCV_BUF:     BBLOCK [0],
        MOM$AB_NICE_XMIT_BUF:    BBLOCK [0],
        MOM$GQ_NICE_RCV_BUF_DSC:VECTOR [0],
        MOM$GL_NICE_RCV_MSG_LEN,
        MOM$GQ_NICE_XMIT_BUF_DSC:VECTOR [0],
        MOM$AB_MSGBLOCK:         BBLOCK [0],
        MOM$AB_ACPQIO_BUFFER:    BBLOCK [0],
        MOM$GQ_ACPQIO_BUF_DSC:   VECTOR [0],
        MOM$AB_CIB:              BBLOCK [0],
        MOM$AB_LOOP_CIB:         BBLOCK [0],
        MOM$AB_TRIGGER_CIB:      BBLOCK [0],
        MOM$AB_MOP_XMIT_BUF:     BBLOCK [0],
```

```
        MOM$GQ_MOP_XMIT_BUF_DSC:VECTOR [0],
        MOM$AB_MOP_RCV_BUF:     BBLOCK [0],
        MOM$GQ_MOP_RCV_BUF_DSC: VECTOR [0],
        MOM$AB_MOP_MSG:         BBLOCK [0],
        MOM$GQ_MOP_MSG_DSC:     VECTOR [0],
        MOM$GW_EVT_CODE:        BYTE,
        MOM$GB_EVT_POPR:        BYTE,
        MOM$GB_EVT_PRSN:        BYTE,
        MOM$GB_EVT_PSER:        BYTE;

    EXTERNAL LITERAL
        SVD$GK_PCNO_ADD,
        SVD$GK_PCNO_SDV,
        SVD$GK_PCNO_CPU,
        SVD$GK_PCNO_STY,
        SVD$GK_PCNO_DAD,
        SVD$GK_PCNO_DCT,
        SVD$GK_PCNO_IHO,
        SVD$GK_PCNO_NNA,
        SVD$GK_PCNO_SLI,
        SVD$GK_PCNO_SPA,
        SVD$GK_PCNO_HWA,
        SVD$GK_PCNO_SNV,
        SVD$GK_PCNO_LOA,
        SVD$GK_PCNO_SLO,
        SVD$GK_PCNO_TLO,
        SVD$GK_PCNO_DFL,
        SVD$GK_PCNO_SID,
        SVD$GK_PCNO_DUM,
        SVD$GK_PCNO_SDU,
        SVD$GK_PCNO_$HNA,
        SVD$GK_PCNO_$HHW,
        SVD$GK_PCNO_$FTY,
        SVD$GK_PCNO_PHA,
        SVD$GK_PCNO_$DA,
        SVD$GK_PCNO_LPC,
        SVD$GK_PCNO_LPL,
        SVD$GK_PCNO_LPD,
        SVD$GK_PCNO_LPH,
        SVD$GK_PCNO_LPA,
        SVD$GK_PCNO_LPN,
        SVD$GK_PCNO_$LNA,
        SVD$GK_PCNO_$LNH,
        SVD$GK_PCNO_LAN,
        SVD$GK_PCNO_$LNN,
        SVD$GK_PCNO_$LAH,
        SVD$GK_PCLI_STI,
        SVDSC_ENTRY_COUNT;

        %;

!
! NPARSE argument block structure definitions
!
MACRO
    $NPA_ARGDEF =
```

```
        BUILTIN
            AP;
        BIND
            NPARSE_BLOCK = AP : REF $NPA_BLKDEF;
    %;

!
! NPARSE argument block definition macro
!
MACRO
    $NPA_BLKDEF =
        BBLOCK [NPA$K_LENGTH0]
    %;
```

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