

%TITLE 'MOMDDL - MOM Data Definitive 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, NFBSC_WILDCARD) OR
  %IDENTICAL (FIELD_ID, NFBSC_COLLATE)
  %THEN
    FIELD_ID
  %ELSE
    %IF %NULL (FIELD_ID)
    %THEN 0
    %ELSE %NAME ('NFBSC_', 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.

```
$NFBDESC (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
    IOSSW_STATUS = [0, 0, 16, 0],    : Status field
    IOSSW_COUNT  = [2, 0, 16, 0],    : Byte count field
    IOSSL_INFO   = [4, 0, 32, 0]    : Device dependent information
  TES:

```

```

MACRO
  $IOSB =
    BBLOCK [8] FIELD (IOSB_FIELDS)
  X:

```

```

: Macro to create constant string descriptor

```

```

MACRO
  $ASCID [] =
    (UPLIT (%CHARCOUNT(%STRING(%REMAINING)),
    UPLIT BYTE (%STRING(%REMAINING))))
  X:

```

```

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

```

```

: Macro to declare frequently used externals in MOM

```

```

MACRO $MOM_EXTERNAIS =
  EXTERNAL
    MOM$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],

```

```

MOMSGQ_MOP_XMIT_BUF_DSC:VECTOR [0],
MOMSGQ_MOP_RCV_BUF:BBLOCK [0],
MOMSGQ_MOP_RCV_BUF_DSC:VECTOR [0],
MOM$AB_MOP_MSG:BBLOCK [0],
MOMSGQ_MOP_MSG_DSC:VECTOR [0],
MOMSGW_EVT_CODE:BYTE,
MOM$GB_EVT_POPR:BYTE,
MOM$GB_EVT_PRSN:BYTE,
MOM$GB_EVT_PSER:BYTE;

```

EXTERNAL LITERAL

```

SVDSGK_PCNO_ADD,
SVDSGK_PCNO_SDV,
SVDSGK_PCNO_CPU,
SVDSGK_PCNO_STY,
SVDSGK_PCNO_DAD,
SVDSGK_PCNO_DCT,
SVDSGK_PCNO_IHO,
SVDSGK_PCNO_NNA,
SVDSGK_PCNO_SLI,
SVDSGK_PCNO_SPA,
SVDSGK_PCNO_HWA,
SVDSGK_PCNO_SNV,
SVDSGK_PCNO_LOA,
SVDSGK_PCNO_SLO,
SVDSGK_PCNO_TLO,
SVDSGK_PCNO_DFL,
SVDSGK_PCNO_SID,
SVDSGK_PCNO_DUM,
SVDSGK_PCNO_SDU,
SVDSGK_PCNO_$HNA,
SVDSGK_PCNO_$HHW,
SVDSGK_PCNO_$FTY,
SVDSGK_PCNO_PHA,
SVDSGK_PCNO_$DA,
SVDSGK_PCNO_LPC,
SVDSGK_PCNO_LPL,
SVDSGK_PCNO_LPD,
SVDSGK_PCNO_LPH,
SVDSGK_PCNO_LPA,
SVDSGK_PCNO_LPN,
SVDSGK_PCNO_$LNA,
SVDSGK_PCNO_$LNH,
SVDSGK_PCNO_LAN,
SVDSGK_PCNO_$LNN,
SVDSGK_PCNO_$LAH,
SVDSGK_PCLI_STI,
SV$SC_ENTRY_COUNT;

```

```

%:

```

```

: NPARSE argument block structure definitions

```

```

MACRO
  $NPA_ARGDEF =

```


BUILTIN

AP;

BIND

NPARSE_BLOCK = AP : REF \$NPA_BLKDEF;

z;

NPARSE argument block definition macro

MACRO

\$NPA_BLKDEF =

BBLOCK [NPASK_LENGTH0]

z;

