

0001 0
0002 0
0003 0
0004 0
0005 0
0006 0
0007 0
0008 0
0009 0
0010 0
0011 0
0012 0
0013 0
0014 0
0015 0
0016 0
0017 0
0018 0
0019 0
0020 0
0021 0
0022 0
0023 0
0024 0
0025 0
0026 0
0027 0
0028 0
0029 0
0030 0
0031 0
0032 0
0033 0
0034 0
0035 0
0036 0
0037 0
0038 0
0039 0
0040 0
0041 0
0042 0
0043 0
0044 0
0045 0
0046 0
0047 0
0048 0
0049 0
0050 0
0051 0
0052 0
0053 0
0054 0
0055 0
0056 0
0057 0

```
Version: '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.
*****

++
NMAHEAD.B32
    Define $EQLST macro to make library from the NMLIBRY.B32 file
    This source is taken from the following source:
--
++
UTLDEF.B32 - UTILITY DEFINITION MACROS FOR BLISS PROCESSING
OF STARLET DEFINITION MACROS.
--

MACRO TO GENERATE EQLST CONSTRUCTS.
MACRO
    $EQLST(P,G,I,S)[A]=
        %NAME(P,GET1ST_ A) =
            %IF NUL2ND_ A
            %THEN (I) %COUNT*(S) ! ASSUMES I, S ALWAYS GENERATED BY CONVERSION PROGRAM
            %ELSE GET2ND_ A
            %FI %,
    GET1ST_(A,B)=
        A %,
    GET2ND_(A,B)=
        B %, ! KNOWN NON-NULL
```

B 12
15-Sep-1984 23:05:08
15-Sep-1984 22:47:04

VAX-11 Bliss-32 V4.0-742
_S255SDUA28:[MOM.SPC]NMAHEAD.B32;1

.. M 0058 0
.. 0059 0
.. 0060 0
.. 0061 0
.. 0062 0
.. 0063 0

NUL2ND (A,B)-
%NULL(B) %;

End of NMAHEAD

```
0064 0 | *****
0065 0 | Created 15-SEP-1984 23:04:56 by VAX-11 SDL V2.0 Source: 15-SEP-1984 22:46:48 _S255SDUA28:[MOM.SRC]MOMDEF.
0066 0 | *****
0067 0 |
0068 0 |
0069 0 | *** MODULE $MOMDEF ***
0070 0 |
0071 0 | Internal codes used to identify entity type of current circuit,
0072 0 | line or node.
0073 0 |
0074 0 | literal MOMSC_NODE = 0;
0075 0 | literal MOMSC_NODEBYNAME = 1;
0076 0 | literal MOMSC_CIRCUIT = 2;
0077 0 | literal MOMSC_LINE = 3;
0078 0 |
0079 0 | MOM constants used in parsing NICE messages.
0080 0 |
0081 0 | literal MOMSC_NODE_ID PARAM = 0; : Parameter is a node address and/or name
0082 0 | literal MOMSC_NODE_ADDR_PARAM = 1; : Parameter is a node address
0083 0 |
0084 0 | Flags set in MOM$GL_SERVICE_FLAGS
0085 0 |
0086 0 | literal MOMSM_AUTOSERVICE = 1;
0087 0 | literal MOMSM_NI_CIRC = 2;
0088 0 | literal MOMSM_MOP_RCV_OUT = 4;
0089 0 | literal MOMSM_LOOP_W_ASSIST = 8;
0090 0 | literal MOMSM_LOOP_W_ACCESS_CTL = 16;
0091 0 | literal MOMSM_NI_MULTICAST = 32;
0092 0 | literal MOMSM_CONSOLE_CARRIER_LOAD = 64;
0093 0 | literal MOMSM_NI_VOLUNTEERING = 128;
0094 0 | literal MOMSK_FAC_CODE = 519; : Facility code
0095 0 | literal MOMSK_SIG_CODE = 34013184; : Signal code (519^16)
0096 0 |
0097 0 | MOM constants - QIO buffer sizes, etc.
0098 0 |
0099 0 | literal MOMSK_NI_ADDR_LENGTH = 6; : Length of Ethernet addresses.
0100 0 | literal MOMSK_NICE_BUF_LEN = 197; : NICE message buffer length
0101 0 | literal MOMSK_QIO_BUF_LEN = 512; : QIO buffer length
0102 0 | literal MOMSK_P2_BUF_LEN = 104; : Max length for P2 buffers.
0103 0 | literal MOMSK_MAX_MOP_MSG_LEN = 1500; : Maximum length for a received MOP message
0104 0 | literal MOMSK_NI_PREFIX = 262314; : Standard NI prefix used for DEC Phase IV
0105 0 | routers.
0106 0 | literal MOMSK_LOADUMP_NI_PROT = 352; : NI Load/Dump Protocol type for NI driver
0107 0 | literal MOMSK_CONSOLE_NI_PROT = 608; : NI Remote Console Protocol type.
0108 0 | literal MOMSK_LOOP_NI_PROT = 144; : NI Loop Protocol type for NI driver
0109 0 | literal MOMSK_MAX_LOOP_HEADER = 30; : Maximum size the NI loop message headers
0110 0 | can be.
0111 0 | literal MOMSK_NO_LOOP_HELP = -1; : Default value for circuit loop help type.
0112 0 | literal MOMSK_SKIP_MULTICASTS = -1; : When looking for a MOP message response,
0113 0 | skip over multicasts.
0114 0 |
0115 0 | Internal loopback type codes.
0116 0 |
0117 0 | literal MOMSC_LOOP_MOP = 0; : MOP loopback operation
0118 0 | literal MOMSC_LOOP_PHASE3 = 1; : Phase III loopback operation
0119 0 | literal MOMSC_LOOP_PHASE2 = 2; : Phase II loopback operation
0120 0 | literal MOMSS_MOMDEF = 1;
```

```
0121 0 macro MOMSV_AUTOSERVICE = 0,0,1,0 %; ! Autoservice function in progress (as
0122 0 macro MOMSV_NI_CIRC = 0,1,1,0 %; ! Service function is on an Ethernet circuit.
0123 0 macro MOMSV_MOP_RCV_OUT = 0,2,1,0 %; ! There is a MOP receive outstanding.
0124 0 macro MOMSV_LOOP_W_ASSIST = 0,3,1,0 %; ! Loop an NI circuit using a third node to
0125 0 macro MOMSV_LOOP_W_ACCESS_CTL = 0,4,1,0 %; ! Loop a node with using an access control
0126 0 macro MOMSV_NI_MULTICAST = 0,5,1,0 %; ! Autoservice request received on an NI was
0127 0 macro MOMSV_CONSOLE_CARRIER_LOAD = 0,6,1,0 %; ! Down line loading console carrier code
0128 0 macro MOMSV_NI_VOLUNTEERING = 0,7,1,0 %; ! A multicast service request was received
0129 0 ! from a target on the NI. MOM is
0130 0 ! volunteering to perform the request.
0131 0 ! If the target does not respond, assume
0132 0 ! some other host was chosen by the target.
0133 0 ! (needed because the file format is
0134 0 ! different than other load files).
0135 0 ! a multicast.
0136 0 ! string (supplied in the NICE message).
0137 0 ! assist.
0138 0 ! opposed to operator requested
0139 0 ! service function.
0140 0
0141 0 MOM facility code
0142 0
0143 0
0144 0 *** MODULE $MOPDEF ***
0145 0
0146 0 MOP message definitions
0147 0
0148 0
0149 0 Define MOP function codes.
0150 0
0151 0 literal MOPS_FCT_MLT = 0; ! Load/dump Memory load with transfer address
0152 0 literal MOPS_FCT_DCM = 1; ! Dump Complete
0153 0 literal MOPS_FCT_MLD = 2; ! Load/dump Memory load
0154 0 literal MOPS_FCT_ASV = 3; ! Load/dump Assistance volunteer (NI only)
0155 0 literal MOPS_FCT_RMD = 4; ! Request memory dump
0156 0 literal MOPS_FCT_RID = 5; ! Remote Console Request ID
0157 0 literal MOPS_FCT_EMM = 6; ! Remote Console Enter MOP mode (boot)
0158 0 literal MOPS_FCT_SID = 7; ! Remote Console System ID
0159 0 literal MOPS_FCT_RPR = 8; ! Load/dump Request program
0160 0 literal MOPS_FCT_RML = 10; ! Load/dump Request memory load
0161 0 literal MOPS_FCT_RDS = 12; ! Load/dump Request Dump Service (old MOP Mode running)
0162 0 literal MOPS_FCT_MDD = 14; ! Load/dump Memory dump data
0163 0 literal MOPS_FCT_PLT = 20; ! Load/dump Parameter load with transfer address
0164 0 literal MOPS_FCT_ALD = 21; ! Active loop data
0165 0 literal MOPS_FCT_PLD = 25; ! Passive looped data
0166 0
0167 0 Define MOP parameter codes.
0168 0
0169 0 literal MOPSC_PAR_NNA = 1; ! Target node name
0170 0 literal MOPSC_PAR_NAD = 2; ! Target node address
0171 0 literal MOPSC_PAR_HNA = 3; ! Host node name
0172 0 literal MOPSC_PAR_HAD = 4; ! Host node address
0173 0 literal MOPSC_PAR_HTI = 5; ! Host system time
0174 0
0175 0 Define MOP Software ID codes for Request Program and Boot messages.
0176 0
0177 0 literal MOPSC_SID_NON = 0; ! No software ID
```

```
0178 0 Literal MOPSC_SID_OSY = 255;           ! Standard Operating System
0179 0 Literal MOPSC_SID_MAI = 254;         ! Maintenance System
0180 0
0181 0 Define MOP Processor codes for Boot messages.
0182 0
0183 0 Literal MOPSC_PRO_SYS = 0;           ! System processor
0184 0 Literal MOPSC_PRO_COM = 1;         ! Communications processor (for loading
0185 0 Console Carrier)
0186 0
0187 0 Define function codes used in NI loop messages.
0188 0
0189 0 Literal MOPSC_NILOOP_REPLY = 1;      ! Reply to looped message
0190 0 Literal MOPSC_NILOOP_FORWARD = 2;  ! Forward looping message
0191 0
0192 0 !*** MODULE $SVDDEF ***
0193 0
0194 0 Service Data
0195 0 Definitions of target node service information. This information is
0196 0 retrieved from the volatile database and, if applicable, the NICE command.
0197 0 It is saved in this table for use by Service during the service operation.
0198 0
0199 0 Literal SVDSM_MSG_PARAM = 1;
0200 0 Literal SVDSK_ENTRY_LEN = 137;      ! Length of SVD table entries
0201 0 Literal SVDSK_ENTRY_LEN = 137;      ! Length of SVD table entries
0202 0 Literal SVDSK_BYTE = 0;
0203 0 Literal SVDSK_WORD = 1;
0204 0 Literal SVDSK_LONG = 2;
0205 0 Literal SVDSK_STRING = 3;
0206 0 Literal SVDS$SVDDEF = 137;
0207 0 macro SVDSL_NFB_ID = 0,0,32,0 %;      ! NFB Field ID for QIOs to NETACP
0208 0 macro SVDSB_NFB_DATABASE = 3,0,8,0 %; ! NFB Field ID database (subfield of NFB_ID)
0209 0 macro SVDSW_NICE_ID = 4,0,16,0 %;    ! NICE message parameter ID
0210 0 macro SVDSB_NICE_TYPE = 6,0,8,0 %;   ! Parameter type in NICE messages
0211 0 macro SVDSB_FLAGS = 7,0,8,0 %;      ! Flags field
0212 0 macro SVDSV_MSG_PARAM = 7,0,1,0 %;   ! The parameter value was supplied by the
0213 0 macro SVDSB_STRING_LEN = 8,0,8,0 %;  ! Byte length of parameter value
0214 0 macro SVDST_STRING = 9,0,0,0 %;
0215 0 Literal SVD$S_STRING = 128;         ! If it's a string parameter, the string.
0216 0 macro SVDSL_PARAM = 9,0,32,0 %;     ! If it's not a string parameter, the parameter
0217 0 value.
0218 0
0219 0 Parametr type definitions
0220 0
0221 0
0222 0 !*** MODULE $MDTDEF ***
0223 0
0224 0 Mop Device Table definitions
0225 0 This table contains the ASCII device name strings associated with a
0226 0 given MOP device code. It is used to construct secondary and tertiary
0227 0 load file names when none are specified in the database. The characters
0228 0 'SEC' or 'TER' are concatenated with the device name string to construct
0229 0 a load file name.
0230 0
0231 0 Literal MDT$K_ENTRYLEN = 5;
0232 0 Literal MDT$C_ENTRYLEN = 5;
0233 0 Literal MDT$S_MDTDEF = 5;
0234 0 macro MDT$B_DEVTYPE = 0,0,8,0 %;
```

```
0235 0 macro MDT$A_DEVSTRING = 1,0,32,0 %;
0236 0
0237 0 !*** MODULE $CIBDEF ***
0238 0
0239 0 Channel Information Block (CIB)
0240 0
0241 0 Literal CIB$M_TARGET_ADDR_FIXED = 1;
0242 0 Literal CIB$K_CIBLEN = 76;
0243 0 Literal CIB$C_CIBLEN = 76;
0244 0 address and protocol to use (among
0245 0 other things)
0246 0 Literal CIB$S_CIBDEF = 76;
0247 0 macro CIB$L_CHAN = 0,0,32,0 %;
0248 0 macro CIB$T_NI_HIORD_ADDR = 4,0,0,0 %;
0249 0 Literal CIB$S_NI_HIORD_ADDR = 6; ! HIORD NI address of target
0250 0 macro CIB$L_NI_HIORD_PREF = 4,0,32,0 %; ! HIORD NI prefix assigned to Digital
0251 0 macro CIB$W_NI_HIORD_NODE = 8,0,16,0 %; ! Node address withing HIORD NI address of target
0252 0 macro CIB$T_NI_HARDWR_ADDR = 10,0,0,0 %;
0253 0 Literal CIB$S_NI_HARDWR_ADDR = 6; ! Hardware NI address of target
0254 0 macro CIB$W_FLAGS = 16,0,16,0 %; ! Flags field
0255 0 macro CIB$V_TARGET_ADDR_FIXED = 16,0,1,0 %; ! Set when there is no alternate address
0256 0 to try in order to communicate with
0257 0 the target. Applies to any service
0258 0 circuit, but is most useful on the NI
0259 0 where the target could be responding to
0260 0 either a hardware address or a DECnet
0261 0 physical address.
0262 0 macro CIB$L_RETRY_CNT = 18,0,32,0 %; ! Transmit retry count for transaction.
0263 0 macro CIB$T_SETMODE_P2_BUF = 22,0,0,0 %;
0264 0 Literal CIB$S_SETMODE_P2_BUF = 54; ! P2 buffer for telling NI driver which NI
0265 0 macro CIB$L_P2_BUF_SIZE = 24,0,32,0 %; ! MOP message buffer size
0266 0 macro CIB$L_P2_PADDING = 30,0,32,0 %; ! Put pad count on NI messages.
0267 0 macro CIB$L_P2_PROTOCOL = 48,0,32,0 %; ! NI Protocol to use.
0268 0 macro CIB$T_NI_PHYS_ADDR = 70,0,8,0 %; ! Target NI physical address
0269 0
0270 0 !*** MODULE $NIHDEF ***
0271 0
0272 0 NI header definitions.
0273 0 The NI device driver (XEDRIVER) returns the NI header for a message
0274 0 in the P3 parameter. There are several cases in which MOM needs to
0275 0 get this header in order to find out which target on the a message
0276 0 came from or whether or not it was sent to a multicast address.
0277 0
0278 0 Literal NIH$K_NI_HEADER_LEN = 14;
0279 0 Literal NIH$C_NI_HEADER_LEN = 14;
0280 0 Literal NIH$S_NIHDEF = 14;
0281 0 macro NIH$T_DEST_NI_ADDR = 0,0,0,0 %;
0282 0 Literal NIH$S_DEST_NI_ADDR = 6; ! destination Ni address of message
0283 0 macro NIH$B_MULTICAST = 0,0,8,0 %; ! Low bit set if destination is a multicast
0284 0 macro NIH$T_SOURCE_NI_ADDR = 6,0,0,0 %;
0285 0 Literal NIH$S_SOURCE_NI_ADDR = 6; ! Source Ni address of message
0286 0 macro NIH$W_PROTOCOL_TYPE = 12,0,16,0 %; ! Protocol type of message
0287 0
0288 0 !*** MODULE $MSBDEF ***
0289 0
0290 0 Message segment block (MSB) definitions. The MSB is used to accumulate
0291 0 information with which to build the NICE response message returned to NCP
```



```
0292 0      | when the operation is completed.
0293 0
0294 0      | literal MSBSM_CODE_FLD = 1;
0295 0      | literal MSBSM_DET_FLD = 2;
0296 0      | literal MSBSM_MSG_FLD = 4;
0297 0      | literal MSBSM_MSG2_FLD = 8;
0298 0      | literal MSBSM_ENTD_FLD = 16;
0299 0      | literal MSBSM_DATA_FLD = 32;
0300 0      | literal MSBSK_LENGTH = 28;      | Maximum MSB size
0301 0      | literal MSBSC_LENGTH = 28;      | Maximum MSB size
0302 0      | literal MSBSS_MSBDEF = 28;
0303 0      | macro MSBSL_FLAGS = 0,0,32,0 %;  | Flags
0304 0      | macro MSBSV_CODE_FLD = 0,0,1,0 %; | Status code present (not used)
0305 0      | macro MSBSV_DET_FLD = 0,1,1,0 %; | Error detail field present (DETAIL)
0306 0      | macro MSBSV_MSG_FLD = 0,2,1,0 %; | Message text field present (TEXT)
0307 0      | macro MSBSV_MSG2_FLD = 0,3,1,0 %; | Second line of message text present (TEXT2)
0308 0      | macro MSBSV_ENTD_FLD = 0,4,1,0 %; | Entity descriptor field present (ENTITY)
0309 0      | macro MSBSV_DATA_FLD = 0,5,1,0 %; | Data descriptor field present (DATA)
0310 0      | macro MSBSB_CODE = 4,0,8,0 %;    | Status code
0311 0      | macro MSBSW_DETAIL = 8,0,16,0 %; | Detail
0312 0      | macro MSBSL_TEXT = 12,0,32,0 %;  | Status code for text message.
0313 0      | macro MSBSL_TEXT2 = 16,0,32,0 %; | Status code for second line of text msg.
0314 0      | macro MSBSA_ENTITY = 20,0,32,0 %; | Entity descriptor address
0315 0      | macro MSBSA_DATA = 24,0,32,0 %;  | Data descriptor address
0316 0
0317 0      | *** MODULE $DBGDEF ***
0318 0
0319 0      | NML internal logging (debugging) flags
0320 0
0321 0      | These flags are used to enable logging of specified data to the NML log
0322 0      | file. The flags are defined by translating the logical name NML$LOG, which
0323 0      | MOM uses also. The permanent database and event logging flags are not
0324 0      | used by MOM.
0325 0
0326 0      | literal DBGSC_NETIO = 0;          | Network send/receive logging
0327 0      | literal DBGSC_FILEIO = 1;        | File read/write logging
0328 0      | literal DBGSC_NPARSE = 2;        | NPARSE state transition logging
0329 0      | literal DBGSC_LOOPIO = 3;        | Loopback transmit/receive logging
0330 0      | literal DBGSC_ACPQIO = 4;        | NETACP QIO logging
0331 0      | literal DBGSC_MOPIO = 5;         | MOP send/receive logging - see MOP_xxx flags below.
0332 0      | literal DBGSC_SRVTRC = 6;        | Trace service operations
0333 0      | literal DBGSC_EVENTS = 7;        | Network event (EVL) logging
0334 0
0335 0      | The following flags must be set in conjunction with the DBGSC_MOPIO
0336 0      | flag.
0337 0
0338 0      | literal DBGSC_MOP_OTHER = 8;     | All other MOP send/receive messages
0339 0      | literal DBGSC_MOP_MLD = 9;       | MOP Memory Load Data messages (MOPS_FCT_MLD)
0340 0      | literal DBGSC_MOP_RML = 10;      | MOP Request Memory Load messages (MOPS_FCT_RML)
0341 0      | literal DBGSC_MOP_RMD = 11;      | MOP Request Memory Dump messages (MOPS_FCT_RMD)
0342 0      | literal DBGSC_MOP_MDD = 12;      | MOP Memory Dump Data messages (MOPS_FCT_MDD)
```

0343 0
0344 0
0345 0
0346 0
0347 0
0348 0
0349 0
0350 0
0351 0
0352 0
0353 0
0354 0
0355 0
0356 0
0357 0
0358 0
0359 0
0360 0
0361 0
0362 0
0363 0
0364 0
0365 0
0366 0
0367 0
0368 0
0369 0
0370 0
0371 0
0372 0
0373 0
0374 0
0375 0
0376 0
0377 0
0378 0
0379 0
0380 0
0381 0
0382 0
0383 0
0384 0
0385 0
0386 0
0387 0
0388 0
0389 0
0390 0
0391 0
0392 0
0393 0
0394 0
0395 0
0396 0
0397 0
0398 0
0399 0

%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;

```
0400 0
0401 0
0402 0
0403 0
0404 0
0405 0
0406 0
0407 0
0408 0
0409 0
0410 0
0411 0
0412 0
0413 0
0414 0
0415 0
0416 0
M 0417 0
M 0418 0
0419 0
```

```
Structure declarations used for system defined structures to
save typing.
STRUCTURE
  BBLOCK [O, P, S, E; N] =
    [N]
    (BBLOCK+0)<P,S,E>.
  BBLOCKVECTOR [I, O, P, S, E; N, BS] =
    [N*BS]
    ((BBLOCKVECTOR+I*BS)+0)<P,S,E>;
Macro to signal status message
MACRO
  $SIGNAL MSG [] =
  SIGNAL (MOMSK_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 ('NFBSC_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 ('NFBSC_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 (NFBSC_ENDOFLIST) ! End delimiter for field ID list.  
  %ELSE  
        LONG (NFBSC_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 NFBSC_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

0420 0
0421 0
0422 0
0423 0
0424 0
0425 0
0426 0
0427 0
0428 0
0429 0
0430 0
0431 0
0432 0
0433 0
0434 0
0435 0
0436 0
0437 0
0438 0
0439 0
0440 0
0441 0
0442 0
0443 0
0444 0
0445 0
0446 0
0447 0
0448 0
0449 0
0450 0
0451 0
0452 0
0453 0
0454 0
0455 0
0456 0
0457 0
0458 0
0459 0
0460 0
0461 0
0462 0
0463 0
0464 0
0465 0
0466 0
0467 0
0468 0
0469 0
0470 0
0471 0
0472 0
0473 0
0474 0
0475 0
0476 0

0477 0
0478 0
M 0479 0
0480 0
0481 0
0482 0
E 0483 0
0484 0
0485 0
0486 0
0487 0
0488 0
0489 0
0490 0
0491 0
0492 0
E 0493 0
0494 0
0495 0
0496 0
0497 0
0498 0
0499 0
0500 0
0501 0
0502 0
0503 0
E 0504 0
0505 0
0506 0
0507 0
0508 0
0509 0
0510 0
0511 0
0512 0
0513 0
0514 0
E 0515 0
0516 0
0517 0
0518 0

```
... field IDs as are supplied.
SFIELD ID LIST (DATABASE) [FIELD ID] =
  LONG T$FIELD_ID (DATABASE, FIELD_ID)
  %,
SFIELD ID (DATABASE, FIELD_ID) =
  %IF %IDENTICAL (FIELD_ID, NFB$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.
$NFB$DSC (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 [] =
  S (MAX(0, %LENGTH-6))
  %:
```

0519 0
0520 0
0521 0
0522 0
0523 0
0524 0
0525 0
0526 0
0527 0
0528 0
0529 0
0530 0
M 0531 0
M 0532 0
M 0533 0
M 0534 0
M 0535 0
M 0536 0
M 0537 0
M 0538 0
M 0539 0
M 0540 0
M 0541 0
M 0542 0
M 0543 0
M 0544 0
M 0545 0
M 0546 0
M 0547 0
M 0548 0
M 0549 0
M 0550 0
M 0551 0
M 0552 0
M 0553 0
M 0554 0
M 0555 0
M 0556 0
M 0557 0
M 0558 0
M 0559 0
M 0560 0
M 0561 0
M 0562 0
M 0563 0
M 0564 0
M 0565 0
M 0566 0
M 0567 0
M 0568 0
M 0569 0
M 0570 0
M 0571 0
M 0572 0
M 0573 0
M 0574 0
M 0575 0

```

: 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)
  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 $MOM EXTERNALS =
  EXTERNAL
    MOM$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],
```

0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632

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

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_SHNA,
SVDSGK_PCNO_SHHW,
SVDSGK_PCNO_SFTY,
SVDSGK_PCNO_PHA,
SVDSGK_PCNO_SDA,
SVDSGK_PCNO_LPC,
SVDSGK_PCNO_LPL,
SVDSGK_PCNO_LPD,
SVDSGK_PCNO_LPH,
SVDSGK_PCNO_LPA,
SVDSGK_PCNO_LPN,
SVDSGK_PCNO_SLNA,
SVDSGK_PCNO_SLNH,
SVDSGK_PCNO_LAN,
SVDSGK_PCNO_SLNN,
SVDSGK_PCNO_SLAH,
SVDSGK_PCLI_STI,
SVDSC_ENTRY_COUNT;

z:

NPARSE argument block structure definitions
MACRO
\$NPA_ARGDEF =
BUILTIN
AP;

0644 0
0645 0
0646 0
0647 0
0648 0
0649 0
0650 0
0651 0
0652 0
0653 0
0654 0
0655 0
0656 0
0657 0
0658 0
0659 0
0660 0
0661 0
0662 0
0663 0
0664 0

Created 15-SEP-1984 23:05:03 by VAX-11 SDL V2.0 Source: 15-SEP-1984 22:46:56 _S255\$DUA28:[MOM.SRC]NPADEF.

```
!*** MODULE $NPADEF ***
literal NPASK_COUNT0 = 8;           | Argument count value
literal NPASK_LENGTH0 = 36;        | Size of argument block structure
literal NPASC_LENGTH0 = 36;        | Size of argument block structure
literal NPASS_NPAD = 36;
macro NPASL_COUNT = 0,0,32,0 %;     | Argument count (NPASK_COUNT0)
macro NPASL_MSGCNT = 4,0,32,0 %;    | Count of bytes remaining in message
macro NPASL_MSGPTR = 8,0,32,0 %;    | Pointer to remaining message
macro NPASL_OPTIONS = 12,0,32,0 %;  | Options (not used)
macro NPASL_FLDCNT = 16,0,32,0 %;   | Count of bytes in matched field
macro NPASL_FLDPTR = 20,0,32,0 %;   | Pointer to matched field
macro NPASL_LONG = 24,0,32,0 %;     | Matched longword value
macro NPASW_WORD = 24,0,16,0 %;     | word value
macro NPASB_BYTE = 24,0,8,0 %;      | byte value
macro NPASL_NUMBER = 28,0,32,0 %;   | Matched signed value (not used)
macro NPASL_PARAM = 32,0,32,0 %;    | Action routine parameter value
```

0665 0
0666 0
0667 0
0668 0
0669 0
0670 0
0671 0
0672 0
0673 0
0674 0
0675 0
0676 0
0677 0
0678 0
0679 0
0680 0
0681 0
0682 0
0683 0
0684 0
0685 0
0686 0
0687 0
0688 0
0689 0
0690 0
0691 0
0692 0
0693 0
0694 0
0695 0
0696 0
0697 0
0698 0
0699 0
0700 0
0701 0
0702 0
0703 0
0704 0
0705 0
0706 0
0707 0

Version: '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.  
*  
*****
```

```
++  
NMATAIL.B32  
  
Source to undeclare the macros required for the precompile of  
NMALIBRY.B32 so they do not appear in the library.  
--
```

```
UNDECLARE %QUOTE $EQLST,  
%QUOTE GET1ST_  
%QUOTE GET2ND_  
%QUOTE NUL2ND_  
;
```

End of NMATAIL.B32

COMMAND QUALIFIERS

BLISS/LIBRARY=LIBS:MOMLIB/LIST=LISS:MOMLIB SRC\$:NMAHEAD+LIBS:MOMDEF+SRCS:MOMDDL+LIBS:NPADEF+SRCS:NMATAIL

Run Time: 00:05.6
Elapsed Time: 00:10.9
Lines/CPU Min: 7643

: Lexemes/CPU-Min: 28518
: Memory Used: 54 pages
: Library Precompilation Complete

