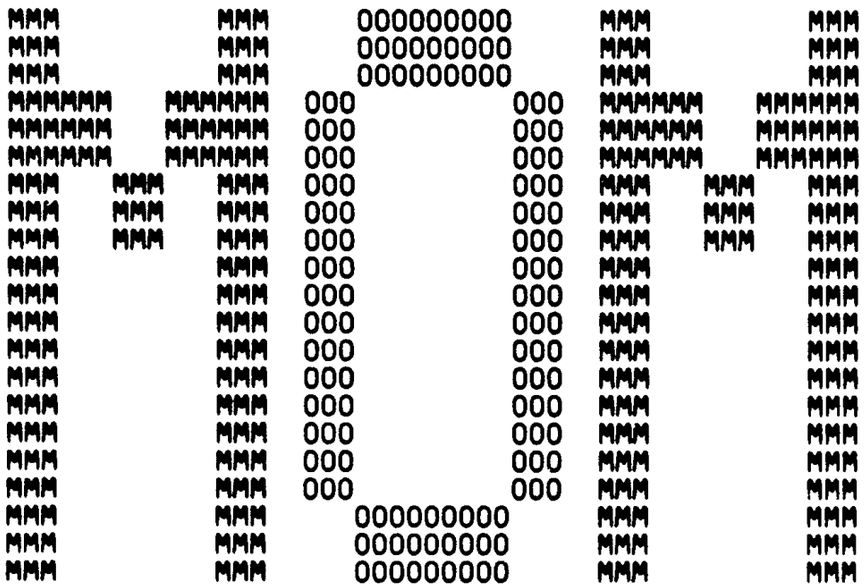

B C D E F G H I J K L M N B C D E F G H I J K L M N B C D E F G H I J K L M N B C D E F G H I

```
MM      MM    000000    MM      MM  DDDDDDDD      AAAAAA    TTTTTTTTTT
MM      MM    000000    MM      MM  DDDDDDDD      AAAAAA    TTTTTTTTTT
MMMM  MMMM  00      00  MMMM  MMMM  DD      DD  AA      AA      TT
MMMM  MMMM  00      00  MMMM  MMMM  DD      DD  AA      AA      TT
MM  MM  MM  00      00  MM  MM  MM  DD      DD  AA      AA      TT
MM  MM  MM  00      00  MM  MM  MM  DD      DD  AA      AA      TT
MM      MM  00      00  MM      MM  DD      DD  AA      AA      TT
MM      MM  00      00  MM      MM  DD      DD  AA      AA      TT
MM      MM  00      00  MM      MM  DD      DD  AAAAAAAAAA      TT
MM      MM  00      00  MM      MM  DD      DD  AAAAAAAAAA      TT
MM      MM  00      00  MM      MM  DD      DD  AA      AA      TT          ....
MM      MM  00      00  MM      MM  DD      DD  AA      AA      TT          ....
MM      MM    000000    MM      MM  DDDDDDDD    AA      AA      TT          ....
MM      MM    000000    MM      MM  DDDDDDDD    AA      AA      TT          ....

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 MODULE MOMDAT (IDENT = 'V04-000') =
;     2          0002  1 BEGIN
;     3          0003  1 !
;     4          0004  1 !**************************************************************************
;     5          0005  1 !*                                                                        *
;     6          0006  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;     7          0007  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
;     8          0008  1 !*  ALL RIGHTS RESERVED.                                                  *
;     9          0009  1 !*                                                                        *
;    10          0010  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    11          0011  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    12          0012  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;    13          0013  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    14          0014  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    15          0015  1 !*  TRANSFERRED.                                                           *
;    16          0016  1 !*                                                                        *
;    17          0017  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;    18          0018  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;    19          0019  1 !*  CORPORATION.                                                           *
;    20          0020  1 !*                                                                        *
;    21          0021  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    22          0022  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    23          0023  1 !*                                                                        *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !**************************************************************************
;    26          0026  1 !
;    27          0027  1
;    28          0028  1 !++
;    29          0029  1 ! FACILITY:  DECnet-VAX Network Maintenanc Operations Module
;    30          0030  1 !
;    31          0031  1 !
;    32          0032  1 ! ABSTRACT:
;    33          0033  1 !       This module contains all global data referenced by the
;    34          0034  1 !       Maintenance Operations Module (MOM).
;    35          0035  1 !
;    36          0036  1 ! ENVIRONMENT:  VAX/VMS Operating System
;    37          0037  1 !
;    38          0038  1 ! AUTHOR:  Kathy Perko
;    39          0039  1 !
;    40          0040  1 ! CREATION DATE:  17-Dec-1982
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !       V03-004 MKP0004         Kathy Perko             21-July-1984
;    44          0044  1 !               Use MOM$K_MAX_MOP_MSG_LEN instead of literals in descriptors.
;    45          0045  1 !               This falls out as part of fix for LOOP CIRC on point-to-point
;    46          0046  1 !               lines.
;    47          0047  1 !
;    48          0048  1 !       V03-003 MKP0003         Kathy Perko             20-May-1984
;    49          0049  1 !               Add QNA device to table used to construct secondary and
;    50          0050  1 !               tertiary load file names which are not suppied in the node
;    51          0051  1 !               database.
;    52          0052  1 !
;    53          0053  1 !       V03-002 MKP0002         Kathy Perko             11-April-1984
;    54          0054  1 !               Add buffer for Network Management version checking.
;    55          0055  1 !
;    56          0056  1 !       V03-001 MKP0001         Kathy Perko             20-Jan-1984
;    57          0057  1 !               Add SERVICE NODE VERSION parameter.
```

```
;   58          0058  1 !
;   59          0059  1 !--
```

```
;    61          0060  1 %SBTTL 'Global data declarations'
;    62          0061  1
;    63          0062  1 !
;    64          0063  1 ! INCLUDE FILES:
;    65          0064  1 !
;    66          0065  1
;    67          0066  1 LIBRARY 'LIB$:MOMLIB.L32';
;    68          0067  1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
;    69          0068  1 LIBRARY 'SHRLIB$:NET.L32';
;    70          0069  1 LIBRARY 'SYS$LIBRARY:STARLET.L32';
;    71          0070  1
;    72          0071  1 PSECT GLOBAL = $GLOBAL$;
;    73          0072  1
;    74          0073  1 !
;    75          0074  1 ! OWN STORAGE:
;    76          0075  1 !
;    77          0076  1 GLOBAL
;    78          0077  1     MOM$GQ_PROPRVMSK : BBLOCK [8],              ! Process privilege mask
;    79          0078  1     MOM$GW_ACP_CHAN;                            ! ACP control channel
;    80          0079  1
;    81          0080  1 !
;    82          0081  1 ! Debugging log mask.  The bit mask is set up at service initialization
;    83          0082  1 ! by translating the logical name MOM$LOG.  The resulting ASCII hex number
;    84          0083  1 ! is converted to binary to provide the appropriate mask bit settings.
;    85          0084  1 !
;    86          0085  1 ! The values for MOM$LOG are defined as follows:
;    87          0086  1 !
;    88          0087  1 !           1   NICE message network I/O.
;    89          0088  1 !           4   NPARSE state transitions.
;    90          0089  1 !           8   Test (node loopback) message network I/O.
;    91          0090  1 !          10   Volatile data base I/O (NETACP QIOs).
;    92          0091  1 !          20   MOP direct line I/O.
;    93          0092  1 !          40   Trace service operation.
;    94          0093  1 !          80   Raw event data.
;    95          0094  1 !
;    96          0095  1 GLOBAL
;    97          0096  1     MOM$GL_LOGMASK    : BLOCK [1] INITIAL (0);    ! Internal logging mask
;    98          0097  1
;    99          0098  1
```

```
;  101          0099  1 %SBTTL 'Data for service operations'
;  102          0100  1 !
;  103          0101  1 ! The following data is used to store information needed for maintenance
;  104          0102  1 ! operations such as LOAD, DUMP, TRIGGER, and line loop.
;  105          0103  1 !
;  106          0104  1
;  107          0105  1 GLOBAL BIND
;  108          0106  1 !
;  109          0107  1 ! Network device name - used to assign a channel to NETACP for getting
;  110          0108  1 ! information from the volatile database.
;  111          0109  1 !
;  112          0110  1     MOM$GQ_NETNAMDSC = $ASCID ('_NET:'),
;  113          0111  1 !
;  114          0112  1 ! Service device name - used to assign a channel to the device.  QIOs to
;  115          0113  1 ! this device will send MOP messages to the target node and receive the
;  116          0114  1 ! response MOP messages.
;  117          0115  1 !
;  118          0116  1     MOM$GQ_DLE_NAMDSC = $ASCID ('_ND:'),
;  119          0117  1 !
;  120          0118  1 ! PSI device name - used to assign a channel to PSI for issuing loop
;  121          0119  1 ! line QIOs.
;  122          0120  1 !
;  123          0121  1     MOM$GQ_PSINAMDSC = $ASCID ('_NW:');
;  124          0122  1
;  125          0123  1 !
;  126          0124  1 ! The following fields are used for parsing NICE commands requesting
;  127          0125  1 ! service operations.
;  128          0126  1 !
;  129          0127  1 GLOBAL
;  130          0128  1     MOM$GL_SVD_INDEX,                   ! Index for parameter's entry in the
;  131          0129  1                                         !       Service Data Table.
;  132          0130  1     MOM$GB_FUNCTION:    BYTE,           ! NICE message function code.
;  133          0131  1     MOM$GB_OPTION_BYTE: BYTE;           ! NICE message option byte.
;  134          0132  1
;  135          0133  1 !
;  136          0134  1 ! NPARSE argument block - this block is used during parsing of NICE messages
;  137          0135  1 ! to keep track of how far into the message the parsing is, and the value and
;  138          0136  1 ! length of the field currently being parsed.
;  139          0137  1 !
;  140          0138  1 GLOBAL
;  141          0139  1     MOM$AB_NPARSE_BLK:  $NPA_BLKDEF;
;  142          0140  1
;  143          0141  1 GLOBAL
;  144          0142  1 !
;  145          0143  1 ! The maintenance entity code can be any one of the following values:
;  146          0144  1 !
;  147          0145  1 !       MOM$C_LINE
;  148          0146  1 !       MOM$C_CIRCUIT
;  149          0147  1 !       MOM$C_NODE
;  150          0148  1 !       MOM$C_NODEBYNAME
;  151          0149  1 !
;  152          0150  1     MOM$GB_ENTITY_CODE : BYTE,          ! Maintenance entity code (key)
;  153          0151  1 !
;  154          0152  1 ! The entity id string is the data used as the key into the volatile data
;  155          0153  1 ! base to get information for the maintenance operation.  The contents of the
;  156          0154  1 ! buffer are determined by the value of the entity id code.
;  157          0155  1 !
```

```
;  158     0156  1 !     MOM$C_LINE          contains line name.
;  159     0157  1 !     MOM$C_CIRCUIT       contains circuit name.
;  160     0158  1 !     MOM$C_NODE          contains node address (always a word).
;  161     0159  1 !     MOM$C_NODEBYNAME    contains node name.
;  162     0160  1 !
;  163     0161  1     MOM$AB_ENTITY_BUF    : BBLOCK [32],  ! Entity id string buffer
;  164     0162  1 !
;  165     0163  1 ! The service id descriptor describes the extent of the entity id in
;  166     0164  1 ! the service id buffer.
;  167     0165  1 !
;  168     0166  1     MOM$GQ_ENTITY_BUF_DSC : VECTOR [2]  ! Maintenance id descriptor
;  169     0167  1                             INITIAL (0, MOM$AB_ENTITY_BUF);
;  170     0168  1
;  171     0169  1 !
;  172     0170  1 ! Service flags.  These flags are set to indicate various options in
;  173     0171  1 ! use by the current service operation.  The options bits are described
;  174     0172  1 ! in MOMDEF.MDL.
;  175     0173  1 !
;  176     0174  1 GLOBAL
;  177     0175  1     MOM$GL_SERVICE_FLAGS;
;  178     0176  1
;  179     0177  1 !
;  180     0178  1 ! For autoservice functions, MOM logs events to indicate the status
;  181     0179  1 ! of the operation.  This serves the same function as the NICE response
;  182     0180  1 ! message for operator service functions.  The event to logged is kept
;  183     0181  1 ! in the following fields, and when completion (successful or not) is
;  184     0182  1 ! signalled, the event is logged by the condition handler.
;  185     0183  1 !
;  186     0184  1 ! Three different events can be logged:
;  187     0185  1 !
;  188     0186  1 !       Automatic line service          0.3
;  189     0187  1 !       Aborted service request         0.7
;  190     0188  1 !       Passive loopback                0.6
;  191     0189  1 !
;  192     0190  1 GLOBAL
;  193     0191  1     MOM$GB_EVT_POPR : BYTE,             ! Passive loopback operation code
;  194     0192  1     MOM$GB_EVT_PRSN : BYTE,             ! Aborted service request reason code
;  195     0193  1     MOM$GB_EVT_PSER : BYTE,             ! Automatic line service request code
;  196     0194  1     MOM$GW_EVT_CODE : WORD;             ! Event code
;  197     0195  1
;  198     0196  1
```

```
;  200     0197  1 !**************************************************************************
;  201     0198  1 !   Buffers for communicating with other components of DECnet:
;  202     0199  1 !       NICE message buffers
;  203     0200  1 !       MOP message buffers
;  204     0201  1 !       NETACP QIO buffers
;  205     0202  1 !**************************************************************************
;  206     0203  1
;  207     0204  1 !
;  208     0205  1 ! Network I/O buffers used for sending and receiving NICE messages from
;  209     0206  1 ! NCP via the Network Management Listener (NML).
;  210     0207  1 !
;  211     0208  1 GLOBAL LITERAL
;  212     0209  1     MOM$K_NML_MBX_BUF_LEN = MOM$K_NICE_BUF_LEN + 3;
;  213     0210  1
;  214     0211  1 GLOBAL
;  215     0212  1     MOM$AB_NML_MAILBOX_BUFFER: BBLOCK [MOM$K_NML_MBX_BUF_LEN];
;  216     0213  1 GLOBAL BIND
;  217     0214  1     MOM$AB_NCP_VERSION = MOM$AB_NML_MAILBOX_BUFFER :    BBLOCK [3],
;  218     0215  1     MOM$AB_NICE_RCV_BUF = MOM$AB_NML_MAILBOX_BUFFER + 3 :
;  219     0216  1                                           BBLOCK [MOM$K_NICE_BUF_LEN];
;  220     0217  1 GLOBAL
;  221     0218  1     MOM$GL_NICE_RCV_MSG_LEN,
;  222     0219  1     MOM$AB_NICE_XMIT_BUF:BBLOCK [MOM$K_NICE_BUF_LEN];
;  223     0220  1
;  224     0221  1 GLOBAL BIND
;  225     0222  1     MOM$GQ_NICE_RCV_BUF_DSC =
;  226     0223  1         UPLIT (MOM$K_NICE_BUF_LEN, MOM$AB_NICE_RCV_BUF),
;  227     0224  1     MOM$GQ_NICE_XMIT_BUF_DSC =
;  228     0225  1         UPLIT (MOM$K_NICE_BUF_LEN, MOM$AB_NICE_XMIT_BUF);
;  229     0226  1
;  230     0227  1 !
;  231     0228  1 ! P4 QIO buffer used to get the target's service parameters from NETACPs
;  232     0229  1 ! volatile database.  NETACP returns the parameters in this buffer.
;  233     0230  1 !
;  234     0231  1 GLOBAL
;  235     0232  1     MOM$AB_ACPQIO_BUFFER:       BBLOCK [MOM$K_QIO_BUF_LEN];
;  236     0233  1 GLOBAL BIND
;  237     0234  1     MOM$GQ_ACPQIO_BUF_DSC =
;  238     0235  1         UPLIT (MOM$K_QIO_BUF_LEN, MOM$AB_ACPQIO_BUFFER);
;  239     0236  1
;  240     0237  1
;  241     0238  1 !
;  242     0239  1 ! MOP I/O Channel Information Blocks (CIBs), buffers, and descriptors.
;  243     0240  1 !
;  244     0241  1 GLOBAL
;  245     0242  1     MOM$GQ_TIMEOUT:     VECTOR [2]              ! Timer set on all MOP QIOs
;  246     0243  1                         INITIAL (0, -1),        ! to target (delta).
;  247     0244  1     MOM$AB_CIB : BBLOCK [CIB$C_CIBLEN],
;  248     0245  1     MOM$AB_LOOP_CIB : BBLOCK [CIB$C_CIBLEN];
;  249     0246  1
;  250     0247  1 GLOBAL BIND
;  251     0248  1     MOM$AB_TRIGGER_CIB = MOM$AB_LOOP_CIB : BBLOCK;
;  252     0249  1
;  253     0250  1 GLOBAL
;  254     0251  1     MOM$AB_MOP_XMIT_BUF: BBLOCK [MOM$K_MAX_MOP_MSG_LEN], ! Transmit buffer
;  255     0252  1     MOM$AB_MOP_RCV_BUF : BBLOCK [MOM$K_MAX_MOP_MSG_LEN], ! Receive buffer
;  256     0253  1     MOM$AB_MOP_MSG     : BBLOCK [MOM$K_MAX_MOP_MSG_LEN], ! Received MOP
```

```
;  257     0254  1                                        !       message buffer.
;  258     0255  1      MOM$GQ_MOP_MSG_DSC : VECTOR [2];    ! Received MOP message descriptor.
;  259     0256  1
;  260     0257  1  GLOBAL BIND
;  261     0258  1      MOM$GQ_MOP_XMIT_BUF_DSC  =
;  262     0259  1                              UPLIT (MOM$K_MAX_MOP_MSG_LEN, MOM$AB_MOP_XMIT_BUF)
;  263     0260  1                              : VECTOR [2],
;  264     0261  1      MOM$GQ_MOP_RCV_BUF_DSC  =
;  265     0262  1                              UPLIT (MOM$K_MAX_MOP_MSG_LEN, MOM$AB_MOP_RCV_BUF)
;  266     0263  1                              : VECTOR [2];
;  267     0264  1
;  268     0265  1  !
;  269     0266  1  ! The following structure is used for accumulating the information
;  270     0267  1  ! to be put into the NICE response message returned to NCP.
;  271     0268  1  ! MOM$BLD_REPLY is called with this block as input.  MOM$BLD_REPLY
;  272     0269  1  ! then constructs the response message.
;  273     0270  1  !
;  274     0271  1  GLOBAL
;  275     0272  1      MOM$AB_MSGBLOCK .BBLOCK [MSB$K_LENGTH];
```

```
;  277          0273  1 !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;  278          0274  1 !
;  279          0275  1 !   Service Data Table
;  280          0276  1 !       For any MOP maintenance operation, certain node and circuit
;  281          0277  1 !   parameters are need.  These parameters are retrieved from the
;  282          0278  1 !   volatile database saved in this table.  Then, if there is a NICE
;  283          0279  1 !   command, any parameters specified there overwrite the ones from the
;  284          0280  1 !   volatile database.  These parameters are then used to perform the
;  285          0281  1 !   requested service function.
;  286          0282  1 !
;  287          0283  1 !   Each parameter's entry in the Service Data Table contains the following
;  288          0284  1 !   information:
;  289          0285  1 !       SVD$L_NFB_ID - The NFB field ID (used to identify the parameter to
;  290          0286  1 !                      NETACP).
;  291          0287  1 !       SVD$W_NICE_ID - The NICE parameter ID (used to identify the parameter
;  292          0288  1 !                       in the command from NCP).
;  293          0289  1 !       SVD$B_NICE_TYPE - The parameter's type (byte, word, longword, or
;  294          0290  1 !                         string) in the NICE message.
;  295          0291  1 !       SVD$B_FLAGS -   There's only one flag, SVD$M_MSG_PARAM, which is set
;  296          0292  1 !                       if the parameter value in this entry was obtained from
;  297          0293  1 !                       the NICE or MOP message specifying parameters for the
;  298          0294  1 !                       current operation.
;  299          0295  1 !       SDV$B_STRING_LEN - Byte length of the parameter if it's a string.
;  300          0296  1 !       SDV$L_PARAM - The parameter value.
;  301          0297  1 !       SVD$T_STRING - The string.
;  302          0298  1 !
;  303          0299  1 !----------------------------------------------------------------------------
;  304          0300  1
;  305          0301  1 !
;  306          0302  1 ! Macro to generate an entry for a parameter in the Service Data Table.
;  307          0303  1 !
;  308        M 0304  1 MACRO SERVICE_TAB (ENTITY) [PARAM_ID, NFB_DATABASE, PARAM_TYPE] =
;  309        M 0305  1
;  310        M 0306  1         [SVD_INDEX, SVD$L_NFB_ID] =
;  311        M 0307  1                         %IF %NULL (NFB_DATABASE)
;  312        M 0308  1                         %THEN 0
;  313        M 0309  1                         %ELSE %NAME ('NFB$C_',NFB_DATABASE,'_',PARAM_ID)
;  314        M 0310  1                         %FI,
;  315        M 0311  1         [SVD_INDEX, SVD$W_NICE_ID] = %NAME ('NMA$C_',ENTITY,'_',PARAM_ID),
;  316        M 0312  1         [SVD_INDEX, SVD$B_NICE_TYPE] = %NAME ('SVD$K_', PARAM_TYPE)
;  317        M 0313  1
;  318        M 0314  1         %ASSIGN (SVD_INDEX, SVD_INDEX+1)
;  319        M 0315  1
;  320          0316  1         %;
;  321          0317  1
;  322          0318  1 !
;  323          0319  1 ! Generate the Service Data Table indices used by the NPARSE tables.
;  324          0320  1 !
;  325        M 0321  1 MACRO SVD_INDEX_GEN (ENTITY) [PARAM_ID, NFB_DATABASE, PARAM_TYPE] =
;  326        M 0322  1
;  327        M 0323  1     GLOBAL LITERAL
;  328        M 0324  1         %NAME ('SVD$GK_',ENTITY,'_',PARAM_ID) = SVD_INDEX;
;  329        M 0325  1     %ASSIGN (SVD_INDEX, SVD_INDEX+1)
;  330        M 0326  1
;  331          0327  1     %;
;  332          0328  1
;  333          0329  1 COMPILETIME
```

```
;  334          0330  1      SVD_INDEX = 0;
;  335          0331  1
;  336          0332  1
;  337          0333  1 !
;  338          0334  1 ! Some of the entries in the Service Data table do not have convenient entries
;  339          0335  1 ! in the volatile database or in the NICE protocol.  They are useful pieces of
;  340          0336  1 ! information to keep around during the maintenance operation.  So pseudo
;  341          0337  1 ! names are used for their entries in the SVD.
;  342          0338  1 !
;  343          0339  1 ! The values all have bit 15 set, indicating a counter value, to avoid
;  344          0340  1 ! conflicts with other network management parameter codes.
;  345          0341  1 !
;  346          0342  1 GLOBAL LITERAL
;  347          0343  1     NMA$C_PCNO_$HNA = 1 ^ 15 OR 0,      ! Host node name
;  348          0344  1     NMA$C_PCNO_$FTY = 1 ^ 14 OR 0,      ! Load file type (operating system
;  349          0345  1                                         !       or diagnostics).
;  350          0346  1     NMA$C_PCNO_$HHW = 1 ^ 13 OR 0,      ! NI Hardware address from volatile
;  351          0347  1                                         !       database, used for loop circuit
;  352          0348  1                                         !       commands
;  353          0349  1     NMA$C_PCNO_$LNA = 1 ^ 12 OR 0,      ! Loop circuit node name.
;  354          0350  1     NMA$C_PCNO_$LNH = 1 ^ 11 OR 0,      ! Loop circuit node hardware address.
;  355          0351  1     NMA$C_PCNO_$LNN = 1 ^ 10 OR 0,      ! Loop circuit assistant node name.
;  356          0352  1     NMA$C_PCNO_$LAH = 1 ^ 9 OR 0,       ! Loop circuit assistant node hardware
;  357          0353  1                                         !       address.
;  358          0354  1     NMA$C_PCNO_$DA = 1 ^ 8 OR 0;        ! Destination Address on MOP message
;  359          0355  1                                         !    initiating an autoservice function.
;  360          0356  1
;  361          0357  1
;  362        M 0358  1 MACRO   NDI_SERVICE_DATA =
;  363        M 0359  1 !
;  364        M 0360  1 !                 NFB
;  365        M 0361  1 !     Param ID  Database  Param type
;  366        M 0362  1 !     --------  --------  ----------
;  367        M 0363  1         ADD,      NDI,      WORD,     ! Target's node address
;  368        M 0364  1         SDV,      NDI,      BYTE,     ! Service device type
;  369        M 0365  1         CPU,      NDI,      BYTE,     ! Target's CPU type
;  370        M 0366  1         STY,      NDI,      BYTE,     ! Software type to start load with
;  371        M 0367  1         DAD,      NDI,      LONG,     ! Address to start dump from
;  372        M 0368  1         DCT,      NDI,      LONG,     ! Dump byte count
;  373        M 0369  1         IHO,      NDI,      WORD,     ! Host node address
;  374        M 0370  1         NNA,      NDI,      STRING,   ! Target's node name
;  375        M 0371  1         SLI,      NDI,      STRING,   ! Service circuit ID
;  376        M 0372  1         SPA,      NDI,      STRING,   ! Service password
;  377        M 0373  1         HWA,      NDI,      STRING,   ! NI hardware address
;  378        M 0374  1         SNV,      NDI,      BYTE,     ! Target's service node version
;  379        M 0375  1         LOA,      NDI,      STRING,   ! Load file ID
;  380        M 0376  1         SLO,      NDI,      STRING,   ! Secondary loader file ID
;  381        M 0377  1         TLO,      NDI,      STRING,   ! Tertiary loader file ID
;  382        M 0378  1         DFL,      NDI,      STRING,   ! Diagnostics file ID
;  383        M 0379  1         SID,      NDI,      STRING,   ! Software ID
;  384        M 0380  1         DUM,      NDI,      STRING,   ! Dump file ID
;  385        M 0381  1         SDU,      NDI,      STRING,   ! Secondary dump file ID
;  386        M 0382  1         $HNA,     ,         STRING,   ! Host node name
;  387        M 0383  1         $HHW,     ,         STRING,   ! Host NI hardware address
;  388        M 0384  1         $FTY,     ,         BYTE,     ! Load file type (Operating system or
;  389        M 0385  1                                       !       diagnostics).
;  390        M 0386  1         PHA,      ,         STRING,   ! Physical address (from NICE command or
```

```
;  391        M 0387  1                                          !       NI header).
;  392        M 0388  1          $DA,     ,          STRING,     ! Destination address of MOP message
;  393        M 0389  1                                          !       which initiated autoservice on
;  394        M 0390  1                                          !       the NI.
;  395        M 0391  1
;  396        M 0392  1          LPC,     LNI,       WORD,       ! Loop count
;  397        M 0393  1          LPL,     LNI,       WORD,       ! Loop length
;  398        M 0394  1          LPD,     LNI,       BYTE,       ! Loop data type
;  399        M 0395  1          LPH,     LNI,       BYTE,       ! Loop help type (xmit, rcv, or full)
;  400        M 0396  1          LPA,     ,          STRING,     ! Loop assist NI address
;  401        M 0397  1          LPN,     ,          WORD,       ! Loop circuit node address.
;  402        M 0398  1          $LNA,    ,          STRING,     ! Loop circuit node name.
;  403        M 0399  1          $LNH,    ,          STRING,     ! Loop circuit node hardware address.
;  404        M 0400  1          LAN,     ,          WORD,       ! Loop circuit assistant node address.
;  405        M 0401  1          $LNN,    ,          STRING,     ! Loop circuit assistant node name.
;  406        M 0402  1          $LAH,    ,          STRING      ! Loop circuit assistant node hardware
;  407        M 0403  1                                          !                             address.
;  408        M 0404  1
;  409          0405  1          %,
;  410          0406  1
;  411        M 0407  1      PLI_SERVICE_DATA =
;  412        M 0408  1  !
;  413        M 0409  1  !                  NFB
;  414        M 0410  1  !     Param ID  Database  Param type
;  415        M 0411  1  !     --------  --------  ----------
;  416        M 0412  1          STI,     PLI,       WORD,       ! Line service timer
;  417        M 0413  1
;  418          0414  1          %;
;  419          0415  1
;  420          0416  1
;  421          0417  1  !
;  422          0418  1  ! Generate the Service Data Table entry count and the indices for each
;  423          0419  1  ! parameter.
;  424          0420  1  !
;  425          0421  1  SVD_INDEX_GEN (PCNO, NDI_SERVICE_DATA);
;  426          0422  1  SVD_INDEX_GEN (PCLI, PLI_SERVICE_DATA);
;  427          0423  1
;  428          0424  1  GLOBAL LITERAL
;  429          0425  1      SVD$C_ENTRY_COUNT = SVD_INDEX;
;  430          0426  1
;  431          0427  1  %ASSIGN (SVD_INDEX, 0)
;  432          0428  1
;  433          0429  1  !
;  434          0430  1  ! Generate the Service Data Table.
;  435          0431  1  !
;  436          0432  1  GLOBAL
;  437          0433  1      MOM$AB_SERVICE_DATA: BBLOCKVECTOR [SVD$C_ENTRY_COUNT, SVD$C_ENTRY_LEN]
;  438          0434  1                           PRESET (SERVICE_TAB (PCNO, NDI_SERVICE_DATA),
;  439          0435  1                                   SERVICE_TAB (PCLI, PLI_SERVICE_DATA));
;  440          0436  1
```

```
;  442          0437  1 %SBTTL 'MOP Device Table'
;  443          0438  1 !
;  444          0439  1 ! MOP device table symbol and macro definitions.
;  445          0440  1 !
;  446          0441  1 MACRO
;  447        M 0442  1     $MOPDEV (SYM, NAM) =
;  448        M 0443  1         SWITCHES UNAMES;
;  449        M 0444  1         PSECT OWN = MOM$MOPDEVNAMES;
;  450        M 0445  1         OWN
;  451        M 0446  1             STR : VECTOR [%CHARCOUNT (%ASCIC NAM), BYTE]
;  452        M 0447  1                   INITIAL (BYTE (%ASCIC NAM))
;  453        M 0448  1                   ALIGN (0);
;  454        M 0449  1         PSECT OWN = MOM$MOPDEVTABLE;
;  455        M 0450  1         OWN
;  456        M 0451  1             IND : VECTOR [MDT$K_ENTRYLEN, BYTE]
;  457        M 0452  1                   INITIAL (BYTE (SYM), LONG (STR))
;  458        M 0453  1                   ALIGN (0);
;  459        M 0454  1         UNDECLARE STR, IND;
;  460        M 0455  1         SWITCHES NOUNAMES;
;  461        M 0456  1         %ASSIGN (MOPDEVCNT, MOPDEVCNT + 1);
;  462        M 0457  1         PSECT OWN = $OWN$;
;  463          0458  1     %;
;  464          0459  1 !
;  465          0460  1 ! Initialize MOP device table and psects.
;  466          0461  1 !
;  467          0462  1 PSECT
;  468          0463  1     GLOBAL = MOM$MOPDEVTABLE (NOWRITE, ALIGN (0));
;  469          0464  1
;  470          0465  1 GLOBAL
;  471          0466  1     MOM$AB_MOPDEVICES : BBLOCKVECTOR [0, MDT$K_ENTRYLEN];
;  472          0467  1
;  473          0468  1 PSECT
;  474          0469  1     GLOBAL = MOM$MOPDEVNAMES (NOWRITE, ALIGN (0));
;  475          0470  1
;  476          0471  1 GLOBAL
;  477          0472  1     MOM$AB_MOPDEVNAMES : VECTOR [0, BYTE];
;  478          0473  1
;  479          0474  1 PSECT
;  480          0475  1     GLOBAL = $GLOBAL$;
;  481          0476  1
;  482          0477  1 COMPILETIME
;  483          0478  1     MOPDEVCNT = 0;
;  484          0479  1 !
;  485          0480  1 ! This table contains the ASCII device name strings associated with a
;  486          0481  1 ! given MOP device code.
;  487          0482  1 !
;  488          0483  1 $MOPDEV (NMA$C_SOFD_DMC, 'DMC');
;  489          0484  1 $MOPDEV (NMA$C_SOFD_UNA, 'UNA');
;  490          0485  1 $MOPDEV (NMA$C_SOFD_UNA, 'QNA');
;  491          0486  1 $MOPDEV (NMA$C_SOFD_DUP, 'DUP');
;  492          0487  1 $MOPDEV (NMA$C_SOFD_DU,  'DU');
;  493          0488  1 $MOPDEV (NMA$C_SOFD_DP,  'DP');
;  494          0489  1 $MOPDEV (NMA$C_SOFD_DQ,  'DQ');
;  495          0490  1 $MOPDEV (NMA$C_SOFD_DL,  'DL');
;  496          0491  1 $MOPDEV (NMA$C_SOFD_DA,  'DA');
;  497          0492  1 $MOPDEV (NMA$C_SOFD_DTE, 'DTE');
;  498          0493  1 $MOPDEV (NMA$C_SOFD_KL8, 'KL');
```

```
;  499          0494  1 $MOPDEV (NMA$C_SOFD_DMP, 'DMP');
;  500          0495  1 $MOPDEV (NMA$C_SOFD_DMV, 'DMV');
;  501          0496  1 $MOPDEV (NMA$C_SOFD_DPV, 'DPV');
;  502          0497  1 $MOPDEV (NMA$C_SOFD_DMF, 'DMF');
;  503          0498  1
;  504          0499  1
;  505          0500  1 GLOBAL LITERAL
;  506          0501  1     MDT$GK_MOPDEVCNT = MOPDEVCNT;
;  507          0502  1
;  508          0503  1 !
;  509          0504  1 ! Clean up.
;  510          0505  1 !
;  511          0506  1 UNDECLARE
;  512          0507  1     %QUOTE $MOPDEV;
;  513          0508  1
;  514          0509  1 END                                     ! End of module
;  515          0510  1
;  516          0511  0 ELUDOM


                                                .TITLE  MOMDAT
                                                .IDENT  \V04-000\

                                                .PSECT  MOM$MOPDEVNAMES,NOWRT,NOEXE,0

                                        00000 MOM$AB_MOPDEVNAMES::
                                                .BLKB   0
                                43  4D  44  03 00000 ;STR
                                                U.1:    .ASCII  <3>\DMC\                ;
                                41  4E  55  03 00004 ;STR
                                                U.3:    .ASCII  <3>\UNA\                ;
                                41  4E  51  03 00008 ;STR
                                                U.5:    .ASCII  <3>\QNA\                ;
                                50  55  44  03 0000C ;STR
                                                U.7:    .ASCII  <3>\DUP\                ;
                                    55  44  02 00010 ;STR
                                                U.9:    .ASCII  <2>\DU\                 ;
                                    50  44  02 00013 ;STR
                                                U.11:   .ASCII  <2>\DP\                 ;
                                    51  44  02 00016 ;STR
                                                U.13:   .ASCII  <2>\DQ\                 ;
                                    4C  44  02 00019 ;STR
                                                U.15:   .ASCII  <2>\DL\                 ;
                                    41  44  02 0001C ;STR
                                                U.17:   .ASCII  <2>\DA\                 ;
                                45  54  44  03 0001F ;STR
                                                U.19:   .ASCII  <3>\DTE\                ;
                                    4C  4B  02 00023 ;STR
                                                U.21:   .ASCII  <2>\KL\                 ;
                                50  4D  44  03 00026 ;STR
                                                U.23:   .ASCII  <3>\DMP\                ;
                                56  4D  44  03 0002A ;STR
                                                U.25:   .ASCII  <3>\DMV\                ;
                                56  50  44  03 0002E ;STR
                                                U.27:   .ASCII  <3>\DPV\                ;
                                46  4D  44  03 00032 ;STR
                                                U.29:   .ASCII  <3>\DMF\                ;
```

```
                                        .PSECT  MOM$MOPDEVTABLE,NOWRT,NOEXE,0

                        00000 MOM$AB_MOPDEVICES::
                                        .BLKB   0
              0C        00000 ;IND
                              U.2:      .BYTE   12
00000000'     00001                     .ADDRESS U.1
              01        00005 ;IND
                              U.4:      .BYTE   1
00000000'     00006                     .ADDRESS U.3
              01        0000A ;IND
                              U.6:      .BYTE   1
00000000'     0000B                     .ADDRESS U.5
              0A        0000F ;IND
                              U.8:      .BYTE   10
00000000'     00010                     .ADDRESS U.7
              02        00014 ;IND
                              U.10:     .BYTE   2
00000000'     00015                     .ADDRESS U.9
              00        00019 ;IND
                              U.12:     .BYTE   0
00000000'     0001A                     .ADDRESS U.11
              06        0001E ;IND
                              U.14:     .BYTE   6
00000000'     0001F                     .ADDRESS U.13
              04        00023 ;IND
                              U.16:     .BYTE   4
00000000'     00024                     .ADDRESS U.15
              08        00028 ;IND
                              U.18:     .BYTE   8
00000000'     00029                     .ADDRESS U.17
              14        0002D ;IND
                              U.20:     .BYTE   20
00000000'     0002E                     .ADDRESS U.19
              20        00032 ;IND
                              U.22:     .BYTE   32
00000000'     00033                     .ADDRESS U.21
              12        00037 ;IND
                              U.24:     .BYTE   18
00000000'     00038                     .ADDRESS U.23
              22        0003C ;IND
                              U.26:     .BYTE   34
00000000'     0003D                     .ADDRESS U.25
              24        00041 ;IND
                              U.28:     .BYTE   36
00000000'     00042                     .ADDRESS U.27
              26        00046 ;IND
                              U.30:     .BYTE   38
00000000'     00047                     .ADDRESS U.29

                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

3A 54 45 4E 5F  00000 P.AAB:    .ASCII  \_NET:\
                00005           .BLKB   3
          00000005  00008 P.AAA:    .LONG   5
00000000'     0000C             .ADDRESS P.AAB
```

```
                3A 44 4E 5F  00010 P.AAD:   .ASCII  \_ND:\
                   00000004  00014 P.AAC:   .LONG   4
                   00000000' 00018          .ADDRESS P.AAD
                3A 57 4E 5F  0001C P.AAF:   .ASCII  \_NW:\
                   00000004  00020 P.AAE:   .LONG   4
                   00000000' 00024          .ADDRESS P.AAF
                   000000C5  00028 P.AAG:   .LONG   197
                   00000000' 0002C          .ADDRESS MOM$AB_NICE_RCV_BUF
                   000000C5  00030 P.AAH:   .LONG   197
                   00000000' 00034          .ADDRESS MOM$AB_NICE_XMIT_BUF
                   00000200  00038 P.AAI:   .LONG   512
                   00000000' 0003C          .ADDRESS MOM$AB_ACPQIO_BUFFER
                   000005DC  00040 P.AAJ:   .LONG   1500
                   00000000' 00044          .ADDRESS MOM$AB_MOP_XMIT_BUF
                   000005DC  00048 P.AAK:   .LONG   1500
                   00000000' 0004C          .ADDRESS MOM$AB_MOP_RCV_BUF

                                            .PSECT  $GLOBAL$,NOEXE,2

                             00000 MOM$GQ_PROPRVMSK::
                                            .BLKB   8
                             00008 MOM$GW_ACP_CHAN::
                                            .BLKB   4
                          00000000 0000C MOM$GL_LOGMASK::
                                            .LONG   0
                             00010 MOM$GL_SVD_INDEX::
                                            .BLKB   4
                             00014 MOM$GB_FUNCTION::
                                            .BLKB   1
                             00015 MOM$GB_OPTION_BYTE::
                                            .BLKB   1
                             00016          .BLKB   2
                             00018 MOM$AB_NPARSE_BLK::
                                            .BLKB   36
                             0003C MOM$GB_ENTITY_CODE::
                                            .BLKB   1
                             0003D          .BLKB   3
                             00040 MOM$AB_ENTITY_BUF::
                                            .BLKB   32
                          00000000 00060 MOM$GQ_ENTITY_BUF_DSC::
                                            .LONG   0
                          00000000' 00064          .ADDRESS MOM$AB_ENTITY_BUF
                             00068 MOM$GL_SERVICE_FLAGS::
                                            .BLKB   4
                             0006C MOM$GB_EVT_POPR::
                                            .BLKB   1
                             0006D MOM$GB_EVT_PRSN::
                                            .BLKB   1
                             0006E MOM$GB_EVT_PSER::
                                            .BLKB   1
                             0006F          .BLKB   1
                             00070 MOM$GW_EVT_CODE::
                                            .BLKB   2
                             00072          .BLKB   2
                             00074 MOM$AB_NML_MAILBOX_BUFFER::
                                            .BLKB   200
                             0013C MOM$GL_NICE_RCV_MSG_LEN::
```

```
                                00140 MOM$AB_NICE_XMIT_BUF::
                                      .BLKB   197
                                00205 .BLKB   3
                                00208 MOM$AB_ACPQIO_BUFFER::
                                      .BLKB   512
                FFFFFFFF  00000000  00408 MOM$GQ_TIMEOUT::
                                      .LONG   0, -1                                                ;
                                00410 MOM$AB_CIB::
                                      .BLKB   76
                                0045C MOM$AB_LOOP_CIB::
                                      .BLKB   76
                                004A8 MOM$AB_MOP_XMIT_BUF::
                                      .BLKB   1500
                                00A84 MOM$AB_MOP_RCV_BUF::
                                      .BLKB   1500
                                01060 MOM$AB_MOP_MSG::
                                      .BLKB   1500
                                0163C MOM$GQ_MOP_MSG_DSC::
                                      .BLKB   8
                                01644 MOM$AB_MSGBLOCK::
                                      .BLKB   28
                          02010012  01660 MOM$AB_SERVICE_DATA::
                                      .LONG   33619986                                             ;
                              01F6  01664         .WORD   502                                      ;
                                01  01666         .BYTE   1                                        ;
                                00# 01667         .BYTE   0[130]                                   ;
                          02010019  016E9         .LONG   33619993                                 ;
                              0070  016ED         .WORD   112                                      ;
                                00  016EF         .BYTE   0                                        ;
                                00# 016F0         .BYTE   0[130]                                   ;
                          0201001A  01772         .LONG   33619994                                 ;
                              0071  01776         .WORD   113                                      ;
                                00  01778         .BYTE   0                                        ;
                                00# 01779         .BYTE   0[130]                                   ;
                          0201001B  017FB         .LONG   33619995                                 ;
                              007D  017FF         .WORD   125                                      ;
                                00  01801         .BYTE   0                                        ;
                                00# 01802         .BYTE   0[130]                                   ;
                          0201001C  01884         .LONG   33619996                                 ;
                              0087  01888         .WORD   135                                      ;
                                02  0188A         .BYTE   2                                        ;
                                00# 0188B         .BYTE   0[130]                                   ;
                          0201001D  0190D         .LONG   33619997                                 ;
                              0088  01911         .WORD   136                                      ;
                                02  01913         .BYTE   2                                        ;
                                00# 01914         .BYTE   0[130]                                   ;
                          0201001F  01996         .LONG   33619999                                 ;
                              008D  0199A         .WORD   141                                      ;
                                01  0199C         .BYTE   1                                        ;
                                00# 0199D         .BYTE   0[130]                                   ;
                          02020043  01A1F         .LONG   33685571                                 ;
                              01F4  01A23         .WORD   500                                      ;
                                03  01A25         .BYTE   3                                        ;
                                00# 01A26         .BYTE   0[130]                                   ;
                          02020044  01AA8         .LONG   33685572                                 ;
                              006E  01AAC         .WORD   110                                      ;
```

```
                03  01AAE          .BYTE   3
                00# 01AAF          .BYTE   0[130]
02020045        01B31              .LONG   33685573
            006F    01B35          .WORD   111
                03  01B37          .BYTE   3
                00# 01B38          .BYTE   0[130]
02020057        01BBA              .LONG   33685591
            0072    01BBE          .WORD   114
                03  01BC0          .BYTE   3
                00# 01BC1          .BYTE   0[130]
02010023        01C43              .LONG   33620003
            0073    01C47          .WORD   115
                00  01C49          .BYTE   0
                00# 01C4A          .BYTE   0[130]
02020046        01CCC              .LONG   33685574
            0078    01CD0          .WORD   120
                03  01CD2          .BYTE   3
                00# 01CD3          .BYTE   0[130]
02020047        01D55              .LONG   33685575
            0079    01D59          .WORD   121
                03  01D5B          .BYTE   3
                00# 01D5C          .BYTE   0[130]
02020048        01DDE              .LONG   33685576
            007A    01DE2          .WORD   122
                03  01DE4          .BYTE   3
                00# 01DE5          .BYTE   0[130]
02020056        01E67              .LONG   33685590
            007B    01E6B          .WORD   123
                03  01E6D          .BYTE   3
                00# 01E6E          .BYTE   0[130]
02020049        01EF0              .LONG   33685577
            007E    01EF4          .WORD   126
                03  01EF6          .BYTE   3
                00# 01EF7          .BYTE   0[130]
0202004A        01F79              .LONG   33685578
            0082    01F7D          .WORD   130
                03  01F7F          .BYTE   3
                00# 01F80          .BYTE   0[130]
0202004B        02002              .LONG   33685579
            0083    02006          .WORD   131
                03  02008          .BYTE   3
                00# 02009          .BYTE   0[130]
00000000        0208B              .LONG   0
            8000    0208F          .WORD   -32768
                03  02091          .BYTE   3
                00# 02092          .BYTE   0[130]
00000000        02114              .LONG   0
            2000    02118          .WORD   8192
                03  0211A          .BYTE   3
                00# 0211B          .BYTE   0[130]
000000C         C   219D           .LONG   0
            4000    021A1          .WORD   16384
                00  021A3          .BYTE   0
                00# 021A4          .BYTE   0[130]
00000000        0?226              .LONG   0
            000A    0222A          .WORD   10
                03  0222C          .BYTE   3
```

```
      00#  0222D          .BYTE   0[130]
00000000   022AF          .LONG   0
    0100   022B3          .WORD   256
      03   022B5          .BYTE   3
      00#  022B6          .BYTE   0[130]
01010025   02338          .LONG   16842789
    0096   0233C          .WORD   150
      01   0233E          .BYTE   1
      00#  0233F          .BYTE   0[130]
01010026   023C1          .LONG   16842790
    0097   023C5          .WORD   151
      01   023C7          .BYTE   1
      00#  023C8          .BYTE   0[130]
01010027   0244A          .LONG   1684279.
    0098   0244E          .WORD   152
      00   02450          .BYTE   0
      00#  02451          .BYTE   0[130]
0101002B   024D3          .LONG   16842795
    009A   024D7          .WORD   154
      00   024D9          .BYTE   0
      00#  024DA          .BYTE   0[130]
00000000   0255C          .LONG   0
    0099   02560          .WORD   153
      03   02562          .BYTE   3
      00#  02563          .BYTE   0[130]
00000000   025E5          .LONG   0
    009B   025E9          .WORD   155
      01   025EB          .BYTE   1
      00#  025EC          .BYTE   0[130]
00000000   0266E          .LONG   0
    1000   02672          .WORD   4096
      03   02674          .BYTE   3
      00#  02675          .BYTE   0[130]
00000000   026F7          .LONG   0
    0800   026FB          .WORD   2048
      03   026FD          .BYTE   3
      00#  026FE          .BYTE   0[130]
00000000   02780          .LONG   0
    009C   02784          .WORD   156
      01   02786          .BYTE   1
      00#  02787          .BYTE   0[130]
00000000   02809          .LONG   0
    0400   0280D          .WORD   1024
      03   0280F          .BYTE   3
      00#  02810          .BYTE   0[130]
00000000   02892          .LONG   0
    0200   02896          .WORD   512
      03   02898          .BYTE   3
      00#  02899          .BYTE   0[130]
05010015   0291B          .LONG   83951637
    0460   0291F          .WORD   1120
      01   02921          .BYTE   1
           02922          .BLKB   130

                MOM$GQ_NETNAMDSC==  P.AAA
                MOM$GQ_DLE_NAMDSC== P.AAC
                MOM$GQ_PSINAMDSC==  P.AAE
```

```
MOM$K_NML_MBX_BUF_LEN==
                        200
MOM$AB_NCP_VERSION==MOM$AB_NML_MAILBOX_BUFFER
MOM$AB_NICE_RCV_BUF==
                        MOM$AB_NML_MAILBOX_BUFFER+3
MOM$GQ_NICE_RCV_BUF_DSC==
                        P.AAG
MOM$GQ_NICE_XMIT_BUF_DSC==
                        P.AAH
MOM$GQ_ACPQIO_BUF_DSC==
                        P.AAI
MOM$AB_TRIGGER_CIB==MOM$AB_LOOP_CIB
MOM$GQ_MOP_XMIT_BUF_DSC==
                        P.AAJ
MOM$GQ_MOP_RCV_BUF_DSC==
                        P.AAK
NMA$C_PCNO_$HNA==    32768
NMA$C_PCNO_$FTY==    16384
NMA$C_PCNO_$HHW==    8192
NMA$C_PCNO_$LNA==    4096
NMA$C_PCNO_$LNH==    2048
NMA$C_PCNO_$LNN==    1024
NMA$C_PCNO_$LAH==    512
NMA$C_PCNO_$DA==     256
SVD$GK_PCNO_ADD==    0
SVD$GK_PCNO_SDV==    1
SVD$GK_PCNO_CPU==    2
SVD$GK_PCNO_STY==    3
SVD$GK_PCNO_DAD==    4
SVD$GK_PCNO_DCT==    5
SVD$GK_PCNO_IHO==    6
SVD$GK_PCNO_NNA==    7
SVD$GK_PCNO_SLI==    8
SVD$GK_PCNO_SPA==    9
SVD$GK_PCNO_HWA==    10
SVD$GK_PCNO_SNV==    11
SVD$GK_PCNO_LOA==    12
SVD$GK_PCNO_SLO==    13
SVD$GK_PCNO_TLO==    14
SVD$GK_PCNO_DFL==    15
SVD$GK_PCNO_SID==    16
SVD$GK_PCNO_DUM==    17
SVD$GK_PCNO_SDU==    18
SVD$GK_PCNO_$HNA==   19
SVD$GK_PCNO_$HHW==   20
SVD$GK_PCNO_$FTY==   21
SVD$GK_PCNO_PHA==    22
SVD$GK_PCNO_$DA==    23
SVD$GK_PCNO_LPC==    24
SVD$GK_PCNO_LPL==    25
SVD$GK_PCNO_LPD==    26
SVD$GK_PCNO_LPH==    27
SVD$GK_PCNO_LPA==    28
SVD$GK_PCNO_LPN==    29
SVD$GK_PCNO_$LNA==   30
SVD$GK_PCNO_$LNH==   31
SVD$GK_PCNO_LAN==    32
```

```
                SVD$GK_PCNO_$LNN==  33
                SVD$GK_PCNO_$LAH==  34
                SVD$GK_PCLI_STI==   35
                SVD$C_ENTRY_COUNT== 36
                MDT$GR_MOPDEVCNT==  15
```

PSECT SUMMARY

Name	Bytes	Attributes
$GLOBAL$	10660	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
$PLIT$	80	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
MOM$MOPDEVTABLE	75	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(0)
MOM$MOPDEVNAMES	54	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(0)
. ABS .	0	NOVEC,NOWRT,NORD ,NOEXE,NOSHR, LCL, ABS, CON,NOPIC,ALIGN(0)

Library Statistics

File	Symbols Total	Symbols Loaded	Symbols Percent	Pages Mapped	Processing Time
_255DUA28:[MOM.OBJ]MOMLIB.L32;1	194	19	9	21	00:00.1
_255DUA28:[SHRLIB]NMALIBRY.L32;1	887	42	4	47	00:00.2
_255DUA28:[SHRLIB]NET.L32;1	1279	24	1	63	00:00.3
_255DUA28:[SYSLIB]STARLET.L32;1	9776	0	0	581	00:03.1

COMMAND QUALIFIERS

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

 517          0512  0
Size:         0 code + 10869 data bytes
Run Time:       00:18.7
Elapsed Time:   00:39.9
Lines/CPU Min:   1642
Lexemes/CPU-Min: 40193
Memory Used:  120 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