J 6

```
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMMMMM   MMMMMM  LLL
NNN         NNN  MMMMMM   MMMMMM  LLL
NNN         NNN  MMMMMM   MMMMMM  LLL
NNNNNN      NNN  MMM   MMM   MMM  LLL
NNNNNN      NNN  MMM   MMM   MMM  LLL
NNNNNN      NNN  MMM   MMM   MMM  LLL
NNN   NNN   NNN  MMM         MMM  LLL
NNN   NNN   NNN  MMM         MMM  LLL
NNN   NNN   NNN  MMM         MMM  LLL
NNN      NNNNNN  MMM         MMM  LLL
NNN      NNNNNN  MMM         MMM  LLL
NNN      NNNNNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLL
NNN         NNN  MMM         MMM  LLLLLLLLLLLLLLL
NNN         NNN  MMM         MMM  LLLLLLLLLLLLLLL
NNN         NNN  MMM         MMM  LLLLLLLLLLLLLLL
```

_$

Ps
--
NP

NP

$G

$O

NP

PA

_L

FILEID**NMLFORWRD

B 13

```
NN      NN  MM      MM  LL            FFFFFFFFFF    OOOOOO    RRRRRRRR    WW      WW  RRRRRRRR    DDDDDDDD
NN      NN  MM      MM  LL            FFFFFFFFFF    OOOOOO    RRRRRRRR    WW      WW  RRRRRRRR    DDDDDDDD
NN      NN  MMMM  MMMM  LL            FF          OO      OO  RR      RR  WW      WW  RR      RR  DD      DD
NN      NN  MMMM  MMMM  LL            FF          OO      OO  RR      RR  WW      WW  RR      RR  DD      DD
NNNN    NN  MM  MM  MM  LL            FF          OO      OO  RR      RR  WW      WW  RR      RR  DD      DD
NNNN    NN  MM  MM  MM  LL            FF          OO      OO  RR      RR  WW      WW  RR      RR  DD      DD
NN  NN  NN  MM      MM  LL            FFFFFFFF    OO      OO  RRRRRRRR    WW      WW  RRRRRRRR    DD      DD
NN  NN  NN  MM      MM  LL            FFFFFFFF    OO      OO  RRRRRRRR    WW      WW  RRRRRRRR    DD      DD
NN    NNNN  MM      MM  LL            FF          OO      OO  RR  RR      WW  WW  WW  RR  RR      DD      DD
NN    NNNN  MM      MM  LL            FF          OO      OO  RR  RR      WW  WW  WW  RR  RR      DD      DD
NN      NN  MM      MM  LL            FF          OO      OO  RR    RR    WWWW  WWWW  RR    RR    DD      DD     ....
NN      NN  MM      MM  LL            FF          OO      OO  RR    RR    WWWW  WWWW  RR    RR    DD      DD     ....
NN      NN  MM      MM  LLLLLLLLLL  FF            OOOOOO    RR      RR  WW      WW  RR      RR  DDDDDDDD           ....
NN      NN  MM      MM  LLLLLLLLLL  FF            OOOOOO    RR      RR  WW      WW  RR      RR  DDDDDDDD           ....

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
```

NM
V0

```
0 %TITLE 'Network Management Listener module to forward NICE messages'
0 MODULE NML$FORWARD (
0                 ADDRESSING_MODE (NONEXTERNAL=GENERAL),
0                 ADDRESSING_MODE (EXTERNAL=GENERAL),
0                 IDENT = 'V04-000') =
0
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:  DECnet-VAX Network Management Listener
1 !
1 ! ABSTRACT:
1 !
1 !       This module forwards NICE messages from NCP to two other programs.
1 !       These programs are:
1 !
1 !       The Maintenance Operations Module (MOM):
1 !       It's function is to perform maintenance functions such as down line
1 !       load, up line dump, trigger, and loop line, circuit or node.
1 !
1 !       The NI Configurator Module:
1 !       It's function is to gather information about the various circuits
1 !       on the NI and, when requested, return this information to NCP.  NML
1 !       is a conduit for the request and the returned information.  In this
1 !       module, NML establishes a logical link to the NI Configurator Module
1 !       and forwards the NICE message from NCP to it.  It then takes whatever
1 !       responses returned by the NI Configurator Module, and sends them back
1 !       to NCP.
1 !
1 ! ENVIRONMENT:  VAX/VMS Operating System
1 !
1 ! AUTHOR:  Kathy Perko
```

```
;   58           0058  1 !
;   59           0059  1 ! CREATION DATE:  17-Jan-1983
;   60           0060  1 !
;   61           0061  1 ! MODIFIED BY:
;   62           0062  1 !       V03-006 MKP0006         Kathy Perko             11-April-1984
;   63           0063  1 !               Add NCP version to buffer passed to MOM.
;   64           0064  1 !
;   65           0065  1 !       V03-005 MKP0005         Kathy Perko             4-Mar-1984
;   66           0066  1 !               Don't create a new mailbox every time MOM is invoked.  It
;   67           0067  1 !               eats up bytlm quota.
;   68           0068  1 !
;   69           0069  1 !       V03-004 MKP0004         Kathy Perko             3-Jan-1984
;   70           0070  1 !               Convert old $TRNLOG system service to new $TRNLNM for
;   71           0071  1 !               translating logical names.
;   72           0072  1 !
;   73           0073  1 !       V03-003 MKP0003         Kathy Perko             10-May-1983
;   74           0074  1 !               Fix mailbox communication with MOM so that NML$MOM_MBX
;   75           0075  1 !               goes into the process logical name table instead of the
;   76           0076  1 !               group logical name table.  This will force multiple incarnations
;   77           0077  1 !               of NML and MOM to use separate mailboxes.
;   78           0078  1 !
;   79           0079  1 !       V03-002 MKP00C2         Kathy Perko             29-April-1983
;   80           0080  1 !               Upcase the logical name of the mailbox used to communicate
;   81           0081  1 !               with MOM.
;   82           0082  1 !
;   83           0083  1 !       V03-001 MKP0001         Kathy Perko             21-April-1983
;   84           0084  1 !               Use a command procedure when invoking MOM.  Also, if
;   85           0085  1 !               the SPAWN to start up MOM fails, assume it's being run
;   86           0086  1 !               from a batch job, and start it up again with null input
;   87           0087  1 !               and output devices.
;   88           0088  1 !
;   89           0089  1 !--
```

```
  1 %SBTTL 'Declarations'
  1
  1 !
  1 ! INCLUDE FILES:
  1 !
  1
  1 LIBRARY 'LIB$:NMLLIB';                     . Facility-wide definitions
  1 LIBRARY 'SHRLIB$:NMALIBRY';                 . NICE definitions
  1 LIBRARY 'SHRLIB$:NET';                      . NETACP QIO interface
  1 LIBRARY 'SYS$LIBRARY:STARLET';              . VMS common definitions
  1
  1 !
  1 ! TABLE OF CONTENTS:
  1 !
  1
  1 FORWARD ROUTINE
  1     nml$call_mom:        NOVALUE,
  1     nml$call_ni_config: NOVALUE,
  1     nml_open_config_link:NOVALUE,
  1     nml_config_qio,
  1     nml_chkerr:          NOVALUE;
  1
  1 !
  1 ! Externals
  1 !
  1 $nml_extdef;
  1
  1 EXTERNAL
  1     nml$gb_ncp_version;
  1
  1 EXTERNAL LITERAL
  1     nml$_opabterm;
  1
  1 EXTERNAL ROUTINE
  1     LIB$SPAWN,
  1     LIB$ASN_WTH_MBX,
  1     nml$bld_reply,
  1     nml$send,
  1     nml$debug_msg;
  1
  1 LITERAL
  1     nml$c_maxmbxmsg = 200;
  1
  1 OWN
  1     nml$w_mom_mbx_chan: WORD INITIAL (0),! Channel to Mailbox for communicating
  1                                         !             with MOM.
  1     nml$w_config_chan:  WORD,           ! Logical link channel to NICONFIG.
  1     nml$w_mbxchan:      WORD,           ! Logical link's Mailbox channel.
  1     nml$q_mbx_iosb:     $iosb,          ! IOSB for mailbox QIOs.
  1     nml$a_mbxmsg:                       ! Mailbox message buffer.
  1                 VECTOR [nml$c_maxmbxmsg, BYTE];
  1
  1
  1 MACRO
M 1     $nml_niconfig_ncb =
M 1         %STRING ('::',                  ! Local node
M 1                 '"TASK=$NICONFIG/',     ! Declared task name
;   148        M 0147  1                    %CHAR (0,0),            ! Word of zero
;   149        M 0148  1                    '''' )
;   150          0149  1            %;
;   151          0150  1
;   152          0151  1 BIND
;   153          0152  1     nml$q_ncb = UPLIT (
;   154          0153  1                          LONG (%CHARCOUNT ($nml_niconfig_ncb),
;   155          0154  1                          UPLIT PSECT ($OWN$) ($nml_niconfig_ncb))
;   156          0155  1                          );
;   157          0156  1
```

```
1 %SBTTL 'NML$CALL_MOM   Routine to invoke Maintenance Operations Module'
1 GLOBAL ROUTINE NML$CALL_MOM: NOVALUE =
1
2 BEGIN
2
2 !++
2 ! FUNCTIONAL DESCRIPTION:
2 !       The Maintenance Operations Module (MOM) is a separate program from
2 !       NML and NCP.  It's function is to perform various maintance operations
2 !       such as down line load, up line dump, trigger, and loop circuit, node,
2 !       or line.  For operator requested maintenance functions, NML is
2 !       a conduit for the NICE request and response.  in this module, NML
2 !       establishes a mailbox to which it writes the NICE messages, and
2 !       then spawns MOM.  MOM performs the function and puts a NICE response
2 !       in the mailbox.  NML then forwards this NICE response to NCP.
2 !
2 ! FORMAL PARAMETERS:
2 !       None
2 !
2 ! IMPLICIT INPUTS:
2 !       The NICE message in nml$ab_rcvbuffer.
2 !
2 ! IMPLICIT OUTPUTS:
2 !       A NICE message is sent to NCP.
2 !
2 ! SIDE EFFECTS:
2 !       The Maintenance Operations Module (MOM) is run.
2 !
2 !--
2
2 FIELD
2     itmlst_fields =
2         SET
2         itm_buf_len     = [0,0,16,0],
2         itm_item_code   = [2,0,16,0],
2         itm_buf_add     = [4,0,32,0],
2         itm_ret_len     = [8,0,32,0],
2         itm_list_end    = [12,0,32,0]
2         TES;
2
2 LOCAL
2     status,
2     mom_status,
2     msg_len,
2     getdvi_itmlst:      BBLOCK [4] FIELD (itmlst_fields),
2     crelnm_itmlst:      BBLOCK [4] FIELD (itmlst_fields);
2
2 OWN
2     mbx_name: BBLOCK [64];
2
2 !
2 !  Create mailbox with which to communicate with MOM.  Create a logical name
2 ! for the mailbox in the process logical name table.  MOM inherits NML's logical
2 ! names as a result of the SPAWN, and putting the logical name in the process
2 ! table makes sure that other incarnations of NML and MOM do not use this
2 ! mailbox.
2 !
```

```
;     2 IF .nml$w_mom_mbx_chan EQL 0 THEN
;     3     BEGIN
;   P 3     status = $CREMBX (CHAN = nml$w_mom_mbx_chan,
;   P 3                       MAXMSG = nml$k_rcvbflen,   ! Max length for a NICE message.
;     3                       PROMSK = %B'1111111100000000');! Protection = S:RWED, O:RWED, G, W
;     3     nml_chkerr (.status, 0);
;     2     END;
;     2 getdvi_itmlst [itm_buf_len] = 64;
;     2 getdvi_itmlst [itm_item_code] = dvi$_devnam;
;     2 getdvi_itmlst [itm_buf_add] = mbx_name;
;     2 getdvi_itmlst [itm_ret_len] = crelnm_itmlst [itm_buf_len];
;     2 getdvi_itmlst [itm_list_end] = 0;
;   P 2 status = $GETDVI (CHAN = .nml$w_mom_mbx_chan,
;     2                   ITMLST = getdvi_itmlst);
;     2 nml_chkerr (.status, 0);
;     2 crelnm_itmlst [itm_item_code] = lnm$_string;
;     2 crelnm_itmlst [itm_buf_add] = mbx_name;
;     2 crelnm_itmlst [itm_ret_len] = crelnm_itmlst [itm_buf_len];
;     2 crelnm_itmlst [itm_list_end] = 0;
;   P 2 status = $CRELNM (TABNAM = %ASCID 'LNM$PROCESS_TABLE',  ! Process logical name table,
;   P 2                   LOGNAM = %ASCID 'NML$MOM_MBX',
;     2                   ITMLST = crelnm_itmlst);
;     2 nml_chkerr (.status, 0);
;     2 !
;     2 ! Put the NCP network management version number at the beginning of the NICE
;     2 ! message being passed to MOM.
;     2 !
;     2 CH$MOVE (.nml$gl_rcvdatlen, nml$ab_rcvbuffer, nml$ab_rcvbuffer+3);
;     2 CH$MOVE (3, nml$gb_ncp_version, nml$ab_rcvbuffer);
;     2 msg_len = .nml$gl_rcvdatlen + 3;
;     2 !
;     2 ! Write NICE message to mailbox
;     2 !
;     2 nml_config_qio (.nml$w_mom_mbx_chan,
;     2                 io$_writevblk OR io$m_now,
;     2                 nml$ab_rcvbuffer,
;     2                 msg_len);
;     2 !
;     2 ! Spawn the Maintenance Operations Module (MOM).  MOM will translate the
;     2 ! logical name, NML$MOM_MBX, and then read the NICE message and process it.
;     2 ! When it is done, it will write a response NICE message to the mailbox.
;     2 !
;     2 status = LIB$SPAWN (%ASCID '$ @SYS$SYSTEM:MOM.COM',
;     2                     0,0,0,0,0,
;     2                     mom_status);
;     2 IF NOT .status THEN
;     2     status = LIB$SPAWN (%ASCID '$ @SYS$SYSTEM:MOM.COM',
;     2                         %ASCID 'NL:',   ! Null input device
;     2                         %ASCID 'NL:',   ! Null output device
;     2                         0,0,0,
;     2                         mom_status);
;     2 nml_chkerr (.status, mom_status);
;     2 !
;     2 ! Read mailbox to get the NICE response MOM puts there when it's finished.
;     2 !
;     2 msg_len = nml$k_sndbflen;
;     2 nml_config_qio (.nml$w_mom_mbx_chan,
;  273        0271  2                    io$_readvblk OR io$m_now,
;  274        0272  2                    nml$ab_sndbuffer,
;  275        0273  2                    msg_len);
;  276        0274  2  !
;  277        0275  2  ! Check to make sure that the message I got back isn't the one I just
;  278        0276  2  ! wrote to the mailbox.  This can happen if MOM isn't successfully
;  279        0277  2  ! started up.
;  280        0278  2  !
;  281        0279  2  IF CH$EQL (.nml$gl_rcvdatlen, nml$ab_rcvbuffer,
;  282        0280  2               .msg_len, nml$ab_sndbuffer, 0) THEN
;  283        0281  2      nml_chkerr (ss$_endoffile, 0);
;  284        0282  2  !
;  285        0283  2  ! Send msg to NCP.
;  286        0284  2  !
;  287        0285  2  nml$send (nml$ab_sndbuffer, .msg_len);
;  288        0286  1  END;                          ! of  nml$call_mom


                                                                          .TITLE  NMLSFORWARD Network Management Listener module
;                                                                                             to forward N
                                                                          .IDENT  \V04-000\

                                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                  00000015  00000 P.AAA:  .LONG    21                                        ;
                                                 00000000'  00004         .ADDRESS P.AAB                                     ;
42  41  54  5F  53  53  45  43  4F  52  50  24  4D  4E  4C  00008 P.AAD:  .ASCII   \LNMSPROCESS_TABLE\<0><0><0>              ;
                                            00  00  00  45  4C  00017                                                    ;
                                                  010E0011  0001C P.AAC:  .LONG    17694737                                  ;
                                                 00000000'  00020         .ADDRESS P.AAD                                     ;
            00  58  42  4D  5F  4D  4F  4D  24  4C  4D  4E  00024 P.AAF:  .ASCII   \NMLSMOM_MBX\<0>                          ;
                                                  010E000B  00030 P.AAE:  .LONG    17694731                                  ;
                                                 00000000'  00034         .ADDRESS P.AAF                                     ;
4D  3A  4D  45  54  53  59  53  24  53  59  53  40  20  24  00038 P.AAH:  .ASCII   \$ @SYS$SYSTEM:MOM.COM\<0><0><0>          ;
                        00  00  00  4D  4F  43  2E  4D  4F  00047                                                    ;
                                                  010E0015  00050 P.AAG:  .LONG    17694741                                  ;
                                                 00000000'  00054         .ADDRESS P.AAH                                     ;
4D  3A  4D  45  54  53  59  53  24  53  59  53  40  20  24  00058 P.AAJ:  .ASCII   \$ @SYS$SYSTEM:MOM.COM\<0><0><0>          ;
                        00  00  00  4D  4F  43  2E  4D  4F  00067                                                    ;
                                                  010E0015  00070 P.AAI:  .LONG    17694741                                  ;
                                                 00000000'  00074         .ADDRESS P.AAJ                                     ;
                                            00  3A  4C  4E  00078 P.AAL:  .ASCII   \NL:\<0>                                  ;
                                                  010E0003  0007C P.AAK:  .LONG    17694723                                  ;
                                                 00000000'  00080         .ADDRESS P.AAL                                     ;
                                            00  3A  4C  4E  00084 P.AAN:  .ASCII   \NL:\<0>                                  ;
                                                  010E0003  00088 P.AAM:  .LONG    17694723                                  ;
                                                 00000000'  0008C         .ADDRESS P.AAN                                     ;

                                                                          .PSECT  $OWN$,NOEXE,2

                                                      0000  00000 NMLSW_MOM_MBX_CHAN:
                                                                          .WORD    0                                         ;
                                                            00002 NMLSW_CONFIG_CHAN:
                                                                          .BLKB    2
                                                            00004 NMLSW_MBXCHAN:
                                                                          .BLKB    2
                                                            00006         .BLKB    2
```

```
                                                         00008 NML$Q_MBX_IOSB:
                                                                       .BLKB    8
                                                         00010 NML$A_MBXMSG:
                                                                       .BLKB    200
46 4E 4F 43 49 4E 24 3D 4B 53 41 54 22 3A 3A             000D8 P.AAB:  .ASCII   \::"TASK=$NICONFIG/\<0><0>\"\<0><0>
                     00 00 22 00 00 2F 47 49             000E7
                                          00             000EF         .ASCII   <0>
                                                         000F0 MBX_NAME:
                                                                       .BLKB    64

                                                               NML$Q_NCB=          P.AAA
                                                                       .EXTRN   NML$GB_EVTSRCTYP
                                                                       .EXTRN   NML$GQ_EVTSRCDSC
                                                                       .EXTRN   NML$GW_EVTCLASS
                                                                       .EXTRN   NML$GB_EVTMSKTYP
                                                                       .EXTRN   NML$GQ_EVTMSKDSC
                                                                       .EXTRN   NML$GW_EVTSNKADR
                                                                       .EXTRN   NML$GW_ACP_CHAN
                                                                       .EXTRN   NML$GL_LOGMASK, NML$GQ_ENTSTRDSC
                                                                       .EXTRN   NML$AB_QIOBUFFER
                                                                       .EXTRN   NML$GQ_QIOBFDSC
                                                                       .EXTRN   NML$AB_EXEBUFFER
                                                                       .EXTRN   NML$GL_EXEDATPTR
                                                                       .EXTRN   NML$GQ_EXEDATDSC
                                                                       .EXTRN   NML$GQ_EXEBFDSC
                                                                       .EXTRN   NML$AB_RCVBUFFER
                                                                       .EXTRN   NML$GQ_RCVBFDSC
                                                                       .EXTRN   NML$AB_SNDBUFFER
                                                                       .EXTRN   NML$GQ_SNDBFDSC
                                                                       .EXTRN   NML$GL_RCVDATLEN
                                                                       .EXTRN   NML$AB_CPTABLE, NML$AB_MSGBLOCK
                                                                       .EXTRN   NML$AB_ENTITY_ID
                                                                       .EXTRN   NML$AB_QUALIFIER_ID
                                                                       .EXTRN   NML$AB_ENTITYDATA
                                                                       .EXTRN   NML$AB_NML_NMV, NML$AB_PRMSEM
                                                                       .EXTRN   NML$AB_RECBUF, NML$AL_ENTINFTAB
                                                                       .EXTRN   NML$AL_PERMINFTAB
                                                                       .EXTRN   NML$AW_PRM_DES, NML$GB_CMD_VER
                                                                       .EXTRN   NML$GB_ENTITY_CODE
                                                                       .EXTRN   NML$GB_ENTITY_FORMAT
                                                                       .EXTRN   NML$GL_QUALIFIER_PST
                                                                       .EXTRN   NML$GB_QUALIFIER_FORMAT
                                                                       .EXTRN   NML$GB_FUNCTION
                                                                       .EXTRN   NML$GB_INFO, NML$GB_OPTIONS
                                                                       .EXTRN   NML$GL_PRMCODE, NML$GL_PRS_FLGS
                                                                       .EXTRN   NML$GL_NML_ENTITY
                                                                       .EXTRN   NML$GQ_NETNAMDSC
                                                                       .EXTRN   NML$GQ_RECBFDSC
                                                                       .EXTRN   NML$GW_PRMDESCNT
                                                                       .EXTRN   NML$GB_NCP_VERSION
                                                                       .EXTRN   NML$_OPABTERM, LIB$SPAWN
                                                                       .EXTRN   LIB$ASN_WTH_MBX
                                                                       .EXTRN   NML$BLD_REPLY, NML$SEND
                                                                       .EXTRN   NML$DEBUG_MSG, SYS$CREMBX
                                                                       .EXTRN   SYS$GETDVI, SYS$CRELNM

                                                                       .PSECT   $CODE$,NOWRT,2
                                          0FFC 00000      .ENTRY   NML$CALL_MOM, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0158
                                                                   R10,R11                                     :
                           5B 00000000G 00  9E 00002      MOVAB    NML$AB_RCVBUFFER, R11                       :
                           5A 00000000V 00  9E 00009      MOVAB    NML_CHKERR, R10                             :
                           59 00000000' 00  9E 00010      MOVAB    P.AAE, R9                                   :
                           58 00000000' 00  9E 00017      MOVAB    NML$W_MOM_MBX_CHAN, R8                      :
                           5E           10  C2 0001E      SUBL2    #16, SP                                     :
                                        68  B5 00021      TSTW     NML$W_MOM_MBX_CHAN                          ; 0214
                                        23  12 00023      BNEQ     1$                                          :
                                        7E  7C 00025      CLRQ     -(SP)                                       ; 0218
                           7E      FF00 8F  3C 00027      MOVZWL   #65280, -(SP)                               :
                                        7E  D4 0002C      CLRL     -(SP)                                       :
                           7E      0200 8F  3C 0002E      MOVZWL   #512, -(SP)                                 :
                                        58  DD 00033      PUSHL    R8                                          :
                                        7E  D4 00035      CLRL     -(SP)                                       :
                00000000G  00           07  FB 00037      CALLS    #7, SYS$CREMBX                              :
                           57           50  D0 0003E      MOVL     R0, STATUS                                  :
                                        7E  D4 00041      CLRL     -(SP)                                       ; 0219
                                        57  DD 00043      PUSHL    STATUS                                      :
                           6A           02  FB 00045      CALLS    #2, NML_CHKERR                              :
                           6E 00200040  8F  D0 00048 1$:  MOVL     #2097216, GETDVI_ITMLST                     ; 0221
                      04   AE      00F0 C8  9E 0004F      MOVAB    MBX_NAME, GETDVI_ITMLST+4                   ; 0223
                      08   AE        04 AE  9E 00055      MOVAB    CRE[NM_ITMLST, GETDVI_ITMLST+8              ; 0224
                                     0C AE  D4 0005A      CLRL     GETDVI_ITMLST+12                            ; 0225
                                        7E  7C 0005D      CLRQ     -(SP)                                       ; 0227
                                        7E  7C 0005F      CLRQ     -(SP)                                       :
                                     10 AE  9F 00061      PUSHAB   GETDVI_ITMLST                               :
                                        7E  D4 00064      CLRL     -(SP)                                       :
                           7E           68  3C 00066      MOVZWL   NML$W_MOM_MBX_CHAN, -(SP)                   :
                                        7E  D4 00069      CLRL     -(SP)                                       :
                00000000G  00           08  FB 0006B      CALLS    #8, SYS$GETDVI                              :
                           57           50  D0 00072      MOVL     R0, STATUS                                  :
                                        7E  D4 00075      CLRL     -(SP)                                       ; 0228
                                        57  DD 00077      PUSHL    STATUS                                      :
                           6A           02  FB 00079      CALLS    #2, NML_CHKERR                              :
                      06   AE           02  B0 0007C      MOVW     #2, CRE[NM_ITMLST+2                         ; 0229
                      08   AE      00F0 C8  9E 00080      MOVAB    MBX_NAME, CRELNM_ITMLST+4                   ; 0230
                      0C   AE        04 AE  9E 00086      MOVAB    CRE[NM_ITMLST, CRELNM_ITMLST+8              ; 0231
                                        6D  D4 0008B      CLRL     CRELNM_ITMLST+12                            ; 0232
                                     04 AE  9F 0008D      PUSHAB   CRELNM_ITMLST                               ; 0235
                                        7E  D4 00090      CLRL     -(SP)                                       :
                                        59  DD 00092      PUSHL    R9                                          :
                                     EC A9  9F 00094      PUSHAB   P.AAC                                       :
                                        7E  D4 00097      CLRL     -(SP)                                       :
                00000000G  00           05  FB 00099      CALLS    #5, SYS$CRELNM                              :
                           57           50  D0 000A0      MOVL     R0, STATUS                                  :
                                        7E  D4 000A3      CLRL     -(SP)                                       ; 0236
                                        57  DD 000A5      PUSHL    STATUS                                      :
                           6A           02  FB 000A7      CALLS    #2, NML_CHKERR                              :
                           56 00000000G 00  D0 000AA      MOVL     NML$GL_RCVDATLEN, R6                        ; 0241
        03   AB            6B           56  28 000B1      MOVC3    R6, NML[SAB_RCVBUFFER, NML$AB_RCVBUFFER+3    :
6B           18            00 00000000G 00  F0 000B6      INSV     NML$GB_NCP_VERSION, #0, #24, -              ; 0242
                                                                   NML$AB_RCVBUFFER                            :
                      0C   AE        03 A6  9E 000BF      MOVAB    3(R6), MSG_LEN                              ; 0243
                                     0C AE  9F 000C4      PUSHAB   MSG_LEN                                     ; 0247
                                        5B  DD 000C7      PUSHL    R11                                         :
                                         7E        70  8F  9A 000C9          MOVZBL  #112, -(SP)                                   ; 0248
                                         7E            68  3C 000CD          MOVZWL  NML$W_MOM_MBX_CHAN, -(SP)                     ; 0247
                              00000000V  00            04  FB 000D0          CALLS   #4, NML_CONFIG_QIO                            ;
                                                   08  AE  9F 000D7          PUSHAB  MOM_STATUS                                    ; 0256
                                                       7E  7C 000DA          CLRQ    -(SP)                                         ;
                                                       7E  7C 000DC          CLRQ    -(SP)                                         ;
                                                       7E  D4 000DE          CLRL    -(SP)                                         ;
                                                   20  A9  9F 000E0          PUSHAB  P.AAG                                         ;
                              00000000G  00            07  FB 000E3          CALLS   #7, LIB$SPAWN                                 ;
                                         57            50  D0 000EA          MOVL    R0, STATUS                                    ;
                                         1A            57  E8 000ED          BLBS    STATUS, 2$                                    ; 0259
                                                   08  AE  9F 000F0          PUSHAB  MOM_STATUS                                    ; 0260
                                                       7E  7C 000F3          CLRQ    -(SP)                                         ;
                                                       7E  D4 000F5          CLRL    -(SP)                                         ;
                                                   58  A9  9F 000F7          PUSHAB  P.AAM                                         ; 0261
                                                   4C  A9  9F 000FA          PUSHAB  P.AAK                                         ; 0260
                                                   40  A9  9F 000FD          PUSHAB  P.AAI                                         ;
                              00000000G  00            07  FB 00100          CALLS   #7, LIB$SPAWN                                 ;
                                         57            50  D0 00107          MOVL    R0, STATUS                                    ;
                                                   08  AE  9F 0010A 2$:      PUSHAB  MOM_STATUS                                    ; 0265
                                                       57  DD 0010D          PUSHL   STATUS                                        ;
                                         6A            02  FB 0010F          CALLS   #2, NML_CHKERR                                ;
                                   0C    AE      0200  8F  3C 00112          MOVZWL  #512, MSG_LEN                                 ; 0269
                                                   0C  AE  9F 00118          PUSHAB  MSG_LEN                                       ; 0270
                                            00000000G  00  9F 0011B          PUSHAB  NML$AB_SNDBUFFER                              ;
                                         7E        71  8F  9A 00121          MOVZBL  #113, -(SP)                                   ; 0271
                                         7E            68  3C 00125          MOVZWL  NML$W_MOM_MBX_CHAN, -(SP)                     ; 0270
                              00000000V  00            04  FB 00128          CALLS   #4, NML_CONFIG_QIO                            ;
     0C    AE           00               6B 00000000G  00  2D 0012F          CMPC5   NML$GL_RCVDATLEN, NML$AB_RCVBUFFER, #0, -     ; 0279
                                            00000000G  00     00139                  MSG_LEN, NML$AB_SNDBUFFER                     ;
                                                       0A  12 0013E          BNEQ    3$                                            ;
                                                       7E  D4 00140          CLRL    -(SP)                                         ; 0281
                                         7E      0870  8F  3C 00142          MOVZWL  #2160, -(SP)                                  ;
                                         6A            02  FB 00147          CALLS   #2, NML_CHKERR                                ;
                                                   0C  AE  DD 0014A 3$:      PUSHL   MSG_LEN                                       ; 0285
                                            00000000G  00  9F 0014D          PUSHAB  NML$AB_SNDBUFFER                              ;
                              00000000G  00            02  FB 00153          CALLS   #2, NML$SEND                                  ;
                                                           04 0015A          RET                                                   ; 0286

; Routine Size:  347 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL 'NML$CALL_NI_CONFIG   Routine to talk to NI Configurator Module'
1 GLOBAL ROUTINE NML$CALL_NI_CONFIG: NOVALUE =
1
2 BEGIN
2
2 !++
2 ! FUNCTIONAL DESCRIPTION:
2 !       This routine is called when NML receives a
2 !               SET/SHOW MODULE CONFIGURATOR command.
2 !       It establishes a logical link to the NI Configurator Module
2 !       (NICONFIG), and then drives the process of sending and receiving
2 !       NICE messages between NCP and NICONFIG.
2 !
2 ! FORMAL PARAMETERS:
2 !       NONE
2 !
2 ! IMPLICIT INPUTS:
2 !       The NICE message in nml$ab_rcvbuffer.
2 !
2 ! IMPLICIT OUTPUTS:
2 !       NICE response message(s) from NICONFIG in nml$ab_sndbuffer.
2 !
2 ! ROUTINE VALUE:
2 ! COMPLETION CODES:
2 !
2 !       NONE
2 !
2 ! SIDE EFFECTS:
2 !
2 !       NONE
2 !
2 !--
2
2 LOCAL
2     msg_len;
2
2 !
2 ! Open a logical link to configurator module.
2 !
2 nml_open_config_link ();
2 !
2 ! Send the NICE message to the NI Configurator Module via the logical
2 ! link just established.
2 !
2 nml_config_qio (.nml$w_config_chan,
2                 io$_writevblk,
2                 nml$ab_rcvbuffer,
2                 nml$gl_rcvdatlen);
2 !
2 ! Now read the response message (or messages) returned by the
2 ! NICONFIG, and forward them to NCP.
2 !
2 msg_len = nml$k_sndbflen;
2 nml_config_qio (.nml$w_config_chan,
2                 io$_readvblk,
2                 nml$ab_sndbuffer,
2                 msg_len);
```

```
0344  2 !
0345  2 ! If NICONFIG is returning multiple responses, go into a loop until all
0346  2 ! have been forwarded to NCP.  Note that the "more" and "done" messages
0347  2 ! are not forwarded because NML already sends them on it's own.
0348  2 !
0349  2 IF .nml$ab_sndbuffer <0,8> EQL nma$c_sts_mor THEN
0350  3     BEGIN
0351  3     WHILE true DO
0352  4         BEGIN
0353  4         msg_len = nml$k_sndbflen;
0354  4         nml_config_qio (.nml$w_config_chan,
0355  4                         io$_readvblk,
0356  4                         nml$ab_sndbuffer,
0357  4                         msg_len);
0358  4         !
0359  4         ! When NICONFIG returns a "done" message, exit.  A "done" message
0360  4         ! is sent to NCP later.
0361  4         !
0362  4         IF .nml$ab_sndbuffer <0,8> EQL (nma$c_sts_don AND %X'FF') THEN
0363  4             EXITLOOP
0364  4         ELSE
0365  4             !
0366  4             ! Forward NICONFIG's response to NCP.
0367  4             !
0368  4             nml$send (nml$ab_sndbuffer, .msg_len);
0369  3         END;
0370  3     END
0371  2 ELSE
0372  2     !
0373  2     ! Send msg to NCP.
0374  2     !
0375  2     nml$send (nml$ab_sndbuffer, .msg_len);
0376  1 END;                          ! of  nml$call_ni_config
```

```
                                 003C 00000          .ENTRY   NML$CALL_NI_CONFIG, Save R2,R3,R4,R5          ; 0288
              55 00000000G    00   9E 00002          MOVAB    NML$SEND, R5                                  ;
              54 00000000V    00   9E 00009          MOVAB    NML_CONFIG_QIO, R4                            ;
              53 00000000'    00   9E 00010          MOVAB    NML$W_CONFIG_CHAN, R3                         ;
              52 00000000G    00   9E 00017          MOVAB    NML$AB_SNDBUFFER, R2                          ;
              5E              04   C2 0001E          SUBL2    #4, SP                                        ;
   00000000V  00              00   FB 00021          CALLS    #0, NML_OPEN_CONFIG_LINK                      ; 0326
                 00000000G    00   9F 00028          PUSHAB   NML$GL_RCVDATLEN                              ; 0331
                 00000000G    00   9F 0002E          PUSHAB   NML$AB_RCVBUFFER                              ;
                              30   DD 00034          PUSHL    #48                                           ;
              7E              63   3C 00036          MOVZWL   NML$W_CONFIG_CHAN, -(SP)                      ;
              64              04   FB 00039          CALLS    #4, NML_CONFIG_QIO                            ;
              6E       0200   8F   3C 0003C          MOVZWL   #512, MSG_LEN                                 ; 0339
                       4004   8F   BB 00041          PUSHR    #^M<R2,SP>                                    ; 0340
                              31   DD 00045          PUSHL    #49                                           ;
              7E              63   3C 00047          MOVZWL   NML$W_CONFIG_CHAN, -(SP)                      ;
              64              04   FB 0004A          CALLS    #4, NML_CONFIG_QIO                            ;
              02              62   91 0004D          CMPB     NML$AB_SNDBUFFER, #2                          ; 0349
                              20   12 00050          BNEQ     2$                                            ;
```

```
                   6E       0200    8F   3C 00052 1$:      MOVZWL   #512, MSG_LEN                              ; 0353
                            4004    8F   BB C0057          PUSHR    #^M<R2,SP>                                 ; 0354
                                    31   DD 0005B          PUSHL    #49                                        ;
                   7E               63   3C 0005D          MOVZWL   NML$W_CONFIG_CHAN, -(SP)                   ;
                   64               04   FB 00060          CALLS    #4, NML_CONFIG_QIO                         ;
            80     8F               62   91 00063          CMPB     NML$AB_SNDBUFFER, #128                     ; 0362
                                    10   13 00067          BEQL     3$                                         ;
                                    6E   DD 00069          PUSHL    MSG_LEN                                    ; 0368
                                    52   DD 0006B          PUSHL    R2                                         ;
                   65               02   FB 0006D          CALLS    #2, NML$SEND                               ;
                                    E0   11 00070          BRB      1$                                         ; 0351
                                    6E   DD 00072 2$:      PUSHL    MSG_LEN                                    ; 0375
                                    52   DD 00074          PUSHL    R2                                         ;
                   65               02   FB 00076          CALLS    #2, NML$SEND                               ;
                                         04 00079 3$:      RET                                                 ; 0376

; Routine Size:  122 bytes,    Routine Base:  $CODE$ + 015B
```

```
  1 %SBTTL 'nml_open_config_link    Open link to NICONFIG'
  1 ROUTINE nml_open_config_link: NOVALUE =
  1
  1 !++
  1 ! FUNCTIONAL DESCRIPTION:
  1 !       This routine opens a logical link to the NI Configurator Module.
  1 !
  1 ! FORMAL PARAMETERS:
  1 !       NONE
  1 !
  1 ! IMPLICIT INPUTS:
  1 !
  1 ! IMPLICIT OUTPUTS:
  1 !        nml$w_config_chan =  Channel to Configurator Module.
  1 !        nml$w_mbxchan =  Channel to mailbox.
  1 !
  1 ! ROUTINE VALUE
  1 ! COMPLETION CODES:
  1 !       NONE
  1 !
  1 ! SIDE EFFECTS:
  1 !       NONE
  1 !
  1 !--
  1
  2 BEGIN
  2
  2 LOCAL
  2     iosb :      $iosb,  ! IO status block
  2     status;
  2
  2 !!    OWN
  2 !!      OBJNAM_DESC : BBLOCK [DSC$C_S_BLN]
  2 !!              INITIAL (%CHARCOUNT ('$NICONFIG'),
  2 !!                      UPLIT PSECT ($OWN$) (%ASCII '$NICONFIG'));
  2
  2 !
  2 ! If there is already a link to the NI Configurator Module, just return.
  2 !
  2 IF .nml$w_config_chan NEQ 0 THEN
  2     RETURN ss$_normal;
  2
  2 status = LIB$ASN_WTH_MBX ( %ASCID '_NET:',      ! Assign channel to NETACP
  2         0,0,                                     ! mailbox MAXMSG, BUFQUO (ignored)
  2         nml$w_config_chan,                       ! Channel to Configurator Module
  2         nml$w_mbxchan);                          ! Channel to mailbox
  2
  2 nml_chkerr (.status, 0);                         ! Check completion status and
  2                                                  !       signal if there's an error
P 2 status = $QIOW (
P 2                 FUNC = io$_access,               ! Request connect
P 2                 CHAN = .nml$w_config_chan,       ! Use assigned channel
P 2                 IOSB = iosb,
  2                 P2 = nml$q_ncb);                 ! Network connect block
  2
  2 nml_chkerr (.status, iosb);                      ! Check completion status and
  2                                                  !       signal if error.
```

```
;   438      P 0434  2 status = $QIOW (
;   439      P 0435  2                 FUNC = io$_readvblk,          ! Request read on mailbox
;   440      P 0436  2                 CHAN = .nml$w_mbxchan,        ! Use assigned channel
;   441      P 0437  2                 IOSB = iosb,
;   442      P 0438  2                 P1 = nml$a_mbxmsg,            ! Buffer to contain mailbox message
;   443        0439  2                 P2 = nml$c_maxmbxmsg);        ! Size maximum on mailbox message
;   444        0440  2
;   445        0441  2 nml_chkerr (.status, iosb);                   ! Check completion status and
;   446        0442  2                                               !       signal if error.
;   447        0443  2 IF .nml$a_mbxmsg [0] NEQ msg$_confirm THEN
;   448        0444  2     nml_chkerr (ss$_endoffile, 0);            ! The connect was  not accepted.
;   449        0445  2
;   450        0446  2 RETURN;
;   451        0447  1 END;                        ! of   nml_open_config_link



                                                                 .PSECT  $PLIT$,NOWRT,NOEXE,2

                  00  00  00  3A  54  45  4E  5F  00090 P.AAP:   .ASCII  \_NET:\<0><0><0>                 ;
                                       010E0005  00098 P.AAO:   .LONG   17694725                         ;
                                       00000000' 0009C          .ADDRESS P.AAP                           ;

                                                                 .EXTRN  SYS$QIOW

                                                                 .PSECT  $CODE$,NOWRT,2

                                            003C 00000 NIL_OPEN_CONFIG_LINK:
                                                                 .WORD   Save R2,R3,R4,R5                 ; 0378
                          55 00000000G  00   9E 00002          MOVAB   SYS$QIOW, R5                     ;
                          54 00000000V  00   9E 00009          MOVAB   NML_CHKERR, R4                   ;
                          53 00000000'  00   9E 00010          MOVAB   NML$W_CONFIG_CHAN, R3            ;
                          5E            08   C2 00017          SUBL2   #8, SP                           ;
                                        63   B5 0001A          TSTW    NML$W_CONFIG_CHAN                ; 0416
                                        78   12 0001C          BNEQ    1$                               ;
                                   02   A3   9F 0001E          PUSHAB  NML$W_MBXCHAN                    ; 0419
                                        53   DD 00021          PUSHL   R3                               ;
                                        7E   7C 00023          CLRQ    -(SP)                            ;
                             00000000'  00   9F 00025          PUSHAB  P.AAO                            ;
                 00000000G  00          05   FB 0002B          CALLS   #5, LIB$ASN_WTH_MBX              ;
                            52          50   D0 00032          MOVL    R0, STATUS                       ;
                                        7E   D4 00035          CLRL    -(SP)                            ; 0424
                                        52   DD 00037          PUSHL   STATUS                           ;
                            64          02   FB 00039          CALLS   #2, NML_CHKERR                   ;
                                        7E   7C 0003C          CLRQ    -(SP)                            ; 0430
                                        7E   7C 0003E          CLRQ    -(SP)                            ;
                             00000000'  00   9F 00040          PUSHAB  NML$Q_NCB                        ;
                                        7E   7C 00046          CLRQ    -(SP)                            ;
                                        7E   D4 00048          CLRL    -(SP)                            ;
                                   20   AE   9F 0004A          PUSHAB  IOSB                             ;
                                        32   DD 0004D          PUSHL   #50                              ;
                            7E          63   3C 0004F          MOVZWL  NML$W_CONFIG_CHAN, -(SP)         ;
                                        7E   D4 00052          CLRL    -(SP)                            ;
                            65          0C   FB 00054          CALLS   #12, SYS$QIOW                    ;
                            52          50   D0 00057          MOVL    R0, STATUS                       ;
                                 4004   8F   BB 0005A          PUSHR   #^M<R2,SP>                       ; 0432
                            64          02   FB 0005E          CALLS   #2, NML_CHKERR                   ;
```

```
                                       7E   7C 00061          CLRQ    -(SP)                                    : 0439
                                       7E   7C 00063          CLRQ    -(SP)                                    :
                      7E          C8   8F   9A 00065          MOVZBL  #200, -(SP)                              :
                                  0E   A3   9F 00069          PUSHAB  NML$A_MBXMSG                             :
                                       7E   7C 0006C          CLRQ    -(SP)                                    :
                                  20   AE   9F 0006E          PUSHAB  IOSB                                     :
                                       31   DD 00071          PUSHL   #49                                      :
                      7E          02   A3   3C 00073          MOVZWL  NML$W_MBXCHAN, -(SP)                     :
                                       7E   D4 00077          CLRL    -(SP)                                    :
                      65               0C   FB 00079          CALLS   #12, SYS$QIOW                            :
                      52               50   D0 0007C          MOVL    R0, STATUS                               :
                                4004   8F   BB 0007F          PUSHR   #^M<R2,SP>                               : 0441
                      64               02   FB 00083          CALLS   #2, NML_CHKERR                           :
                      31          0E   A3   91 00086          CMPB    NML$A_MBXMSG, #49                        : 0443
                                       0A   13 0008A          BEQL    1$                                       :
                                       7E   D4 0008C          CLRL    -(SP)                                    : 0444
                      7E        0870   8F   3C 0008E          MOVZWL  #2160, -(SP)                             :
                      64               02   FB 00093          CALLS   #2, NML_CHKERR                           :
                                            04 00096 1$:      RET                                              : 0447

; Routine Size:  151 bytes,    Routine Base:  $CODE$ + 01D5
```

```
  1 %SBTTL 'nml_config_qio          Issue QIO to NICONFIG'
  1 ROUTINE nml_config_qio (forward_chan, function, buffer_addr, buffer_len) =
  1
  1 !++
  1 ! FUNCTIONAL DESCRIPTION:
  1 !       Issue a read or a write on the logical link to NICONFIG.
  1 !
  1 ! FORMAL PARAMETERS:
  1 !       forward_chan - channel on which to do QIO
  1 !       function - io$_readvblk or io$_writevblk
  1 !       buffer_addr - Address of buffer from which to put or get data.
  1 !       buffer_len - byte count of data to write, or size of buffer
  1 !               to receive data.
  1 !
  1 ! OUTPUTS:
  1 !       buffer_len - length of data read (if it's a read).
  1 !
  1 ! IMPLICIT INPUTS:
  1 !
  1 !
  1 ! IMPLICIT OUTPUTS:
  1 !
  1 !
  1 ! ROUTINE VALUE:
  1 ! COMPLETION CODES:
  1 !
  1 !       NONE
  1 !
  1 ! SIDE EFFECTS:
  1 !
  1 !       NONE
  1 !
  1 !--
  1
  2 BEGIN
  2
  2 LOCAL
  2     status,
  2     iosb:       $iosb;
  2
  2 IF .function EQL io$_writevblk THEN
  2     nml$debug_msg (dbg$c_netio,
  2                     .buffer_addr,
  2                     ..buffer_len,
  2                     %ASCID 'NICE message forwarded to NICONFIG or MOM');
P 2 status = $QIOW (CHAN = .forward_chan,
P 2                 FUNC = .function,
P 2                 IOSB = iosb,
P 2                 P1 = .buffer_addr,
  2                 P2 = ..buffer_len);
  2
  2 nml_chkerr (.status, iosb);                     ! Check completion status and
  2                                                 !       signal if error.
  2 IF .function EQL io$_readvblk OR
  2    .function EQL (io$_readvblk OR io$m_now) THEN
  3     BEGIN
  3     .buffer_len = .iosb [ios$w_count];
```

```
;  510          0505  3         nml$debug_msg (dbg$c_netio,
;  511          0506  3                         .buffer_addr,
;  512          0507  3                         ..buffer_len,
;  513          0508  3                         %ASCID 'NICE message received from NICONFIG or MOM');
;  514          0509  2         END;
;  515          0510  2   RETURN nml$_sts_suc;
;  516          0511  2
;  517          0512  1   END;                          ! of nml_config_qio



                                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

6F  66  20  65  67  61  73  73  65  6D  20  45  43  49  4E  000A0 P.AAR:   .ASCII  \NICE message forwarded to NICONFIG or MO\    :
4F  43  49  4E  20  6F  74  20  64  65  64  72  61  77  72  000AF                                                                :
                    4F  4D  20  72  6F  20  47  49  46  4E  000BE                                                                :
                                            00  00  00  4D  000C8          .ASCII  \M\<0><0><0>                                  :
                                              010E0029  000CC P.AAQ:   .LONG   17694761                                              :
                                             00000000'  000D0          .ADDRESS P.AAR                                                :
65  72  20  65  67  61  73  73  65  6D  20  45  43  49  4E  000D4 P.AAT:   .ASCII  \NICE message received from NICONFIG or M\    :
43  49  4E  20  6D  6F  72  66  20  64  65  76  69  65  ͵3  000E3                                                                :
                    4D  20  72  6F  20  47  49  46  4E  4F  000F2                                                                :
                                            00  00  4D  4F  000FC          .ASCII  \OM\<0><0>                                    :
                                              010E002A  00100 P.AAS:   .LONG   17694762                                              :
                                             00000000'  00104          .ADDRESS P.AAT                                                :



                                                                            .PSECT  $CODE$,NOWRT,2

                                                      0004 00000 NML_CONFIG_QIO:
                                                                            .WORD   Save R2                                    ; 0449
                          52 00000000G  00  9E 00002                        MOVAB   NML$DEBUG_MSG, R2                            :
                          5E            08  C2 00009                        SUBL2   #8, SP                                       :
                          30       08   AC  D1 0000C                        CMPL    FUNCTION, #48                              ; 0488
                                        11  12 00010                        BNEQ    1$                                           :
                             00000000'  00  9F 00012                        PUSHAB  P.AAQ                                      ; 0491
                                   10   BC  DD 00018                        PUSHL   @BUFFER_LEN                                  :
                                   0C   AC  DD 0001B                        PUSHL   BUFFER_ADDR                                ; 0490
                                        7E  D4 0001E                        CLRL    -(SP)                                      ; 0489
                          62            04  FB 00020                        CALLS   #4, NML$DEBUG_MSG                            :
                                        7E  7C 00023 1$:                    CLRQ    -(SP)                                      ; 0497
                                        7E  7C 00025                        CLRQ    -(SP)                                        :
                                   10   BC  DD 00027                        PUSHL   @BUFFER_LEN                                  :
                                   0C   AC  DD 0002A                        PUSHL   BUFFER_ADDR                                  :
                                        7E  7C 0002D                        CLRQ    -(SP)                                        :
                                   20   AE  9F 0002F                        PUSHAB  IOSB                                         :
                          7E       04   AC  7D 00032                        MOVQ    FORWARD_CHAN, -(SP)                          :
                                        7E  D4 00036                        CLRL    -(SP)                                        :
                00000000G  00           0C  FB 00038                        CALLS   #12, SYS$QIOW                                :
                                 4001   8F  BB 0003F                        PUSHR   #^M<R0,SP>                                 ; 0499
                00000000V  00           02  FB 00043                        CALLS   #2, NML_CHKERR                               :
                           31      08   AC  D1 0004A                        CMPL    FUNCTION, #49                              ; 0501
                                        0A  13 0004E                        BEQL    2$                                           :
                00000071   8F      08   AC  D1 00050                        CMPL    FUNCTION, #113                             ; 0502
                                        16  12 00058                        BNEQ    3$                                           :
                     10    BC      02   AE  3C 0005A 2$:                    MOVZWL  IOSB+2, @BUFFER_LEN                        ; 0504
```

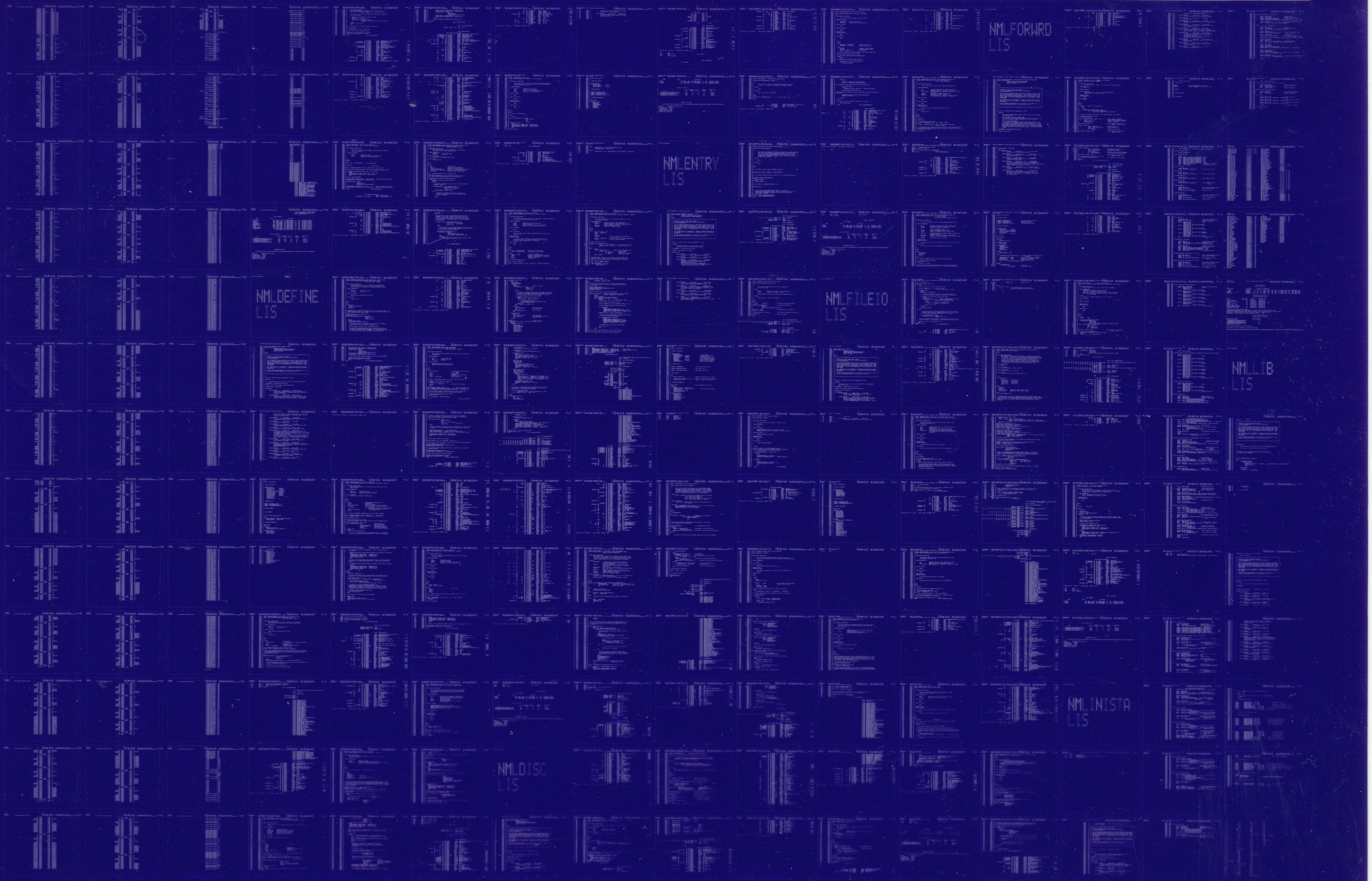
```
                                         00000000'  00  9F 0005F          PUSHAB  P.AAS                              ; 0507
                                                10  BC  DD 00065          PUSHL   @BUFFER_LEN                        ;
                                                0C  AC  DD 00068          PUSHL   BUFFER_ADDR                        ; 0506
                                                    7E  D4 0006B          CLRL    -(SP)                              ; 0505
                                        62          04  FB 0006D          CALLS   #4, NML$DEBUG_MSG                  ;
                                        50          01  D0 00070  3$:     MOVL    #1, R0                             ; 0510
                                                        04 00073          RET                                        ; 0512

; Routine Size:  116 bytes,    Routine Base:  $CODE$ + 026C
```

```
1 %SBTTL 'nml_chkerr      Check QIO completion status'
1 ROUTINE nml_chkerr (status, iosb): NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine is called to check the status returns for QIOs
1 !       on the logical link to NICONFIG.  If there is an error, a response
1 !       message is built, and sent to NCP via the handler.
1 !
1 ! FORMAL PARAMETERS:
1 !       status - the completion status of the QIO
1 !       iosb - the address of the iosb for the QIO.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     iosb:       REF $iosb;
2
2 LOCAL
2     msgsize;
2
2 IF .status AND
2    .iosb NEQ 0 THEN
2     status = .iosb [ios$w_status];
2 IF NOT .status THEN
3     BEGIN
3     !
3     ! Get rid of the logical link to NICONFIG, and clear the channel number
3     ! so the next request to NICONFIG causes NML to establish another logical
3     ! link to NICONFIG.
3     !
3     $DASSGN (CHAN = .nml$w_config_chan);
3     nml$w_config_chan = 0;
3     !
3     ! Send an error response to NCP.
3     !
3     nml$ab_msgblock [msb$b_code] = nma$c_sts_ope;
3     IF .status EQL ss$_endoffile THEN
4         BEGIN
4         nml$ab_msgblock [msb$l_flags] = msb$m_msg_fld;
4         nml$ab_msgblock [msb$l_text] = nml$_opabterm;
4         END
3     ELSE
4         BEGIN
4         nml$ab_msgblock [msb$l_flags] = msb$m_msg_fld OR msb$m_sysm_fld;
4         nml$ab_msgblock [msb$l_text] = .status;
3         END;
3     nml$bld_reply (nml$ab_msgblock, msgsize);
3     $signal_msg (nml$ab_sndbuffer, .msgsize);
2     END;
```

```
;  576       0570  2
;  577       0571  1 END;          ! of    nml_chkerr



                                                      .EXTRN  SYS$DASSGN

                                      000C 00000 NML_CHKERR:
                                                      .WORD   Save R2,R3                            ; 0514
                        53 00000000' 00 9E 00002      MOVAB   NML$W_CONFIG_CHAN, R3                 ;
                        52 00000000G 00 9E 00009      MOVAB   NML$AB_MSGBLOCK, R2                   ;
                        5E           04 C2 00010      SUBL2   #4, SP                                ;
                        0E        04 AC E9 00013      BLBC    STATUS, 2$                            ; 0541
                                  08 AC D5 00017      TSTL    IOSB                                  ; 0542
                                     05 13 0001A      BEQL    1$                                    ;
                     04 AC        08 BC 3C 0001C      MOVZWL  @IOSB, STATUS                         ; 0543
                        50        04 AC E8 00021 1$:  BLBS    STATUS, 5$                            ; 0544
                        7E           63 3C 00025 2$:  MOVZWL  NML$W_CONFIG_CHAN, -(SP)              ; 0551
              00000000G 00           01 FB 00028      CALLS   #1, SYS$DASSGN                        ;
                                     63 B4 0002F      CLRW    NML$W_CONFIG_CHAN                     ; 0552
                     04 A2           19 8E 00031      MNEGB   #25, NML$AB_MSGBLOCK+4                ; 0556
               00000870 8F        04 AC D1 00035      CMPL    STATUS, #2180                         ; 0557
                                     0D 12 0003D      BNEQ    3$                                    ;
                        62           04 D0 0003F      MOVL    #4, NML$AB_MSGBLOCK                   ; 0559
                     0C A2 00000000G 8F D0 00042      MOVL    #NML$_OPABTERM, NML$AB_MSGBLOCK+12    ; 0560
                                     09 11 0004A      BRB     4$                                    ; 0557
                        62        44 8F 9A 0004C 3$:  MOVZBL  #68, NML$AB_MSGBLOCK                  ; 0564
                     0C A2        04 AC D0 00050      MOVL    STATUS, NML$AB_MSGBLOCK+12            ; 0565
                                4004 8F BB 00055 4$:  PUSHR   #^M<R2,SP>                            ; 0567
              00000000G 00           02 FB 00059      CALLS   #2, NML$BLD_REPLY                     ;
                                     6E DD 00060      PUSHL   MSGSIZE                               ; 0568
                           00000000G 00 9F 00062      PUSHAB  NML$AB_SNDBUFFER                      ;
                            01F90000 8F DD 00068      PUSHL   #33095680                             ;
              00000000G 00           03 FB 0006E      CALLS   #3, LIB$SIGNAL                        ;
                                        04 00075 5$.  RET                                           ; 0571

; Routine Size:  118 bytes,    Routine Base:  $CODE$ + 02E0


;  578       0572  1
;  579       0573  1
;  580       0574  1 END
;  581       0575  0 ELUDOM



                                                      .EXTRN  LIB$SIGNAL

;                            PSECT SUMMARY
;
;       Name                       Bytes                     Attributes
;
;  $OWN$                              304  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                             264  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                             854  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;                               Library Statistics
;
;                                               -------- Symbols --------       Pages       Processing
;       File                                    Total     Loaded   Percent      Mapped      Time
;
;   _$255$DUA28:[NML.OBJ]NMLLIB.L32;1             341        31         9          27        00:00.1
;   _$255$DUA28:[SHRLIB]NMALIBRY.L32;1            887         3         0          47        00:00.2
;   _$255$DUA28:[SHRLIB]NET.L32;1                1279         0         0          63        00:00.3
;   _$255$DUA28:[SYSLIB]STARLET.L32;1            9776        16         0         581        00:03.2



;                                       COMMAND QUALIFIERS

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

; Size:          854 code + 568 data bytes
; Run Time:         00:18.4
; Elapsed Time:     00:58.4
; Lines/CPU Min:      1877
; Lexemes/CPU-Min: 14193
; Memory Used:  135 pages
; Compilation Complete
```

0283 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



NMLFORWRD
LIS

NMLENTRY
LIS

NMLDEFINE
LIS

NMLFILEIO
LIS

NMLLIB
LIS

NMLINISTA
LIS

NMLDISC
LIS