N 6

```
NNN         NNN     CCCCCCCCCCCC  PPPPPPPPPPPP
NNN         NNN     CCCCCCCCCCCC  PPPPPPPPPPPP
NNN         NNN     CCCCCCCCCCCC  PPPPPPPPPPPP
NNN         NNN  CCC              PPP         PPP
NNN         NNN  CCC              PPP         PPP
NNN         NNN  CCC              PPP         PPP
NNNNNN      NNN  CCC              PPP         PPP
NNNNNN      NNN  CCC              PPP         PPP
NNNNNN      NNN  CCC              PPP         PPP
NNN   NNN   NNN  CCC              PPPPPPPPPPPP
NNN   NNN   NNN  CCC              PPPPPPPPPPPP
NNN   NNN   NNN  CCC              PPPPPPPPPPPP
NNN      NNNNNN  CCC              PPP
NNN      NNNNNN  CCC              PPP
NNN      NNNNNN  CCC              PPP
NNN         NNN  CCC              PPP
NNN         NNN  CCC              PPP
NNN         NNN  CCC              PPP
NNN         NNN     CCCCCCCCCCCC  PPP
NNN         NNN     CCCCCCCCCCCC  PPP
NNN         NNN     CCCCCCCCCCCC  PPP
```

```
                                                                     M 16
**FILE**ID**NCPVRBACT
```

```
NN      NN    CCCCCCCC  PPPPPPPP    VV      VV  RRRRRRRR    BBBBBBBB      AAAAAA      CCCCCCCC  TTTTTTTTTT
NN      NN    CCCCCCCC  PPPPPPPP    VV      VV  RRRRRRRR    BBBBBBBB      AAAAAA      CCCCCCCC  TTTTTTTTTT
NN      NN  CC          PP      PP  VV      VV  RR      RR  BB      BB  AA      AA  CC              TT
NN      NN  CC          PP      PP  VV      VV  RR      RR  BB      BB  AA      AA  CC              TT
NNNN    NN  CC          PP      PP  VV      VV  RR      RR  BB      BB  AA      AA  CC              TT
NNNN    NN  CC          PP      PP  VV      VV  RR      RR  BB      BB  AA      AA  CC              TT
NN  NN  NN  CC          PPPPPPPP    VV      VV  RRRRRRRR    BBBBBBBB    AA      AA  CC              TT
NN  NN  NN  CC          PPPPPPPP    VV      VV  RRRRRRRR    BBBBBBBB    AA      AA  CC              TT
NN    NNNN  CC          PP          VV      VV  RR  RR      BB      BB  AAAAAAAAAA  CC              TT
NN    NNNN  CC          PP          VV      VV  RR  RR      BB      BB  AAAAAAAAAA  CC              TT
NN      NN  CC          PP            VV  VV    RR    RR    BB      BB  AA      AA  CC              TT          ....
NN      NN  CC          PP            VV  VV    RR    RR    BB      BB  AA      AA  CC              TT          ....
NN      NN    CCCCCCCC  PP              VV      RR      RR  BBBBBBBB    AA      AA    CCCCCCCC      TT          ....
NN      NN    CCCCCCCC  PP              VV      RR      RR  BBBBBBBB    AA      AA    CCCCCCCC      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
```

```
0001  0 %TITLE  'Action Routines for Verbs'
0002  0 MODULE NCPVRBACT(IDENT = 'V04-000',
0003  0                 ADDRESSING_MODE(EXTERNAL=GENERAL),
0004  0                 ADDRESSING_MODE(NONEXTERNAL=GENERAL)) =
0005  1 BEGIN
0006  1
0007  1 !
0008  1 !****************************************************************************
0009  1 !*                                                                          *
0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0012  1 !*  ALL RIGHTS RESERVED.                                                    *
0013  1 !*                                                                          *
0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0019  1 !*  TRANSFERRED.                                                            *
0020  1 !*                                                                          *
0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0023  1 !*  CORPORATION.                                                            *
0024  1 !*                                                                          *
0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0027  1 !*                                                                          *
0028  1 !*                                                                          *
0029  1 !****************************************************************************
0030  1 !
0031  1
0032  1 !++
0033  1 ! FACILITY:     Network Control Program (NCP)
0034  1 !
0035  1 ! ABSTRACT:
0036  1 !
0037  1 !       This module contains action routines, other routines and supporting
0038  1 !       data for performing NCP functions at the conclusion of the parsing
0039  1 !       of a command.  Routines here save the parameters into temporary
0040  1 !       storage and build the messages to be sent to NML to perform the
0041  1 !       actual functions.
0042  1 !
0043  1 ! ENVIRONMENT:  VAX/VMS Operating System
0044  1 !
0045  1 ! AUTHOR:       Darrell Duffy   , CREATION DATE: 22-October-1979
0046  1 !
0047  1 ! MODIFIED BY:
0048  1 !
0049  1 !
0050  1 !       V03-027 PRD0112         Paul R. DeStefano       02-Aug-1984
0051  1 !               Defeat PRD0099 don't default to area 1 for TELL or
0052  1 !               SET EXEC command when area isn't specified.
0053  1 !
0054  1 !       V03-026 PRD0105         Paul R. DeStefano       20-Jul-1984
0055  1 !               Correct an error in PRD0083 which caused the
0056  1 !               NCP$BLD_ENTITY routine to store an extra byte
0057  1 !               if the format type is negative.
```

```
!
!       V03-025 PRD0099         Paul R. DeStefano       01-May-1984
!               Modified ACT$SAVPRM to store a default area address
!               of 1 when ACT$GL_NO_XAREA_Q flag is set and the
!               parameter block address is PBK$G_VRB_XID.  The above
!               is true when a TELL or SET EXEC command is entered
!               and the exec address specified doesn't specify an
!               area.
!
!       V03-024 PRD0088         Paul R. DeStefano       27-Mar-1984
!               Modify ACT$SAVPRM so that the area number is saved
!               if the parameter type is PBK$K_ESNO.
!
!       V03-023 PRD0083         Paul R. DeStefano       26-Mar-1984
!               Fixed NCP$BLD_ENTITY routine to store only one byte
!               when entity = area.
!
!       V03-022 PRD0065         Paul R. DeStefano       05-Feb-1984
!               Change ACT$GL_NODADR_Q references to ACT$GL_ADR_Q.
!
!       V03-021 PRD0058         Paul R. DeStefano       05-Feb-1984
!               Fixed storing of NI address parameter in ACT$SAVPRM
!               routine when address is specified as nn-nn-nn-nn-nn-nn,
!               i.e., with the dashes.
!
!       V03-020 PRD0054         Paul R. DeStefano       05-Feb-1984
!               Add new module name X25-ACCCESS.
!
!       V03-019 PRD0049         Paul R. DeStefano       01-Feb-1984
!               Clear ACT$GL_ADR_Q in ACT$SAVPRM routine when
!               storing a Area and Node address.
!
!       V03-018 RPG0018         Bob Grosso              10-Jun-1983
!               Give more info with LOGIC error.
!
!       V03-017 RPG0017         Bob Grosso              11-Mar-1983
!               Add saving and building of NIADR type.
!
!       V03-016 RPG0016         Bob Grosso              24-Feb-1983
!               Support ESMO type to store the LOGGING module name
!               in PDB$G_VRB_EVE.
!
!       V03-015 RPG0015         Bob Grosso              09-Nov-1982
!               Save new parameter type AADR, an area and node address,
!               and build a message.
!
!       V03-014 RPG0014         Bob Grosso              24-Sep-1982
!               Add act$gl_nodarea so that the Area can be stored.
!
!       V03-013 RPG0013         Bob Grosso              15-Sep-1982
!               Add NCP$GL_QUALPRS to note if a qualifier has been
!               parsed to determine if ALL should be sent.
!               Fix HEX storage from reversing nibbles.
!               Add NCP$GL_NOPARMS to note not to check for parameters.
!
!       V03-012 RPG0012         Bob Grosso              08-Sep-1982
!               Support new data type, HEX which replaces HXPS for
```

```
;  115     0115  1 !                 use by CALL MASK and CALL VALUE.
;  116     0116  1 !
;  117     0117  1 !         V03-011 RPG0011          Bob Grosso             03-Aug-1982
;  118     0118  1 !                 Correct building of NICE from RNGL pairs.
;  119     0119  1 !
;  120     0120  1 !         V3-010  RPG0010          Bob Grosso             14-Jul-1982
;  121     0121  1 !                 Add new module entity names, X25-TRACE, X29-SERVER,
;  122     0122  1 !                 CONSOLE, CONFIGURATOR, LOADER, LOOPER.
;  123     0123  1 !
;  124     0124  1 !         V3-009  RPG0009          Bob Grosso             22-Jun-1982
;  125     0125  1 !                 Store away the entity name if the entity id is MODULE.
;  126     0126  1 !                 Store away the first parameter if entity is module,
;  127     0127  1 !                 so that ncpsholis can figure out which display table
;  128     0128  1 !                 to use.
;  129     0129  1 !                 Add ncp$_acccir.
;  130     0130  1 !
;  131     0131  1 !         V008    TMH0008          Tim Halvorsen          05-Apr-1982
;  132     0132  1 !                 Add ACT$TESTLONG routine.
;  133     0133  1 !                 Fix bug in storing of subaddress range value.
;  134     0134  1 !
;  135     0135  1 !         V007    TMH0007          Tim Halvorsen          11-Jan-1982
;  136     0136  1 !                 Map circuit requests to a V2.0 NML into the
;  137     0137  1 !                 appropriate line request.
;  138     0138  1 !
;  139     0139  1 !         V006    TMH0006          Tim Halvorsen          02-Dec-1981
;  140     0140  1 !                 Change text of message displayed in TMH0005.
;  141     0141  1 !                 Add check so that message is not displayed for
;  142     0142  1 !                 error responses.
;  143     0143  1 !
;  144     0144  1 !         V005    TMH0005          Tim Halvorsen          11-Nov-1981
;  145     0145  1 !                 Add ESCI parameter type handling.
;  146     0146  1 !                 Output "No information in database" message if all
;  147     0147  1 !                 responses from a show request are null.
;  148     0148  1 !
;  149     0149  1 !         V004    TMH0004          Tim Halvorsen          02-Sep-1981
;  150     0150  1 !                 Remove Phase II compatibility code.
;  151     0151  1 !
;  152     0152  1 !         V003    TMH0003          Tim Halvorsen          10-Jul-1981
;  153     0153  1 !                 Add action routine ACT$COPY_VALUE
;  154     0154  1 !                 Add parameter types:
;  155     0155  1 !                         SAD - Subaddress range
;  156     0156  1 !                         OBJ - Object ID
;  157     0157  1 !                 Do not copy string in MOV_STR if the byte count is negative
;  158     0158  1 !                 (meaning ACTIVE or KNOWN of that parameter).
;  159     0159  1 !
;  160     0160  1 !         V002    TMH0002          Tim Halvorsen          22-Jun-1981
;  161     0161  1 !                 Treat negative entity type codes in SDB as system-specific.
;  162     0162  1 !                 Add ENT parameter type, entity type and ID.
;  163     0163  1 !
;  164     0164  1 !         V001    TMH0001          Tim Halvorsen          13-Jun-1981
;  165     0165  1 !                 Make all external references longword relative.
;  166     0166  1 !--
```

```
1 %SBTTL  'Definitions'
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1         ACT$SAVPRM,                              ! Action routine to save a parameter
1         NCP$MOV_QSTR     : NOVALUE,              ! Move quoted string
1         NCP$MOV_STR      : NOVALUE,              ! Move unquoted string
1         ACT$CLRLONG,                             ! Action routine to clear a longword
1         ACT$TESTLONG,                            ! Action routine to test a longword
1         ACT$COPY_VALUE,                          ! Action routine to copy a longword
1         ACT$VRB_EXIT     : NOVALUE,              ! Action routine for EXIT command
1         ACT$VRB_UTILITY,                         ! Action routine for most commands
1         ACT$VRB_SHOLIS,                          ! Show/List action routine
1         NCP$HNDL_SHOLIS,                         ! Show/list handler
1         ACT$VRB_LOOP,                            ! Action routine for LOOP command
1         V2_REQUESTS,                             ! V2.0 NML request compatibility
1         NCP$BLD_PROLOG   : NOVALUE,              ! Build prolog of message
1         NCP$BLD_ENTITY   : NOVALUE,              ! Build entity code
1         NCP$MODULE_TYPE,                         ! Return code for module type
1         NCP$BLD_PRMS     : NOVALUE;              ! Build parameters into message
1
1 !
1 ! INCLUDE FILES:
1 !
1
1         LIBRARY 'SYS$LIBRARY:STARLET.L32';
1         LIBRARY 'LIB$:NCPLIBRY.L32';
1         LIBRARY 'LIB$:NMALIBRY.L32';
1
1 !
1 ! EQUATED SYMBOLS:
1 !
1
1 LITERAL
1         NCP$C_MSGSIZ = 1000      ! Message buffer size
1         ;
1
1 !
1 ! OWN STORAGE:
1 !
1
1 GLOBAL
1         NCP$GL_ENTITY:  SIGNED,                  ! Entity type code.  If negative, then
1                                                  ! system-specific entity (NMA$C_SENT_)
1         NCP$GL_MODTYP,                           ! Code for MODULE ENTITY type
1         NCP$GW_PRMTYP,                           ! Code for first parameter type,
1                                                  !  used with MODULE entities
1         NCP$GL_QUALPRS,                          ! Qualifier present on parsed line
1         NCP$GL_NOPARMS,                          ! Entity has no parameters so don't signal
1         NCP$GT_MSGBFR:  VECTOR [NCP$C_MSGSIZ, BYTE];    ! Message buffer itself
```

```
!
! EXTERNAL REFERENCES:
!

EXTERNAL LITERAL
        ACT$C_RNGLSTMAX,                ! Size of range list storage vector
        NCP$_ACCCIR,                    ! Access control with loop circuit
        NCP$_ACCLIN,                    ! Access control with loop line
        NCP$_INVEVE,                    ! Invalid event range was specified
        NCP$_INVRSP,                    ! Invalid management response
        NCP$_LOGIC,                     ! There is a bug in NCP.
        NCP$_NOPARM,                    ! Status for no parameters saved
        NCP$_REPEAT,                    ! Parameter value has been repeated
        NCP$_V2COMP;                    ! V2.0 NML parameter conversion error


EXTERNAL
        ACT$GA_RNGLST : VECTOR [(2 * MAX_RNGLST_PAIRS) + 1, WORD],
                                        ! Range list vector
        PDB$G_VRB_ALL,                  ! Parameter block for ALL code
        PDB$G_LOO_ACC,                  ! Access control for loop command
        PDB$G_LOO_PSW,
        PDB$G_LOO_USR,
        NCP$GL_FNC_CODE : BBLOCK [4],   ! Function code byte
        NCP$GL_OPTION : BBLOCK [4],     ! Option code byte
        NCP$GL_EXELCB: REF BBLOCK;      ! Pointer to executor link control blk

EXTERNAL ROUTINE
        NCP$SIG_CMDERR,                 ! Signal a command error
        NCP$OPENLINK: NOVALUE,          ! Open a link to the listener
                                        ! (and get its version #)
        NCP$SENDMSG : NOVALUE,          ! Write a message to the listner
        NCP$READRSP,                    ! Read a response from the listner
        NCP$OPENSHO : NOVALUE,          ! Open show/list output file
        NCP$SHOHEAD : NOVALUE,          ! Write the heading
        NCP$SHOLIS  : NOVALUE,          ! Write the parameters from one msg
        NCP$WRITESHO : NOVALUE,         ! Write a line to output file
        NCP$CLOSESHO : NOVALUE;         ! Close the output file
```

```
1 %SBTTL  'ACT$SAVPRM     Save a Parameter'
1 GLOBAL ROUTINE ACT$SAVPRM (OPT, STRCNT, STRPTR, TKNCNT, TKNPTR,
1                       CHR, NUM, PBLK) =       !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Action routine to store a parameter away for later use.
1 !       The parameter PBLK is the address of a control block which controls
1 !       the type of parameter saved and the address.  If the parameter
1 !       already has been saved in this parse, this routine complains by
1 !       signalling an error and returns failure.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       Parse state table
1 !
1 !       TKNCNT, TKNPTR  Descriptor of the token which may be used if the
1 !                       type specifies a token
1 !
1 !       PBLK            Address of the parameter control block
1 !                       BYTE (PBK$K_code)       code for type of parameter
1 !                       LONG (ADR)              Address of the parameter data
1 !                       LONG (PRM)              Value of the parameter for
1 !                                               the code specified
1 !
1 !               The parameter data has the following form
1 !                       BYTE (STS)              0 for not yet saved here
1 !                                               1 for data saved here
1 !                       ...                     Data of the parameter itself
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Success or an error signalled and failure
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
```

```
;  312          0308  1
;  313          0309  2        BEGIN
;  314          0310  2
;  315          0311  2        MAP
;  316          0312  2            PBLK : REF BBLOCK [PBK$C_SIZE]  ! Parameter control block
;  317          0313  2            ;
;  318          0314  2
;  319          0315  2        LOCAL
;  320          0316  2            CPTR,                           ! Character pointer
;  321          0317  2            PTR                             ! Pointer to data segment
;  322          0318  2            ;                  !
;  323          0319  2
;  324          0320  2        OWN
;  325          0321  2            EVELST                          ! Last event code seen
;  326          0322  2            ;
;  327          0323  2
;  328          0324  2        GLOBAL
;  329          0325  2            ACT$GL_ADR_Q,                   ! Flag for Node address, object number
;  330          0326  2            ACT$GL_NODAREA,                 ! Node area
;  331          0327  2            ACT$GL_SAD_BEGIN,               ! Subaddress beginning
;  332          0328  2            ACT$GL_SAD_END;                 ! Subaddress end
;  333          0329  2
;  334          0330  2        EXTERNAL
;  335          0331  2            PBK$G_VRB_XID,                  ! Param block for exec node ID
;  336          0332  2            ACT$GL_NO_XAREA_Q;              ! Flag for no exec area specified
;  337          0333  2
```

```
2
2      PTR = .PBLK [PBK$L_PDB_ADR];          ! Address of param data block
2
2      IF .BBLOCK [.PTR, PDB$B_STS_FLG]      ! Is there already one here?
2      THEN
3          BEGIN
3          IF NOT                            ! Exclude error check for some codes
4              (SELECTONEU .PBLK [PBK$B_TYPECODE]
4              OF
4                  SET
4                  [PBK$K_ESET TO PBK$K_ESEX, PBK$K_ESCI]: 1;
4                  [PBK$K_PRVL]: 1;
4                  [OTHERWISE]: 0;
4                  TES)
3          THEN
4              BEGIN
4              NCP$SIG_CMDERR(NCP$_REPEAT,   ! Yes, signal an error
4                      .TKNCNT, .TKNPTR,
4                      .STRCNT, .STRPTR);
4              RETURN FAILURE;               ! Return a syntax error
3              END;
2          END;
2
2      BBLOCK [.PTR, PDB$B_STS_FLG] = 1;     ! Set the status for we have one
2      PTR = BBLOCK [.PTR, PDB$T_DATA];      ! Advance the address to data
2
2 !
2 !        Dispatch on the type code
2 !
2
2      CASE .PBLK [PBK$B_TYPECODE]
2                  FROM PBK$K_LOW
2                  TO PBK$K_HIGH
2                  OF
2      SET
2
2      [PBK$K_LITB] :                        ! Literal byte
2          CH$WCHAR (.PBLK [PBK$L_PARAM],
2                  CH$PTR (.PTR)
2                  )
2          ;
2
2      [PBK$K_NUMB] :                        ! Number as a byte
2          CH$WCHAR (.NUM, CH$PTR (.PTR) )
2          ;
```

```
2
2          [PBK$K_NUMW] :                           ! Number as a word
2              (.PTR) <0,16,0> = .NUM <0,16,0>
2              ;
2
2          [PBK$K_NUML] :                           ! Number as a long word
2              .PTR = .NUM
2              ;
2
2          [PBK$K_LITL] :                           ! Literal longword
2              .PTR = .PBLK [PBK$L_PARAM]
2              ;
2
2          [PBK$K_TKN] :                            ! Token as a string
2  !           IF .ACT$GL_NO_XAREA_Q                ! If no exec area specified,
2  !               AND .PBLK EQL PBK$G_VRB_XID !  and TELL or SET EXEC,
2  !           THEN
2  !               BEGIN                        !  store the token moved down 2 bytes,
2  !               NCP$MOV_STR(.TKNCNT, .TKNPTR, .PTR + 2);
2  !               (.PTR) <0,8> = .TKNCNT + 2; !  store length increased by 2,
2  !               (.PTR) <8,16> = %ASCII'1.'; !  and store area "1." before the token.
2  !               ACT$GL_NO_XAREA_Q = 0;       ! Zero the flag.
2  !               END
2  !           ELSE
2                  NCP$MOV_STR (.TKNCNT, .TKNPTR, .PTR)
2              ;
2
2          [PBK$K_TKNQ] :                           ! Token as a quoted string
2              NCP$MOV_QSTR (.TKNCNT, .TKNPTR, .PTR)
2              ;
2
2          [PBK$K_STRQ] :                           ! A described string as a quoted str
2              NCP$MOV_QSTR
2                  (
2                  .VECTOR [.PBLK [PBK$L_PARAM], 0],
2                  .VECTOR [.PBLK [PBK$L_PARAM], 1],
2                  .PTR
2                  )
2              ;
2
2          [PBK$K_NADR] :                           ! Node address
3              BEGIN
3              IF .ACT$GL_ADR_Q                     ! Did we save a node address??
3              THEN
4                  BEGIN
4                  LOCAL
4                      NODE_AREA;
4
4                  ACT$GL_ADR_Q = 0;                ! Zero the flag
4                  (.PTR) <0,8,0> = 0;              ! Indicate that it is an address
4                  NODE_AREA = .NUM <0,16,0> +      ! The address
4                      (.ACT$GL_NODAREA ^ 10);      !  and the area
4                  (.PTR) <8,16,0> = .NODE_AREA;
4                  END
3              ELSE                                 ! Move a string with no quotes
3                  NCP$MOV_STR (.TKNCNT, .TKNPTR, .PTR)
3              END
```

```
;  442        0436  2            ;
;  443        0437  2
;  444        0438  2        [PBK$K_AADR] :                    ! Area and Node address to be stored in a word
;  445        0439  3            BEGIN
;  446        0440  3            LOCAL
;  447        0441  3                NODE_AREA;
;  448        0442  3
;  449        0443  3            ACT$GL_ADR_Q = 0;             ! Zero the flag
;  450        0444  3            NODE_AREA = .NUM <0,16,0> +   ! The address
;  451        0445  3                (.ACT$GL_NODAREA ^ 10);   !  and the area
;  452        0446  3            (.PTR) <0,16,0> = .NODE_AREA;
;  453        0447  2            END;
;  454        0448  2
;  455        0449  2        [PBK$K_OBJ]:                      ! Object ID
;  456        0450  2            IF .ACT$GL_ADR_Q              ! Did we save a object number?
;  457        0451  2            THEN
;  458        0452  3                BEGIN
;  459        0453  3                ACT$GL_ADR_Q = 0;         ! Zero the flag
;  460        0454  3                (.PTR) <0,8,0> = 0;       ! Indicate an object number, not name
;  461        0455  3                (.PTR) <8,8,0> = .NUM <0,8,0>; ! Object number
;  462        0456  3                END
;  463        0457  2            ELSE                          ! Move a string with no quotes
;  464        0458  2                NCP$MOV_STR (.TKNCNT, .TKNPTR, .PTR);
;  465        0459  2
;  466        0460  2        [PBK$K_ENT]:                      ! Entity type and ID
;  467        0461  3            BEGIN
;  468        0462  3            (.PTR) <0,8> = NMA$C_ENT_NOD;   ! &&& Only nodes handled for now &&&
;  469        0463  3            IF .ACT$GL_ADR_Q              ! Did we save a node address??
;  470        0464  3            THEN
;  471        0465  4                BEGIN
;  472        0466  4                ACT$GL_ADR_Q = 0;         ! Zero the flag
;  473        0467  4                (.PTR+1) <0,8> = 0;       ! Indicate a node address
;  474        0468  4                (.PTR+2) <0,16> = .NUM <0,16>; ! The node address
;  475        0469  4                END
;  476        0470  3            ELSE                          ! Move a string with no quotes
;  477        0471  3                NCP$MOV_STR (.TKNCNT, .TKNPTR, .PTR+1);
;  478        0472  2            END;
;  480        0473  2
;  481        0474  2      [PBK$K_HXPS] :                       ! Hex password
;  482        0475  3          BEGIN
;  483        0476  3          (.PTR) <0,8,0> =
;  484        0477  3              (.TKNCNT+1 )/ 2;             ! Save the byte count
;  485        0478  3                                           ! Zap last byte for high zero if odd
;  486        0479  3                                           ! Number of bytes in string
;  487        0480  3          (.PTR + 1 + ( (.TKNCNT + 1) / 2) )<0, 8, 0> = 0;
;  488        0481  3          CPTR = CH$PTR (.TKNPTR);          ! Make a pointer to the string
;  489        0482  3
;  490        0483  3
;  491        0484  3 !        Step backward from last nibble used to beginning
;  492        0485  3 !        of string
;  493        0486  3 !
;  494        0487  3
;  495        0488  4          DECRU BPOS FROM 8 + ( (.TKNCNT-1) * 4) ! Last nibble
;  496        0489  3                   TO 8                     ! First nibble
;  497        0490  3                   BY 4                     ! A nibble wide
;  498        0491  3          DO
;  499        0492  4              BEGIN
;  500        0493  4              LOCAL CHAR;
;  501        0494  4              CHAR = CH$RCHAR_A (CPTR);     ! Obtain the character
;  502        0495  5              IF (.CHAR GEQU 'A' )          ! Adjust the range of hex
;  503        0496  5                  AND (.CHAR LEQU 'F')
;  504        0497  4              THEN CHAR = .CHAR + 9;        ! Make lower nibble valid
;  505        0498  4              (.PTR) <.BPOS, 4, 0> = .CHAR ! Insert the nibble in the string
;  506        0499  4              END
;  507        0500  3          END
;  508        0501  2          ;
```

```
2
2      [PBK$K_HEX] :                           ! Hex number
3          BEGIN
3          LOCAL
3              FIRST_NIBBLE;
3
3          (.PTR) <0,8,0> =
3              (.TKNCNT+1 )/ 2;                ! Save the byte count
3          (.PTR + 1)<0, 8, 0> = 0;            ! Zap first byte for high zero if odd
3                                              ! Number of bytes in string
3          CPTR = CH$PTR (.TKNPTR);            ! Make a pointer to the string
3
3 !
3 !        Step forward from first nibble used to end
3 !        of string
3 !
3
3          FIRST_NIBBLE = FALSE;               ! Toggle which nibble of the byte
3                                              !  will be filled
3          INCRU BPOS FROM 8                   ! First nibble
4             TO 8 + ( (.TKNCNT-1) * 4)        ! Last nibble
3             BY 4                             ! A nibble wide
3          DO
4              BEGIN
4              LOCAL CHAR;
4              CHAR = CH$RCHAR_A (CPTR);       ! Obtain the character
5              IF (.CHAR GEQU 'A' )            ! Adjust the range of hex
5                  AND (.CHAR LEQU 'F')
4              THEN CHAR = .CHAR + 9;          ! Make lower nibble valid
4              IF .FIRST_NIBBLE
4              THEN
4                  (.PTR) <.BPOS -4, 4, 0> = .CHAR ! Store second char in first nibble
4              ELSE
4                  (.PTR) <.BPOS +4,4,0> = .CHAR;  ! Store the first char in the second nibble
4              FIRST_NIBBLE = NOT .FIRST_NIBBLE;   ! Toggle
4              END
3          END
2          ;
```

```
2
2 !
2 !       Event building types
2 !
2
2     [PBK$K_ESET] :                          ! Setup to collect an event
3         BEGIN
3         (.PTR) <16, 8, 0> = 8;              ! The type mask counter
3         (.PTR+11) <0, 8, 0> = -1;           ! The source entity code (none)
3         BBLOCK [.PTR-1, PDB$B_STS_FLG] = 0; ! Inhibit repeated store checking
2         END;
2
2     [PBK$K_ECLS] :                          ! The event class code
3         BEGIN
3         (.PTR) <0, 16, 0> = .NUM;
2         END;
2
2     [PBK$K_EMSK] :                          ! A simple type
3         BEGIN
3         EVELST = .NUM;                      ! Save the event type for a range
3         (.PTR+3) <.NUM, 1, 0> = 1;          ! Set the bit
2         END;
2
2     [PBK$K_ERNG] :                          ! Set a range of event type bits
3         BEGIN
3         IF .EVELST GTRU .NUM                ! Check the range for increasing
3         THEN
3             NCP$SIG_CMDERR(NCP$_INVEVE,     ! Yes, signal an error
3                   .TKNCNT, .TKNPTR,
3                   .STRCNT, .STRPTR);
3
3         INCRA IDX FROM .EVELST TO .NUM      ! Scan the bits
3         DO
3             (.PTR+3) <.IDX, 1, 0> = 1;      ! And set them
3
3         EVELST = .NUM;                      ! Set for the next range
2         END;
```

```
2
2       [PBK$K_EWLD] :                          ! Event wild card
2           0;                                  ! Sets the parameter as active
2                                               ! Code is stored by mask operation
2                                               ! we don't do anything here
2
2       [PBK$K_ESNO] :                          ! Event source node
3           BEGIN
3           EXTERNAL
3               ACT$GL_ADR_Q;
3
3           IF  CH$RCHAR (.PTR+11) NEQ 255  ! If a parameter saved here
3           THEN
3               NCP$SIG_CMDERR(NCP$_REPEAT, ! Yes, signal an error
3                       .TKNCNT, .TKNPTR,
3                       .STRCNT, .STRPTR);
3
3           IF .ACT$GL_ADR_Q                    ! Node address saved?
3           THEN
4               BEGIN
4               LOCAL
4                   NODE_AREA;
4               (.PTR+12) <0,8,0> = 0;          ! Indicate that it is an address
4               NODE_AREA = .NUM <0,16,0> + ! Save the address
4                   (.ACT$GL_NODAREA ^ 10);
4               (.PTR+12) <8,16,0> =            !  and the area
4                   .NODE_AREA;
4               ACT$GL_ADR_Q = 0;               ! Clear the address flag
4               END
3           ELSE                                ! If its a node name, save it
3               NCP$MOV_STR (.TKNCNT, .TKNPTR, .PTR+12);
3           CH$WCHAR (0, .PTR+11);              ! Indicate its a node
2           END;
2
2       [PBK$K_ESLI] :                          ! Save a source line id
3           BEGIN
3           IF  CH$RCHAR (.PTR+11) NEQ 255  ! If a parameter saved here
3           THEN
3               NCP$SIG_CMDERR(NCP$_REPEAT, ! Yes, signal an error
3                       .TKNCNT, .TKNPTR,
3                       .STRCNT, .STRPTR);
3           NCP$MOV_STR (.TKNCNT, .TKNPTR, .PTR+12);
3           CH$WCHAR (1, .PTR+11);              ! Indicate its a line id
2           END;
2
2       [PBK$K_ESCI] :                          ! Save a source circuit id
3           BEGIN
3           IF  CH$RCHAR (.PTR+11) NEQ 255  ! If a parameter saved here
3           THEN
3               NCP$SIG_CMDERR(NCP$_REPEAT, ! Yes, signal an error
3                       .TKNCNT, .TKNPTR,
3                       .STRCNT, .STRPTR);
3           NCP$MOV_STR (.TKNCNT, .TKNPTR, .PTR+12);
3           CH$WCHAR (3, .PTR+11)               ! Indicate its a circuit id
2           END;
2
2       [PBK$K_ESEX] :                          ! Save the executor as source
```

```
3          BEGIN
3          IF  CH$RCHAR (.PTR+11) NEQ 255  ! If a parameter saved here
3          THEN
3              NCP$SIG_CMDERR(NCP$_REPEAT, ! Yes, signal an error
3                      .TKNCNT, .TKNPTR,
3                      .STRCNT, .STRPTR);
3          CH$FILL (0, 4, .PTR+11);        ! Indicate node source type and
3                                          ! store zero node address
2          END;
```

```
2
2
2 !       Privilege list types
2 !
2
2     [PBK$K_PRVC] :                         ! Clear the privilege mask
3         BEGIN
3         CH$FILL (0, LEN_PRV_MSK+1, .PTR+1);
3         (.PTR) <0, 8, 0> = 8               ! The count for the image field
3         END
2         ;
2
2     [PBK$K_PRVL] :                         ! Set a bit in the mask
3         BEGIN
3         EXTERNAL ROUTINE
3             PRV$SETPRIV                    ! Set privilege bit in mask
3             ;
3         IF  NOT
3             PRV$SETPRIV                    ! Set privilege bit according
3                 (                          ! to the token string
3                 TKNCNT,                    ! Address of the token descriptor
3                 .PTR+1                     ! Address of the mask (quadword)
3                 )
3         THEN
3             RETURN FAILURE                 ! Report syntax error
3         END
2         ;
```

```
2
2 !
2 !       Subaddress range
2 !
2
2     [PBK$K_SAD]:
3         BEGIN
3         (.PTR) <0,16,0> = .ACT$GL_SAD_BEGIN;
3         (.PTR) <16,16,0> = .ACT$GL_SAD_END;
2         END;
2
2 !
2 !       Range list
2 !
2
2     [PBK$K_RNGL]:
3         BEGIN                           ! copy the channel list vector
3         (.PTR) <0,16,0> = .ACT$GA_RNGLST [0];   ! Count of range elements
3         PTR = .PTR + 2;
3         INCR IDX FROM 1 TO .ACT$GA_RNGLST [0] DO
4             BEGIN
4             (.PTR) <0,16,0> = .ACT$GA_RNGLST [.IDX];
4             ACT$GA_RNGLST [.IDX] = 0;
4             PTR = .PTR + 2;
3             END;
3         ACT$GA_RNGLST [0] = 0;
2         END;
2
2
2     [PBK$K_AREA]:                       ! Node area
3         BEGIN
3         (.PTR) <0,8,0> = 0;             ! Indicate that it is an area
3         (.PTR) <8,8,0> = .NUM <0,8,0>;  ! the area
2         END;
2
2     [PBK$K_MODPRM] :                    ! Save a Module name
3         BEGIN
3         BIND
3             MODACC = UPLIT (%ASCIC 'X25-ACCESS') : VECTOR [,BYTE],
3             MODPRO = UPLIT (%ASCIC 'X25-PROTOCOL') : VECTOR [,BYTE],
3             MODSER = UPLIT (%ASCIC 'X25-SERVER') : VECTOR [,BYTE],
3             MOD29S = UPLIT (%ASCIC 'X29-SERVER') : VECTOR [,BYTE];
3
3         IF  CH$RCHAR (.PTR+11) NEQ 255  ! If a parameter saved here
3         THEN
3             NCP$SIG_CMDERR(NCP$_REPEAT, ! Yes, signal an error
3                     .TKNCNT, .TKNPTR,
3                     .STRCNT, .STRPTR);
3
3         SELECTONE .PBLK [PBK$L_PARAM] OF
3         SET
3             [NCP$C_ENT_MODACC]:
4                 BEGIN
4                 NCP$MOV_STR (.MODACC [0], MODACC [1], .PTR+12);
3                 END;
3             [NCP$C_ENT_MODPRO]:
4                 BEGIN
```

```
4                 NCP$MOV_STR (.MODPRO [0], MODPRO [1], .PTR+12);
3                 END;
3             [NCP$C_ENT_MODSER]:
4                 BEGIN
4                 NCP$MOV_STR (.MODSER [0], MODSER [1], .PTR+12);
3                 END;
3             [NCP$C_ENT_MOD29S]:
4                 BEGIN
4                 NCP$MOV_STR (.MOD29S [0], MOD29S [1], .PTR+12);
3                 END;
3             [OTHERWISE]:
4                 BEGIN
4                 SIGNAL_STOP (NCP$_LOGIC, 1, ASCIC (' Error while saving parameter'));
3                 END;
3         TES;
3
3         CH$WCHAR (4, .PTR+11);          ! Indicate its a Module name
2         END;
2
2     [PBK$K_NIADR]:
3         BEGIN
3         LOCAL
3             FIRST_NIBBLE,
3             BPOS,
3             LAST_BPOS;
3
3         (.PTR) <0,8,0> = 6;                     ! NI addresses are 12 characters
3         CPTR = CH$PTR (.TKNPTR);                ! Make a pointer to the string
3         BPOS = 8;                               ! Bit position of first nibble.
3         LAST_BPOS = 8 + ( (.TKNCNT-1) * 4);     ! Bit position of last nibble
3                                                 !  (not adjusted for dashes)
3 !
3 !       Step forward from first nibble used to end
3 !       of string
3 !
3
3         FIRST_NIBBLE = FALSE;                   ! Toggle which nibble of the
3                                                 !  byte will be filled
3         WHILE .BPOS LEQ .LAST_BPOS
3         DO
4             BEGIN
4                 LOCAL CHAR;
4             CHAR = CH$RCHAR_A (CPTR);   ! Obtain the character
5             IF (.CHAR EQLU '-' )                ! Remove dashes
4             THEN
5                 BEGIN
5                 LAST_BPOS = .LAST_BPOS - 4;     ! Adjust last nibble position
5                                                 !  to exclude dash.
5                 CHAR = CH$RCHAR_A (CPTR);       ! Obtain another character
4                 END;
5             IF (.CHAR GEQU 'A' )                ! Adjust the range of hex
5                 AND (.CHAR LEQU 'F')
4             THEN CHAR = .CHAR + 9;              ! Make lower nibble valid
4             IF .FIRST_NIBBLE
4             THEN
4                 (.PTR) <.BPOS -4, 4, 0> = .CHAR ! Store second char in first nibble
4             ELSE
```

```
;  796       0784  4                  (.PTR) <.BPOS +4,4,0> = .CHAR;  ! Store the first char in the second nibble
;  797       0785  4              FIRST_NIBBLE = NOT .FIRST_NIBBLE;   ! Toggle
;  798       0786  4              BPOS = .BPOS +4;                    ! Point to next nibble.
;  799       0787  4              END
;  800       0788  2          END;
;  801       0789  2
;  802       0790  2      [OUTRANGE,
;  803       0791  2          PBK$K_END,
;  804       0792  2          PBK$K_TRIPL,                 ! These parameter types are for read only parameters which need spec
;  805       0793  2          PBK$K_DELTIM,                !  Formatting in NCPSHOLIS, but which don't need to be saved by ACT$
;  806       0794  2          PBK$K_DAYTIM,
;  807       0795  2          PBK$K_LITLST] :
;  808       0796  2                                       ! For any error in coding block
;  809       0797  2          RETURN FAILURE;              ! Report a syntax error
;  810       0798  2
;  811       0799  2      TES;
;  812       0800  2
;  813       0801  2      RETURN SUCCESS
;  814       0802  2
;  815       0803  1      END;


                                                                  .TITLE  NCPVRBACT Action Routines for Verbs
                                                                  .IDENT  \V04-000\

                                                                  .PSECT  $PLIT$,NOWRT,NOEXE,2

         00 53 53 45 43 43 41 2D 35 32 58 0A  00000 P.AAA:   .ASCII  <10>\X25-ACCESS\<0>
00 00 4C 4F 43 4F 54 4F 52 50 2D 35 32 58 0C  0000C P.AAB:   .ASCII  <12>\X25-PROTOCOL\<0><0><0>
                                          00  0001B
         00 52 45 56 52 45 53 2D 35 32 58 0A  0001C P.AAC:   .ASCII  <10>\X25-SERVER\<0>
         00 52 45 56 52 45 53 2D 39 32 58 0A  00028 P.AAD:   .ASCII  <10>\X29-SERVER\<0>
73 20 65 6C 69 68 77 20 72 6F 72 72 45 20 1D  00034 P.AAE:   .ASCII  <29>\ Error while saving parameter\
72 65 74 65 6D 61 72 61 70 20 67 6E 69 76 61  00043

                                                                  .PSECT  $OWN$,NOEXE,2

                                              00000 EVELST:  .BLKB   4

                                                                  .PSECT  $GLOBAL$,NOEXE,2

                                              00000 NCP$GL_ENTITY::
                                                             .BLKB   4
                                              00004 NCP$GL_MODTYP::
                                                             .BLKB   4
                                              00008 NCP$GW_PRMTYP::
                                                             .BLKB   4
                                              0000C NCP$GL_QUALPRS::
                                                             .BLKB   4
                                              00010 NCP$GL_NOPARMS::
                                                             .BLKB   4
                                              00014 NCP$GT_MSGBFR::
                                                             .BLKB   1000
                                              003FC ACT$GL_ADR_Q::
                                                             .BLKB   4
                                              00400 ACT$GL_NODAREA::
                                                             .BLKB   4
```

```
                                                 00404 ACT$GL_SAD_BEGIN::
                                                                 .BLKB   4
                                                 00408 ACT$GL_SAD_END::
                                                                 .BLKB   4

                                                       MODACC=               P.AAA
                                                       MODPRO=               P.AAB
                                                       MODSER=               P.AAC
                                                       MOD29S=               P.AAD
                                                             .EXTRN   ACT$C_RNGLSTMAX
                                                             .EXTRN   NCP$_ACCCIR, NCP$_ACCLIN
                                                             .EXTRN   NCP$_INVEVE, NCP$_INVRSP
                                                             .EXTRN   NCP$_LOGIC, NCP$_NOPARM
                                                             .EXTRN   NCP$_REPEAT, NCP$_V2COMP
                                                             .EXTRN   ACT$GA_RNGLST, PDB$G_VRB_ALL
                                                             .EXTRN   PDB$G_LOO_ACC, PDB$G_LOO_PSW
                                                             .EXTRN   PDB$G_LOO_USR, NCP$GL_FNC_CODE
                                                             .EXTRN   NCP$GL_OPTION, NCP$GL_EXECCB
                                                             .EXTRN   NCP$SIG_CMDERR, NCP$OPENLINK
                                                             .EXTRN   NCP$SENDMSG, NCP$READRSP
                                                             .EXTRN   NCP$OPENSHO, NCP$SHOHEAD
                                                             .EXTRN   NCP$SHOLIS, NCP$WRITESHO
                                                             .EXTRN   NCP$CLOSESHO, PBK$G_VRB_XID
                                                             .EXTRN   ACT$GL_NO_XAREA_Q
                                                             .EXTRN   PRV$SETPRIV

                                                             .PSECT   $CODE$,NOWRT,2

                                            0FFC 00000       .ENTRY   ACT$SAVPRM, Save R2,R3,R4,R5,R6,R7,R8,R9,-   ; 0260
                                                                      R10,R11                                      ;
                         5B 00000000G  8F   D0 00002         MOVL     #NCP$_REPEAT, R11                            ;
                         5A 00000000G  00   9E 00009         MOVAB    NCP$SIG_CMDERR, R10                          ;
                         59 00000000'  00   9E 00010         MOVAB    MODACC+1, R9                                 ;
                         58 00000000'  00   9E 00017         MOVAB    ACT$GL_ADR_Q, R8                             ;
                         57        20  AC   D0 0001E         MOVL     PBLK, R7                                     ; 0335
                         56        01  A7   D0 00022         MOVL     1(R7), PTR                                   ;
                         23            66   E9 00026         BLBC     (PTR), 2$                                    ; 0337
                         0E            67   91 00029         CMPB     (R7), #14                                    ; 0344
                                       05   1F 0002C         BLSSU    1$                                           ;
                         15            67   91 0002E         CMPB     (R7), #21                                    ;
                                       19   1B 00031         BLEQU    2$                                           ;
                         1A            67   91 00033 1$:     CMPB     (R7), #26                                    ;
                                       14   13 00036         BEQL     2$                                           ;
                         0C            67   91 00038         CMPB     (R7), #12                                    ; 0345
                                       0F   13 0003B         BEQL     2$                                           ;
                         7E        08  AC   7D 0003D         MOVQ     STRCNT, -(SP)                                ; 0352
                         7E        10  AC   7D 00041         MOVQ     TKNCNT, -(SP)                                ; 0351
                                       5B   DD 00045         PUSHL    R11                                          ; 0350
                         6A            05   FB 00047         CALLS    #5, NCP$SIG_CMDERR                           ;
                                       4D   11 0004A         BRB      4$                                           ; 0353
                         86            01   90 0004C 2$:     MOVB     #1, (PTR)+                                   ; 0357
               22        01            67   8F 0004F         CASEB    (R7), #1, #34                                ; 0377
      0055    017F       004F        0049      00053 3$:     .WORD    5$-3$,-                                      ;
      00D4    007B       0061        00AF      0005B                  6$-3$,-                                      ;
      028E    005B       03B2        0069      00063                  33$-3$,-                                     ;
      0185    017F       0172        0282      0006B                  7$-3$,-                                      ;
      0219    01D0       03AE        0195      00073                  15$-3$,-                                     ;
```

```
     02A1       03B2       00B3       0269     0007B                       9$-3$,-                         ;
     011A       02AC       0241       00A7     00083                       12$-3$,-                        ;
     03B2       0356       0094       02D3     0008B                       21$-3$,-                        ;
                02DC       03B2       03B2     00093                       10$-3$,-                        ;
                                                                           79$-3$,-                        ;
                                                                           8$-3$,-                         ;
                                                                           55$-3$,-                        ;
                                                                           53$-3$,-                        ;
                                                                           32$-3$,-                        ;
                                                                           33$-3$,-                        ;
                                                                           34$-3$,-                        ;
                                                                           36$-3$,-                        ;
                                                                           78$-3$,-                        ;
                                                                           42$-3$,-                        ;
                                                                           46$-3$,-                        ;
                                                                           51$-3$,-                        ;
                                                                           16$-3$,-                        ;
                                                                           79$-3$,-                        ;
                                                                           56$-3$,-                        ;
                                                                           14$-3$,-                        ;
                                                                           49$-3$,-                        ;
                                                                           58$-3$,-                        ;
                                                                           26$-3$,-                        ;
                                                                           61$-3$,-                        ;
                                                                           13$-3$,-                        ;
                                                                           72$-3$,-                        ;
                                                                           79$-3$,-                        ;
                                                                           79$-3$,-                        ;
                                                                           79$-3$,-                        ;
                                                                           63$-3$                          ;
                                        0369  31 00099 4$:      BRW        79$                             ; 0797
                           66       05    A7  90 0009C 5$:      MOVB       5(R7), (PTR)                    ; 0373
                                          73  11 000A0          BRB        17$                             ; 0377
                           66       1C    AC  90 000A2 6$:      MOVB       NUM, (PTR)                      ;
                                          7D  11 000A6          BRB        20$                             ;
                           66       1C    AC  D0 000A8 7$:      MOVL       NUM, (PTR)                      ; 0385
                                          77  11 000AC          BRB        20$                             ;
                           66       05    A7  D0 000AE 8$:      MOVL       5(R7), (PTR)                    ; 0389
                                          71  11 000B2          BRB        20$                             ;
                                          56  DD 000B4 9$:      PUSHL      PTR                             ; 0407
                           7E       10    AC  7D 000B6          MOVQ       TKNCNT, -(SP)                   ;
                                          09  11 000BA          BRB        11$                             ;
                                          56  DD 000BC 10$:     PUSHL      PTR                             ; 0415
                           50       05    A7  D0 000BE          MOVL       5(R7), R0                       ; 0414
                           7E             60  7D 000C2          MOVQ       (R0), -(SP)                     ; 0413
                 00000000V 00             03  FB 000C5 11$:     CALLS      #3, NCP$MOV_QSTR                ;
                                          57  11 000CC          BRB        20$                             ; 0412
                           31             68  E9 000CE 12$:     BLBC       ACT$GL_ADR_Q, 15$               ; 0421
                                          68  D4 000D1          CLRL       ACT$GL_ADR_Q                    ; 0427
                                          66  94 000D3          CLRB       (PTR)                           ; 0428
             50       04   A8             0A  78 000D5          ASHL       #10, ACT$GL_NODAREA, R0         ; 0430
                           51       1C    AC  3C 000DA          MOVZWL     NUM, R1                         ;
                           50             51  C0 000DE          ADDL2      R1, NODE_AREA                   ;
                      01   A6             50  B0 000E1          MOVW       NODE_AREA, 1(PTR)               ; 0431
                                          3E  11 000E5          BRB        20$                             ; 0420
                                          68  D4 000E7 13$:     CLRL       ACT$GL_ADR_Q                    ; 0443
             50       04   A8             0A  78 000E9          ASHL       #10, ACT$GL_NODAREA, R0         ; 0445
```

```
                               51      1C   AC  3C 000EE          MOVZWL  NUM, R1                                    ;
                               50           51  C0 000F2          ADDL2   R1, NODE_AREA                              ;
                               66           50  B0 000F5          MOVW    NODE_AREA, (PTR)                           ; 0446
                                            71  11 000F8          BRB     25$                                        ; 0377
                               05           68  E9 000FA 14$:     BLBC    ACT$GL_ADR_Q, 15$                          ; 0450
                                            68  D4 000FD          CLRL    ACT$GL_ADR_Q                               ; 0453
                                          0224  31 000FF          BRW     61$                                        ; 0454
                                            56  DD 00102 15$:     PUSHL   PTR                                        ; 0458
                                            14  11 00104          BRB     19$                                        ;
                                            66  94 00106 16$:     CLRB    (PTR)                                      ; 0462
                               0C           68  E9 00108          BLBC    ACT$GL_ADR_Q, 18$                          ; 0463
                                            68  D4 0010B          CLRL    ACT$GL_ADR_Q                               ; 0466
                                       01   A6  94 0010D          CLRB    1(PTR)                                     ; 0467
                       02      A6      1C   AC  B0 00110          MOVW    NUM, 2(PTR)                                ; 0468
                                            54  11 00115 17$:     BRB     25$                                        ; 0463
                                       01   A6  9F 00117 18$:     PUSHAB  1(PTR)                                     ; 0471
                               7E      10   AC  7D 0011A 19$:     MOVQ    TKNCNT, -(SP)                              ;
                  00000000V    00           03  FB 0011E          CALLS   #3, NCP$MOV_STR                            ;
                                            44  11 00125 20$:     BRB     25$                                        ; 0377
             50        10      AC           01  C1 00127 21$:     ADDL3   #1, TKNCNT, R0                             ; 0477
                               50           02  C6 0012C          DIVL2   #2, R0                                     ;
                               66           50  90 0012F          MOVB    R0, (PTR)                                  ;
                                       01 A046  94 00132          CLRB    1(R0)[PTR]                                 ; 0480
                               54      14   AC  D0 00136          MOVL    TKNPTR, CPTR                               ; 0481
                               50      10   AC  D0 0013A          MOVL    TKNCNT, R0                                 ; 0488
                               50           04  C4 0013E          MULL2   #4, R0                                     ;
                               50           08  C0 00141          ADDL2   #8, BPOS                                   ;
                                            1D  11 00144          BRB     24$                                        ;
                               51           84  9A 00146 22$:     MOVZBL  (CPTR)+, CHAR                              ; 0494
                   00000041    8F           51  D1 00149          CMPL    CHAR, #65                                  ; 0495
                                            0C  1F 00150          BLSSU   23$                                        ;
                   00000046    8F           51  D1 00152          CMPL    CHAR, #70                                  ; 0496
                                            03  1A 00159          BGTRU   23$                                        ;
                               51           09  C0 0015B          ADDL2   #9, CHAR                                   ; 0497
    66       04                50           51  F0 0015E 23$:     INSV    CHAR, BPOS, #4, (PTR)                      ; 0498
                               50           04  C2 00163 24$:     SUBL2   #4, BPOS                                   ;
                               08           50  D1 00166          CMPL    BPOS, #8                                   ;
                                            DB  1E 00169          BGEQU   22$                                        ;
                                            79  11 0016B 25$:     BRB     35$                                        ; 0488
             50        10      AC           01  C1 0016D 26$:     ADDL3   #1, TKNCNT, R0                             ; 0509
             51                50           02  C7 00172          DIVL3   #2, R0, R1                                 ;
                               66           51  9B 00176          MOVZBW  R1, (PTR)                                  ;
                               54      14   AC  D0 00179          MOVL    TKNPTR, CPTR                               ; 0512
                                            53  D4 0017D          CLRL    FIRST_NIBBLE                               ; 0519
                               50      10   AC  D0 0017F          MOVL    TKNCNT, R0                                 ; 0522
                               50           04  C4 00183          MULL2   #4, R0                                     ;
                               50           04  C0 00186          ADDL2   #4, R0                                     ;
                               55           08  D0 00189          MOVL    #8, BPOS                                   ; 0521
                                            30  11 0018C          BRB     31$                                        ;
                               51           84  9A 0018E 27$:     MOVZBL  (CPTR)+, CHAR                              ; 0527
                   00000041    8F           51  D1 00191          CMPL    CHAR, #65                                  ; 0528
                                            0C  1F 00198          BLSSU   28$                                        ;
                   00000046    8F           51  D1 0019A          CMPL    CHAR, #70                                  ; 0529
                                            03  1A 001A1          BGTRU   28$                                        ;
                               51           09  C0 001A3          ADDL2   #9, CHAR                                   ; 0530
                               06           53  E9 001A6 28$:     BLBC    FIRST_NIBBLE, 29$                          ; 0531
                               52      FC   A5  9E 001A9          MOVAB   -4(R5), R2                                 ; 0533
```

```
                                       04  11 001AD         BRB     30$                                            ;
                   52        04        A5  9E 001AF 29$:    MOVAB   4(R5), R2                                      ; 0535
      66      04   52                  51  F0 001B3 30$:    INSV    CHAR, R2, #4, (PTR)                            ;
                   53                  53  D2 001B8         MCOML   FIRST_NIBBLE, FIRST_NIBBLE                     ; 0536
                   55                  04  C0 001BB         ADDL2   #4, BPOS                                       ; 0521
                   50                  55  D1 001BE 31$:    CMPL    BPOS, R0                                       ;
                                       CB  1B 001C1         BLEQU   27$                                            ;
                                       5C  11 001C3         BRB     41$                                            ;
                   02   A6             08  90 001C5 32$:    MOVB    #8, 2(PTR)                                     ; 0547
                   0B   A6             01  8E 001C9         MNEGB   #1, 11(PTR)                                    ; 0548
                             FF        A6  94 001CD         CLRB    -1(PTR)                                        ; 0549
                                       4F  11 001D0         BRB     41$                                            ; 0377
                        66   1C        AC  B0 001D2 33$:    MOVW    NUM, (PTR)                                     ; 0554
                                       49  11 001D6         BRB     41$                                            ; 0377
           00000000'    00   1C        AC  D0 001D8 34$:    MOVL    NUM, EVELST                                    ; 0559
              3B   03   A6   1C        AC  E2 001E0         BBSS    NUM, 3(PTR), 41$                               ; 0560
                                       39  11 001E6 35$:    BRB     41$                                            ; 0377
                   1C   AC 00000000'   00  D1 001E8 36$:    CMPL    EVELST, NUM                                    ; 0565
                                       11  1B 001F0         BLEQU   37$                                            ;
                        7E   08        AC  7D 001F2         MOVQ    STRCNT, -(SP)                                  ; 0569
                        7E   10        AC  7D 001F6         MOVQ    TKNCNT, -(SP)                                  ; 0568
                           00000002G   8F  DD 001FA         PUSHL   #NCP$_INVEVE                                   ; 0567
                        6A             05  FB 00200         CALLS   #5, NCP$SIG_CMDERR                             ;
                        50 00000000'   00  D0 00203 37$:    MOVL    EVELST, IDX                                    ; 0573
                                       07  11 0020A         BRB     40$                                            ;
              00   03   A6             50  E2 0020C 38$:    BBSS    IDX, 3(PTR), 39$                               ;
                                       50  D6 00211 39$:    INCL    IDX                                            ;
                   1C   AC             50  D1 00213 40$:    CMPL    IDX, NUM                                       ;
                                       F3  1B 00217         BLEQU   38$                                            ;
           00000000'    00   1C        AC  D0 00219         MOVL    NUM, EVELST                                    ; 0575
                                       6F  11 00221 41$:    BRB     48$                                            ; 0377
                   FF   8F   0B        A6  91 00223 42$:    CMPB    11(PTR), #255                                  ; 0588
                                       0D  13 00228         BEQL    43$                                            ;
                        7E   08        AC  7D 0022A         MOVQ    STRCNT, -(SP)                                  ; 0592
                        7E   10        AC  7D 0022E         MOVQ    TKNCNT, -(SP)                                  ; 0591
                                       5B  DD 00232         PUSHL   R11                                            ; 0590
                        6A             05  FB 00234         CALLS   #5, NCP$SIG_CMDERR                             ;
                        1B 00000000G   00  E9 00237 43$:    BLBC    ACT$GL_ADR_Q, 44$                              ; 0594
                             0C        A6  94 0023E         CLRB    12(PTR)                                        ; 0599
              50   04   A8             0A  78 00241         ASHL    #10, ACT$GL_NODAREA, R0                        ; 0601
                        51   1C        AC  3C 00246         MOVZWL  NUM, R1                                        ;
                        50             51  C0 0024A         ADDL2   R1, NODE_AREA                                  ;
                   0D   A6             50  B0 0024D         MOVW    NODE_AREA, 13(PTR)                             ; 0603
                           00000000G   00  D4 00251         CLRL    ACT$GL_ADR_Q                                   ; 0604
                                       0E  11 00257         BRB     45$                                            ; 0594
                             0C        A6  9F 00259 44$:    PUSHAB  12(PTR)                                        ; 0607
                        7E   10        AC  7D 0025C         MOVQ    TKNCNT, -(SP)                                  ;
           00000000V    00             03  FB 00260         CALLS   #3, NCP$MOV_STR                                ;
                             0B        A6  94 00267 45$:    CLRB    11(PTR)                                        ; 0608
                                       73  11 0026A         BRB     54$                                            ; 0377
                   FF   8F   0B        A6  91 0026C 46$:    CMPB    11(PTR), #255                                  ; 0613
                                       0D  13 00271         BEQL    47$                                            ;
                        7E   08        AC  7D 00273         MOVQ    STRCNT, -(SP)                                  ; 0617
                        7E   10        AC  7D 00277         MOVQ    TKNCNT, -(SP)                                  ; 0616
                                       5B  DD 0027B         PUSHL   R11                                            ; 0615
                        6A             05  FB 0027D         CALLS   #5, NCP$SIG_CMDERR                             ;
                             0C        A6  9F 00280 47$:    PUSHAB  12(PTR)                                        ; 0618
```

```
                                    7E      10   AC  7D 00283          MOVQ     TKNCNT, -(SP)                          :
                        00000000V   00           03  FB 00287          CALLS    #3, NCP$MOV_STR                        :
                               0B   A6           01  90 0028E          MOVB     #1, 11(PTR)                            : 0619
                                                 69  11 00292  48$:    BRB      57$                                    : 0377
                         FF         8F      0B   A6  91 00294  49$:    CMPB     11(PTR), #255                          : 0624
                                                 0D  13 00299          BEQL     50$                                    :
                                    7E      08   AC  7D 0029B          MOVQ     STRCNT, -(SP)                          : 0628
                                    7E      10   AC  7D 0029F          MOVQ     TKNCNT, -(SP)                          : 0627
                                                 5B  DD 002A3          PUSHL    R11                                    : 0626
                                    6A           05  FB 002A5          CALLS    #5, NCP$SIG_CMDERR                     :
                                            0C   A6  9F 002A8  50$:    PUSHAB   12(PTR)                                : 0629
                                    7E      10   AC  7D 002AB          MOVQ     TKNCNT, -(SP)                          :
                        00000000V   00           03  FB 002AF          CALLS    #3, NCP$MOV_STR                        :
                               0B   A6           03  90 002B6          MOVB     #3, 11(PTR)                            : 0630
                                                 71  11 002BA          BRB      62$                                    : 0377
                         FF         8F      0B   A6  91 002BC  51$:    CMPB     11(PTR), #255                          : 0635
                                                 0D  13 002C1          BEQL     52$                                    :
                                    7E      08   AC  7D 002C3          MOVQ     STRCNT, -(SP)                          : 0639
                                    7E      10   AC  7D 002C7          MOVQ     TKNCNT, -(SP)                          : 0638
                                                 5B  DD 002CB          PUSHL    R11                                    : 0637
                                    6A           05  FB 002CD          CALLS    #5, NCP$SIG_CMDERR                     :
                                            0B   A6  D4 002D0  52$:    CLRL     11(PTR)                                : 0640
                                                 58  11 002D3          BRB      62$                                    : 0377
       09              00           6E           00  2C 002D5  53$:    MOVC5    #0, (SP), #0, #9, 1(PTR)               : 0650
                                            01   A6     002DA                                                          :
                                    66           08  90 002DC          MOVB     #8, (PTR)                              : 0651
                                                 4C  11 002DF  54$:    BRB      62$                                    :
                                            01   A6  9F 002E1  55$:    PUSHAB   1(PTR)                                 : 0664
                                            10   AC  9F 002E4          PUSHAB   TKNCNT                                 : 0662
                        00000000G   00           02  FB 002E7          CALLS    #2, PRV$SETPRIV                        :
                                    3C           50  E8 002EE          BLBS     R0, 62$                                :
                                               0111  31 002F1          BRW      79$                                    : 0667
                                    66      08   A8  B0 002F4  56$:    MOVW     ACT$GL_SAD_BEGIN, (PTR)                : 0677
                               02   A6      0C   A8  B0 002F8          MOVW     ACT$GL_SAD_END, 2(PTR)                 : 0678
                                                 2E  11 002FD  57$:    BRB      62$                                    : 0377
                                    52 00000000G 00  3C 002FF  58$:    MOVZWL   ACT$GA_RNGLST, R2                      : 0687
                                    86           52  B0 00306          MOVW     R2, (PTR)+                             :
                                                 50  D4 00309          CLRL     IDX                                    : 0689
                                                 0D  11 0030B          BRB      60$                                    :
                                    51 00000000G0040 3E 0030D  59$:    MOVAW    ACT$GA_RNGLST[IDX], R1                 : 0691
                                    86           61  B0 00315          MOVW     (R1), (PTR)+                           :
                                                 61  B4 00318          CLRW     (R1)                                   : 0692
                       EF           50           52  F3 0031A  60$:    AOBLEQ   R2, IDX, 59$                           : 0689
                                       00000000G 00  B4 0031E          CLRW     ACT$GA_RNGLST                          : 0695
                                                 07  11 00324          BRB      62$                                    : 0377
                                                 66  94 00326  61$:    CLRB     (PTR)                                  : 0701
                               01   A6      1C   AC  90 00328          MOVB     NUM, 1(PTR)                            : 0702
                                                 78  11 0032D  62$:    BRB      71$                                    : 0377
                         FF         8F      0B   A6  91 0032F  63$:    CMPB     11(PTR), #255                          : 0713
                                                 0D  13 00334          BEQL     64$                                    :
                                    7E      08   AC  7D 00336          MOVQ     STRCNT, -(SP)                          : 0717
                                    7E      10   AC  7D 0033A          MOVQ     TKNCNT, -(SP)                          : 0716
                                                 5B  DD 0033E          PUSHL    R11                                    : 0715
                                    6A           05  FB 00340          CALLS    #5, NCP$SIG_CMDERR                     :
                                    50      05   A7  D0 00343  64$:    MOVL     5(R7), R0                              : 0719
                                    05           50  D1 00347          CMPL     R0, #5                                 : 0721
                                                 0B  12 0034A          BNEQ     65$                                    :
```

```
                          0C    A6  9F 0034C          PUSHAB  12(PTR)                             ; 0723
                                59  DD 0034F          PUSHL   R9                                  ;
                7E        FF    A9  9A 00351          MOVZBL  MODACC, -(SP)                       ;
                                31  11 00355          BRB     68$                                 ;
                06              50  D1 00357 65$:     CMPL    R0, #6                              ; 0725
                                0C  12 0035A          BNEQ    66$                                 ;
                          0C    A6  9F 0035C          PUSHAB  12(PTR)                             ; 0727
                          0C    A9  9F 0035F          PUSHAB  MODPRO+1                            ;
                7E        0B    A9  9A 00362          MOVZBL  MODPRO, -(SP)                       ;
                                20  11 00366          BRB     68$                                 ;
                07              50  D1 00368 66$:     CMPL    R0, #7                              ; 0729
                                0C  12 0036B          BNEQ    67$                                 ;
                          0C    A6  9F 0036D          PUSHAB  12(PTR)                             ; 0731
                          1C    A9  9F 00370          PUSHAB  MODSER+1                            ;
                7E        1B    A9  9A 00373          MOVZBL  MODSER, -(SP)                       ;
                                0F  11 00377          BRB     68$                                 ;
                09              50  D1 00379 67$:     CMPL    R0, #9                              ; 0733
                                13  12 0037C          BNEQ    69$                                 ;
                          0C    A6  9F 0037E          PUSHAB  12(PTR)                             ; 0735
                          28    A9  9F 00381          PUSHAB  MOD29S+1                            ;
                7E        27    A9  9A 00384          MOVZBL  MOD29S, -(SP)                       ;
       00000000V  00            03  FB 00388 68$:     CALLS   #3, NCP$MOV_STR                     ;
                                12  11 0038F          BRB     70$                                 ; 0719
                          33    A9  9F 00391 69$:     PUSHAB  P.AAE                               ; 0739
                                01  DD 00394          PUSHL   #1                                  ;
                     00000000G  8F  DD 00396          PUSHL   #NCP$_LOGIC                         ;
       00000000G  00            03  FB 0039C          CALLS   #3, LIB$STOP                        ;
              0B  A6            04  90 003A3 70$:     MOVB    #4, 11(PTR)                         ; 0743
                                58  11 003A7 71$:     BRB     78$                                 ; 0377
                66              06  90 003A9 72$:     MOVB    #6, (PTR)                           ; 0753
                54        14    AC  D0 003AC          MOVL    TKNPTR, CPTR                        ; 0754
                50              08  D0 003B0          MOVL    #8, BPOS                            ; 0755
                51        10    AC  D0 003B3          MOVL    TKNCNT, R1                          ; 0756
                51              04  C4 003B7          MULL2   #4, LAST_BPOS                       ;
                51              04  C0 003BA          ADDL2   #4, LAST_BPOS                       ;
                                55  D4 003BD          CLRL    FIRST_NIBBLE                        ; 0763
                51              50  D1 003BF 73$:     CMPL    BPOS, LAST_BPOS                     ; 0765
                                3D  14 003C2          BGTR    78$                                 ;
                52              84  9A 003C4          MOVZBL  (CPTR)+, CHAR                       ; 0769
                2D              52  D1 003C7          CMPL    CHAR, #45                           ; 0770
                                06  12 003CA          BNEQ    74$                                 ;
                51              04  C2 003CC          SUBL2   #4, LAST_BPOS                       ; 0773
                52              84  9A 003CF          MOVZBL  (CPTR)+, CHAR                       ; 0775
       00000041   8F            52  D1 003D2 74$:     CMPL    CHAR, #65                           ; 0777
                                0C  1F 003D9          BLSSU   75$                                 ;
       00000046   8F            52  D1 003DB          CMPL    CHAR, #70                           ; 0778
                                03  1A 003E2          BGTRU   75$                                 ;
                52              09  C0 003E4          ADDL2   #9, CHAR                            ; 0779
                06              55  E9 003E7 75$:     BLBC    FIRST_NIBBLE, 76$                   ; 0780
                53        FC    A0  9E 003EA          MOVAB   -4(R0), R3                          ; 0782
                                04  11 003EE          BRB     77$                                 ;
                53        04    A0  9E 003F0 76$:     MOVAB   4(R0), R3                           ; 0784
    66          04  53          52  F0 003F4 77$:     INSV    CHAR, R3, #4, (PTR)                 ;
                55              55  D2 003F9          MCOML   FIRST_NIBBLE, FIRST_NIBBLE          ; 0785
                50              04  C0 003FC          ADDL2   #4, BPOS                            ; 0786
                                BE  11 003FF          BRB     73$                                 ; 0765
                50              01  D0 00401 78$:     MOVL    #1, R0                              ; 0801
```

```
                                   04 00404            RET                                      ;
                              50   D4 00405 79$:       CLRL     R0                              ; 0803
                                   04 00407            RET                                      ;

; Routine Size:  1032 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL  'NCP$MOV_QSTR   Move Quoted String'
1 ROUTINE NCP$MOV_QSTR (STRCNT, STRPTR, DST) :NOVALUE =   !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Copy a quoted string to DST.  Quotes are removed if present
1 !       and trailing spaces are removed if no quotes are found.
1 !       String is converted into a counted string format
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       STRCNT, STRPTR          String descriptor for source string
1 !       DST                     Address of the area to hold counted string
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
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         CPTR1,                          ! Temporary string pointers
2         CPTR2,
2         CPEND,
2         CHAR                            ! A character as we see it
2         ;               !
2
2     CPTR1 = CH$PTR (.STRPTR);           ! Point to beginning of source
2
2     IF CH$RCHAR (.CPTR1) EQL '"'        ! Is it quoted??
2     THEN                                ! Yes
3         BEGIN
3         CPTR1 = CH$PLUS (.CPTR1, 1);    ! Skip initial quote
3         CPTR2 = CH$PTR (.DST + 1);      ! Storing pointer
3         CPEND = CH$PTR (.STRPTR +       ! Pointer to end
3                 .STRCNT);
3
3         WHILE .CPTR1 LSSA .CPEND        ! There is more string to process
3         DO
4             BEGIN
4             CHAR = CH$RCHAR_A (CPTR1);  ! Obtain a character
```

```
;   874         0861  4                 IF .CHAR EQLU '"'             . Double quotes are stripped out
;   875         0862  4                 THEN
;   876         0863  5                     BEGIN
;   877         0864  5                     CHAR = CH$RCHAR (.CPTR1);
;   878         0865  5                     IF .CHAR NEQU '"'
;   879         0866  5                     THEN  EXITLOOP             ! Single quote ends the string
;   880         0867  5                     ELSE  CPTR1 = .CPTR1 + 1 ! Advance over quote
;   881         0868  5                     END
;   882         0869  4                 ;
;   883         0870  4                 CH$WCHAR_A (.CHAR, CPTR2)    ! Write a character we save
;   884         0871  4                 END
;   885         0872  3             ;
;   886         0873  3
;   887         0874  3             CPTR1 = CH$PTR (.DST + 1);      ! Build a pointer to new string
;   888         0875  3
;   889         0876  3             (.DST) <0, 8, 0> = .CPTR2 - .CPTR1; ! Store the count for the string
;   890         0877  3
;   891         0878  3             END
;   892         0879  3
;   893         0880  2         ELSE                                ! Deal with a string without quotes
;   894         0881  2             NCP$MOV_STR (.STRCNT, .STRPTR, .DST)
;   895         0882  2         ;
;   896         0883  2
;   897         0884  2         RETURN
;   898         0885  2
;   899         0886  1         END;



                                              001C 00000 NCP$MOV_QSTR:
                                                                .WORD    Save R2,R3,R4                         ; 0805
                      50       08    AC    D0 00002             MOVL     STRPTR, CPTR1                         ; 0847
                      22             60    91 00006             CMPB     (CPTR1), #34                          ; 0849
                                     35    12 00009             BNEQ     4$                                    ;
                                     50    D6 0000B             INCL     CPTR1                                 ; 0852
          54     0C   AC             01    C1 0000D             ADDL3    #1, DST, R4                           ; 0853
                      53             54    D0 00012             MOVL     R4, CPTR2                             ;
          52     08   AC       04    AC    C1 00015             ADDL3    STRCNT, STRPTR, CPEND                 ; 0855
                      52             50    D1 0001B 1$:         CMPL     CPTR1, CPEND                          ; 0857
                                     17    1E 0001E             BGEQU    3$                                    ;
                      51             80    9A 00020             MOVZBL   (CPTR1)+, CHAR                        ; 0860
                      22             51    D1 00023             CMPL     CHAR, #34                             ; 0861
                                     0A    12 00026             BNEQ     2$                                    ;
                      51             60    9A 00028             MOVZBL   (CPTR1), CHAR                         ; 0864
                      22             51    D1 0002B             CMPL     CHAR, #34                             ; 0865
                                     07    12 0002E             BNEQ     3$                                    ;
                                     50    D6 00030             INCL     CPTR1                                 ; 0867
                      83             51    90 00032 2$:         MOVB     CHAR, (CPTR2)+                        ; 0870
                                     E4    11 00035             BRB      1$                                    ; 0857
                      50             54    D0 00037 3$:         MOVL     R4, CPTR1                             ; 0874
     0C     BC        53             50    83 0003A             SUBB3    CPTR1, CPTR2, @DST                    ; 0876
                                           04 0003F             RET                                            ; 0849
                      7E       08    AC    7D 00040 4$:         MOVQ     STRPTR, -(SP)                         ; 0881
                               04    AC    DD 00044             PUSHL    STRCNT                                ;
            00000000V 00             03    FB 00047             CALLS    #3, NCP$MOV_STR                       ;
```

```
                                                04 0004E        RET                                                        ; 0886

; Routine Size:  79 bytes,    Routine Base:  $CODE$ + 0408
```

```
1 %SBTTL  'NCP$MOV_STR    Move an Unquoted String'
1 ROUTINE NCP$MOV_STR (STRCNT, STRPTR, DST) :NOVALUE =    !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Copy a token string to DST.  Trailing spaces are removed.
1 !       String is stored in counted string format.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       STRCNT, STRPTR          String descriptor for source string
1 !       DST                     Address of target counted string
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
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         CPTR                                  ! Character pointer
2         ;               !
2
2     CPTR = .STRPTR + .STRCNT - 1;             ! Last character
2     DECRA IDX                                 ! Strip off trailing spaces
2             FROM .CPTR
2             TO .STRPTR
2     DO
2         IF  CH$RCHAR (.IDX) NEQU ' '          ! Strip trailing spaces and
2             AND
2             CH$RCHAR (.IDX) NEQU 9            ! tabs too
2         THEN                                  ! Exit with pointer to end
2             EXITLOOP CPTR = .IDX
2     ;
2
2     CH$WCHAR                                  ! Store the count
2         (
2         CH$DIFF (.CPTR, .STRPTR) + 1,         ! Size of string
2         CH$PTR (.DST)                         ! Here is the count
2         );
2
```

```
;  958            0944  2          CH$MOVE                                 . Move the string to the target
;  959            0945  2              (
;  960            0946  2              CH$DIFF (.CPTR, .STRPTR) + 1,       ! How many
;  961            0947  2              CH$PTR (.STRPTR),                   ! Source
;  962            0948  2              CH$PTR (.DST + 1)                   ! Beyond the count
;  963            0949  2              );
;  964            0950  2
;  965            0951  2          RETURN
;  966            0952  2
;  967            0953  1          END;



                                                       003C 00000 NCP$MOV_STR:
                                                                          .WORD     Save R2,R3,R4,R5                                 ; 0888
                      51        08   AC        04   AC   C1 00002         ADDL3     STRCNT, STRPTR, R1                               ; 0926
                                     50             71   9E 00008         MOVAB     -(CPTR), IDX                                     ; 0927
                                                    11   11 0000B         BRB       3$                                               ;
                                     20             60   91 0000D 1$:     CMPB      (IDX), #32                                       ; 0931
                                                    0A   13 00010         BEQL      2$                                               ;
                                     09             60   91 00012         CMPB      (IDX), #9                                        ; 0933
                                                    05   13 00015         BEQL      2$                                               ;
                                     51             50   D0 00017         MOVL      IDX, CPTR                                        ; 0935
                                                    08   11 0001A         BRB       4$                                               ;
                                                    50   D7 0001C 2$:     DECL      IDX                                              ; 0931
                                08   AC             50   D1 0001E 3$:     CMPL      IDX, STRPTR                                      ;
                                                    E9   1E 00022         BGEQU     1$                                               ;
                                     50        0C   AC   D0 00024 4$:     MOVL      DST, R0                                          ; 0941
                                     51        08   AC   C2 00028         SUBL2     STRPTR, R1                                       ; 0940
                                                    51   D6 0002C         INCL      R1                                               ;
                                     60             51   90 0002E         MOVB      R1, (R0)                                         ; 0942
                 01   A0        08   BC             51   28 00031         MOVC3     R1, @STRPTR, 1(R0)                               ; 0948
                                                         04 00037         RET                                                        ; 0953

; Routine Size:  56 bytes,    Routine Base:  $CODE$ + 0457
;  969          0954  1 %SBTTL  'ACT$CLRLONG    Clear a Longword Flag'
;  970          0955  1 GLOBAL ROUTINE ACT$CLRLONG (OPT, STRCNT, STRPTR, TKNCNT, TKNPTR,
;  971          0956  1                                   CHR, NUM, PRM) =        !
;  972          0957  1
;  973          0958  1 !++
;  974          0959  1 ! FUNCTIONAL DESCRIPTION:
;  975          0960  1 !
;  976          0961  1 !       Clear a longword flag or mask
;  977          0962  1 !
;  978          0963  1 ! FORMAL PARAMETERS:
;  979          0964  1 !
;  980          0965  1 !       Parse state table
;  981          0966  1 !       PRM             Address of the longword to clear
;  982          0967  1 !
;  983          0968  1 ! IMPLICIT INPUTS:
;  984          0969  1 !
;  985          0970  1 !       NONE
;  986          0971  1 !
;  987          0972  1 ! IMPLICIT OUTPUTS:
;  988          0973  1 !
;  989          0974  1 !       NONE
;  990          0975  1 !
;  991          0976  1 ! ROUTINE VALUE:
;  992          0977  1 ! COMPLETION CODES:
;  993          0978  1 !
;  994          0979  1 !       success
;  995          0980  1 !
;  996          0981  1 ! SIDE EFFECTS:
;  997          0982  1 !
;  998          0983  1 !       NONE
;  999          0984  1 !
; 1000          0985  1 !--
; 1001          0986  1
; 1002          0987  2     BEGIN
; 1003          0988  2
; 1004          0989  2     .PRM = 0;                       ! Clear the longword
; 1005          0990  2
; 1006          0991  2     RETURN SUCCESS
; 1007          0992  2
; 1008          0993  1     END;



                                          0000 00000          .ENTRY  ACT$CLRLONG, Save nothing                    ; 0955
                                  20   BC  D4 00002          CLRL    @PRM                                         ; 0989
                        50             01  D0 00005          MOVL    #1, R0                                       ; 0991
                                           04 00008          RET                                                  ; 0993

; Routine Size:  9 bytes,    Routine Base:  $CODE$ + 048F
```

```
0994  1 %SBTTL  'ACT$TESTLONG   Test a Longword Flag'
0995  1 GLOBAL ROUTINE ACT$TESTLONG (OPT, STRCNT, STRPTR, TKNCNT, TKNPTR,
0996  1                                  CHR, NUM, PRM) =
0997  1
0998  1 !++
0999  1 ! FUNCTIONAL DESCRIPTION:
1000  1 !
1001  1 !       Test a longword flag or mask
1002  1 !
1003  1 ! FORMAL PARAMETERS:
1004  1 !
1005  1 !       Parse state table
1006  1 !       PRM             Address of the longword to test
1007  1 !
1008  1 ! ROUTINE VALUE:
1009  1 ! COMPLETION CODES:
1010  1 !
1011  1 !       Success if the longword is non-zero
1012  1 !       Failure if the longword is zero
1013  1 !
1014  1 !--
1015  1
1016  2     BEGIN
1017  2
1018  2     IF ..PRM NEQ 0              ! If longword non-zero,
1019  2     THEN
1020  2         RETURN SUCCESS          ! then return success
1021  2     ELSE
1022  2         RETURN FAILURE;         ! else, failure
1023  2
1024  1     END;



                                0000 00000          .ENTRY  ACT$TESTLONG, Save nothing                  ; 0995
                    20    BC  D5 00002          TSTL    @PRM                                        ; 1018
                          04  13 00005          BEQL    1$                                          ;
            50            01  D0 00007          MOVL    #1, R0                                      ; 1022
                              04 0000A          RET                                                 ;
                          50  D4 0000B 1$:      CLRL    R0                                          ;
                              04 0000D          RET                                                 ; 1024

; Routine Size:  14 bytes,    Routine Base:  $CODE$ + 0498
; 1042          1025  1 %SBTTL  'ACT$COPY_VALUE Copy a longword value'
; 1043          1026  1 GLOBAL ROUTINE ACT$COPY_VALUE (OPT, STRCNT, STRPTR, TKNCNT, TKNPTR,
; 1044          1027  1                                CHR, NUM, PRM) =
; 1045          1028  1 !++
; 1046          1029  1 !
; 1047          1030  1 !       Copy a longword va    to another location.
; 1048          1031  1 !
; 1049          1032  1 ! Inputs:
; 1050          1033  1 !
; 1051          1034  1 !       Parse state table
; 1052          1035  1 !       NUM             Source longword
; 1053          1036  1 !       PRM             Address of the destination longword
; 1054          1037  1 !
; 1055          1038  1 ! Outputs:
; 1056          1039  1 !
; 1057          1040  1 !       None
; 1058          1041  1 !--
; 1059          1042  1
; 1060          1043  2 BEGIN
; 1061          1044  2
; 1062          1045  2 .PRM = .NUM;                    ! Copy the longword
; 1063          1046  2
; 1064          1047  2 RETURN SUCCESS;
; 1065          1048  2
; 1066          1049  1 END;



                                          0000 00000          .ENTRY  ACT$COPY_VALUE, Save nothing                    ; 1026
                        20   BC      1C    AC  D0 00002          MOVL    NUM, @PRM                                      ; 1045
                             50            01  D0 00007          MOVL    #1, R0                                         ; 1047
                                              04 0000A          RET                                                    ; 1049

; Routine Size:  11 bytes,    Routine Base:  $CODE$ + 04A6
; 1068      1050  1 %SBTTL  'ACT$VRB_EXIT  Action Routine to Exit NCP'
; 1069      1051  1 GLOBAL ROUTINE ACT$VRB_EXIT :NOVALUE =  !
; 1070      1052  1
; 1071      1053  1 !++
; 1072      1054  1 ! FUNCTIONAL DESCRIPTION:
; 1073      1055  1 !
; 1074      1056  1 !       This action routine leaves NCP and returns control to VMS
; 1075      1057  1 !
; 1076      1058  1 ! FORMAL PARAMETERS:
; 1077      1059  1 !
; 1078      1060  1 !       NONE
; 1079      1061  1 !
; 1080      1062  1 ! IMPLICIT INPUTS:
; 1081      1063  1 !
; 1082      1064  1 !       NONE
; 1083      1065  1 !
; 1084      1066  1 ! IMPLICIT OUTPUTS:
; 1085      1067  1 !
; 1086      1068  1 !       NONE
; 1087      1069  1 !
; 1088      1070  1 ! ROUTINE VALUE:
; 1089      1071  1 ! COMPLETION CODES:
; 1090      1072  1 !
; 1091      1073  1 !       NONE
; 1092      1074  1 !
, 1093      1075  1 ! SIDE EFFECTS:
; 1094      1076  1 !
; 1095      1077  1 !       NONE
; 1096      1078  1 !
; 1097      1079  1 !--
; 1098      1080  1
; 1099      1081  2     BEGIN
; 1100      1082  2
; 1101      1083  2 !
; 1102      1084  2 !       No cleanup is done here.  It is assumed that the system
; 1103      1085  2 !       knows how to cleanup the logical links and open channels
; 1104      1086  2 !       we have.  Cleanup will be instituted if necessary.
; 1105      1087  2 !
; 1106      1088  2
; 1107      1089  3     $EXIT ( CODE = SS$_NORMAL)          ! Use the system service
; 1108      1090  3
; 1109      1091  1     END;


                                                         .EXTRN  SYS$EXIT

                                        0000 00000       .ENTRY  ACT$VRB_EXIT, Save nothing                 ; 1051
                                      01  DD 00002       PUSHL   #1                                         ; 1089
                    00000000G  00     01  FB 00004       CALLS   #1, SYS$EXIT                               ;
                                          04 0000B       RET                                                ; 1091

; Routine Size:  12 bytes,    Routine Base:  $CODE$ + 04B1
```

```
1 %SBTTL  'ACT$VRB_UTILITY Action Routine for Most Verbs'
1 GLOBAL ROUTINE ACT$VRB_UTILITY (OPT, STRCNT, STRPTR, TKNCNT, TKNPTR,
1                                 CHR, NUM, PRM: REF BBLOCK) =    !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Action routine to perform parameter processing for almost all
1 !       functions.  The NICE messages are built and sent
1 !       to the network object.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       Parse state table
1 !       PRM             Address of the SDB table to control this routine
1 !                       The SDB has the following structure
1 !                               BYTE (TYP)      Code for type of entity.  If
1 !                                               negative, system-specific entityt
1 !                               LONG (ADR)      Address of entity PDB
1 !                               LONG (ADR)      Address of PCL list
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NICE message built and sent, response processed
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         STATUS,                         ! Returned status
2         LEN,                            ! Returned buffer length
2         BFR,                            ! Returned buffer address
2         MSGPTR;                         ! Used to build message
2 !
2 ! If we are dealing with a V2.0 NML, then reformat any circuit requests
2 ! into the appropriate line request.
2 !
2
2     IF NOT .NCP$GL_EXELCB [LCB$B_STS]   ! If link is not open yet,
2     THEN
2         NCP$OPENLINK(.NCP$GL_EXELCB);   ! Open the link; signal any errors
2
2     IF NOT V2_REQUESTS(.PRM)            ! Handle V2.0 conversion (if any)
2     THEN
2         RETURN SUCCESS;                 ! If error, exit from action routine
2
2 !
2 ! Build the message prologue fields
2 !
2
2     NCP$BLD_PROLOG(.PRM, MSGPTR);       ! Build the prolog for the message
2
2 !
2 ! Build the parameter entries in the message
2 !
```

```
; 1168        1149  2
; 1169        1150  2      NCP$BLD_PRMS (.PRM [SDB$L_PCL_ADR], MSGPTR, TRUE); ! Enable parameter check
; 1170        1151  2
; 1171        1152  2      NCP$SENDMSG                             ! Send the message to NML and check
; 1172        1153  2          (                                   ! Response for error
; 1173        1154  2          .NCP$GL_EXELCB,                     ! Link control block
; 1174        1155  2          .MSGPTR - NCP$GT_MSGBFR,            ! Length of message
; 1175        1156  2          NCP$GT_MSGBFR                       ! Message start
; 1176        1157  2          );
; 1177        1158  2
; 1178        1159  2      STATUS = NCP$READRSP                    ! Read the response from NML
; 1179        1160  2          (
; 1180        1161  2          .NCP$GL_EXELCB,                     ! LCB
; 1181        1162  2          LEN,                                ! Return the length here
; 1182        1163  2          BFR,                                ! Return buffer address here
; 1183        1164  2          FALSE                               ! Not show or list
; 1184        1165  2          );
; 1185        1166  2
; 1186        1167  2      IF .STATUS EQL NMA$C_STS_MOR            ! Beginning of multiple responses?
; 1187        1168  2      THEN
; 1188        1169  3          BEGIN
; 1189        1170  3          DO
; 1190        1171  4              BEGIN
; 1191        1172  4              IF .LEN NEQ 0                   ! Length must be zero
; 1192        1173  4              THEN
; 1193        1174  4                  SIGNAL (NCP$_INVRSP)        ! Or call out nasty
; 1194        1175  4              ;
; 1195        1176  4              STATUS = NCP$READRSP            ! Read the response from NML
; 1196        1177  4                  (
; 1197        1178  4                  .NCP$GL_EXELCB,             ! LCB
; 1198        1179  4                  LEN,                        ! Return the length here
; 1199        1180  4                  BFR,                        ! Return buffer address here
; 1200        1181  4                  FALSE                       ! Not show or list
; 1201        1182  4                  )
; 1202        1183  4              END
; 1203        1184  3          UNTIL .STATUS EQL NMA$C_STS_DON ! Read multiple until done
; 1204        1185  3          END
; 1205        1186  2      ELSE
; 1206        1187  3          BEGIN
; 1207        1188  3          IF .LEN NEQ 0                       ! Call out unclean if data here
; 1208        1189  3          THEN
; 1209        1190  3              SIGNAL (NCP$_INVRSP)            ! Call out the nasty
; 1210        1191  3          END
; 1211        1192  2      ;
; 1212        1193  2      RETURN SUCCESS                          ! Never a syntax error reported here
; 1213        1194  2
; 1214        1195  1      END;



                                  00FC 00000          .ENTRY  ACT$VRB_UTILITY, Save R2,R3,R4,R5,R6,R7        ; 1093
                      57 00000000G  00  9E 00002          MOVAB   LIB$SIGNAL, R7                                 ;
                      56 00000000G  8F  D0 00009          MOVL    #NCP$_INVRSP, R6                               ;
                      55 00000000G  00  9E 00010          MOVAB   NCP$READRSP, R5                                ;
                      54 00000000'  00  9E 00017          MOVAB   NCP$GT_MSGBFR, R4                              ;
```

```
                              53 00000000G  00  9E 0001E          MOVAB   NCP$GL_EXELCB, R3                     ;
                              5E            0C  C2 00025          SUBL2   #12, SP                               ;
                              50            63  D0 00028          MOVL    NCP$GL_EXELCB, R0                     ; 1132
                              09            60  E8 0002B          BLBS    (R0), 1$                              ;
                                            50  DD 0002E          PUSHL   R0                                    ; 1134
                     00000000G  00          01  FB 00030          CALLS   #1, NCP$OPENLINK                      ;
                              52      20    AC  D0 00037 1$:      MOVL    PRM, R2                               ; 1136
                                            52  DD 0003B          PUSHL   R2                                    ;
                     00000000V  00          01  FB 0003D          CALLS   #1, V2_REQUESTS                       ;
                              71            50  E9 00044          BLBC    R0, 5$                                ;
                                    4004    8F  BB 00047          PUSHR   #^M<R2,SP>                            ; 1144
                     00000000V  00          02  FB 0004B          CALLS   #2, NCP$BLD_PROLOG                    ;
                                            01  DD 00052          PUSHL   #1                                    ; 1150
                                      04    AE  9F 00054          PUSHAB  MSGPTR                                ;
                                      05    A2  DD 00057          PUSHL   5(R2)                                 ;
                     00000000V  00          03  FB 0005A          CALLS   #3, NCP$BLD_PRMS                      ;
                                            54  DD 00061          PUSHL   R4                                    ; 1153
                              50            64  9E 00063          MOVAB   NCP$GT_MSGBFR, R0                     ; 1155
                  7E          04  AE        50  C3 00066          SUBL3   R0, MSGPTR, -(SP)                     ;
                                            63  DD 0006B          PUSHL   NCP$GL_EXELCB                         ; 1154
                     00000000G  00          03  FB 0006D          CALLS   #3, NCP$SENDMSG                       ;
                                            7E  D4 00074          CLRL    -(SP)                                 ; 1160
                                      08    AE  9F 00076          PUSHAB  BFR                                   ;
                                      10    AE  9F 00079          PUSHAB  LEN                                   ;
                                            63  DD 0007C          PUSHL   NCP$GL_EXELCB                         ; 1161
                              65            04  FB 0007E          CALLS   #4, NCP$READRSP                       ;
                              52            50  D0 00081          MOVL    R0, STATUS                            ;
                              02            52  D1 00084          CMPL    STATUS, #2                            ; 1167
                                            25  12 00087          BNEQ    4$                                    ;
                                      08    AE  D5 00089 2$:      TSTL    LEN                                   ; 1172
                                            05  13 0008C          BEQL    3$                                    ;
                                            56  DD 0008E          PUSHL   R6                                    ; 1174
                              67            01  FB 00090          CALLS   #1, LIB$SIGNAL                        ;
                                            7E  D4 00093 3$:      CLRL    -(SP)                                 ; 1177
                                      08    AE  9F 00095          PUSHAB  BFR                                   ;
                                      10    AE  9F 00098          PUSHAB  LEN                                   ;
                                            63  DD 0009B          PUSHL   NCP$GL_EXELCB                         ; 1178
                              65            04  FB 0009D          CALLS   #4, NCP$READRSP                       ;
                              52            50  D0 000A0          MOVL    R0, STATUS                            ;
                     FFFFFF80   8F          52  D1 000A3          CMPL    STATUS, #-128                         ; 1184
                                            DD  12 000AA          BNEQ    2$                                    ;
                                            0A  11 000AC          BRB     5$                                    ; 1169
                                      08    AE  D5 000AE 4$:      TSTL    LEN                                   ; 1188
                                            05  13 000B1          BEQL    5$                                    ;
                                            56  DD 000B3          PUSHL   R6                                    ; 1190
                              67            01  FB 000B5          CALLS   #1, LIB$SIGNAL                        ;
                              50            01  D0 000B8 5$:      MOVL    #1, R0                                ; 1193
                                                04 000BB          RET                                           ; 1195

; Routine Size:  188 bytes,    Routine Base:  $CODE$ + 04BD
```

```
%SBTTL  'ACT$VRB_SHOLIS  Action Routine for Display Verbs'
GLOBAL ROUTINE ACT$VRB_SHOLIS (OPT, STRCNT, STRPTR, TKNCNT, TKNPTR,
                               CHR, NUM, PRM) =        !

!++
! FUNCTIONAL DESCRIPTION:
!
!       Action routine to perform parameter processing for show and list
!       functions.  The NICE messages are built and sent
!       to the network object.  The responses are read and parsed
!       and written to the desired output file.
!
! FORMAL PARAMETERS:
!
!       Parse state table
!       PRM             Address of the SDB table to control this routine
!                       The SDB has the following structure
!                               BYTE (TYP)      Code for type of entity.  If
!                                               negative, system-specific entity
!                               LONG (ADR)      Address of entity PDB
!                               LONG (ADR)      Address of PCL list
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       SUCCESS
!
! SIDE EFFECTS:
!
!       NICE message built and sent, response processed
!
!--

    BEGIN

    MAP
        PRM : REF BBLOCK [SDB$C_SIZE]   ! SET DEFINE Block
        ;

    LOCAL
        STATUS,                         ! Returned status
        LEN,                            ! Returned buffer length
        BFR,                            ! Returned buffer address
        INFO_RETURNED,                  ! True if non-null reply returned
        MSGPTR;                         ! Used to build message

!
!       Enable a condition handler to close the output file for show/list
!       We know that the file is not opened immediately and that a
```

```
2 !          signal may occur before the file is opened.  This will cause the
2 !          Close to be attempted before the file is opened.  The close
2 !          operation does not signal however so this is not a problem now.
2 !
2
2       ENABLE NCP$HNDL_SHOLIS;                 ! Enable handler to close file
2
2 !
2 ! If we are dealing with a V2.0 NML, then reformat SHOW CIRCUIT
2 ! into SHOW LINE.
2 !
2
2       IF NOT .NCP$GL_EXELCB [LCB$B_STS]     ! If link is not open yet,
2       THEN
2           NCP$OPENLINK(.NCP$GL_EXELCB);     ! Open the link; signal any errors
2
2       IF NOT V2_REQUESTS(.PRM)              ! Handle V2.0 conversion (if any)
2       THEN
2           RETURN SUCCESS;                   ! If error, exit from action routine
2
2 !
2 ! Build message prologue
2 !
2
2       NCP$OPENSHO ();                       ! Open output file for show or list
2
2       NCP$BLD_PROLOG(.PRM, MSGPTR);         ! Build the prolog for the message
2
2 !
2 ! Build the parameter entries in the message
2 !
2
2       NCP$BLD_PRMS (.PRM [SDB$L_PCL_ADR], MSGPTR, FALSE); ! Disable parameter check
2
2       NCP$SENDMSG                           ! Send the message to NML and check
2           (                                 ! Response for error
2           .NCP$GL_EXELCB,                   ! Link control block
2           .MSGPTR - NCP$GT_MSGBFR,          ! Length of message
2           NCP$GT_MSGBFR                     ! Message start
2           );
2
2       NCP$SHOHEAD ();                       ! Write the heading for the data
2
2       STATUS = NCP$READRSP                  ! Read the response from NML
2           (
2           .NCP$GL_EXELCB,                   ! LCB
2           LEN,                              ! Return the length here
2           BFR,                              ! Return buffer address here
2           TRUE                              ! This is a show or list
2           );
2
2       INFO_RETURNED = FALSE;                ! Assume no information returned
2
2       IF .STATUS EQL NMA$C_STS_MOR          ! Multiple responses
2       THEN
3           BEGIN
3           DO
```

```
4               BEGIN                           ! Read them all
4               IF .LEN NEQ 0                   ! Process them if data is here
4               THEN
5                   BEGIN
5                   NCP$SHOLIS(.LEN, .BFR); ! Pass the buffer length and address
5                   INFO_RETURNED = TRUE;   ! Indicate we got information back
4                   END;
4               STATUS = NCP$READRSP        ! Read the response from NML
4                   (
4                   .NCP$GL_EXELCB,         ! LCB
4                   LEN,                    ! Return the length here
4                   BFR,                    ! Return buffer address here
4                   TRUE                    ! This is a show or list
4                   );
4               IF .STATUS NEQ NMA$C_STS_SUC        ! If not successful,
4                   AND .STATUS NEQ NMA$C_STS_DON
4               THEN
4                   INFO_RETURNED = TRUE;   ! Then mark information (an error)
4                                           ! WAS returned.
4               END
3           UNTIL .STATUS EQL NMA$C_STS_DON ! Until done is in
3           END
3
2       ELSE
3           BEGIN                           ! Not multiple responses
3           IF .LEN NEQ 0                   ! Report data if any
3           THEN
4               BEGIN
4               NCP$SHOLIS (.LEN, .BFR);    ! Display information
4               INFO_RETURNED = TRUE;       ! Indicate we got information back
3               END;
3           IF .STATUS NEQ NMA$C_STS_SUC    ! If not successful,
3           THEN
3               INFO_RETURNED = TRUE;       ! Then mark information (an error)
3                                           ! WAS returned.
2           END;
2
2       IF NOT .INFO_RETURNED               ! If no information was returned,
2       THEN
3           BEGIN
3           NCP$WRITESHO(ASCID('No information in database'));
3           NCP$WRITESHO(ASCID(' '));
2           END;
2
2       NCP$CLOSESHO ();                    ! Close the output file
2
2       RETURN SUCCESS;                     ! Never a syntax error reported here
2
1       END;


                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

                                               00052      .BLKB   2
20 6E 6F 69 74 61 6D 72 6F 66 6E 69 20 6F 4E  00054 P.AAG: .ASCII  \No information in database\<0><0>
      00 00 65 73 61 62 61 74 61 64 20 6E 69  00063
```

```
                                    0000001A  00070 P.AAF:   .LONG    26                                       ;
                                    00000000' 00074          .ADDRESS P.AAG                                    ;
                         00  00  00  20  00078 P.AAI:   .ASCII   \ \<0><0><0>                               ;
                                    00000001  0007C P.AAH:   .LONG    1                                        ;
                                    00000000' 00080          .ADDRESS P.AAI                                    ;


                                                     .PSECT  $CODE$,NOWRT,2

                                      01FC 00000          .ENTRY  ACT$VRB_SHOLIS, Save R2,R3,R4,R5,R6,R7,R8   ; 1197
                 58 00000000G  00  9E 00002          MOVAB   NCP$WRITESHO, R8                            ;
                 57 00000000G  00  9E 00009          MOVAB   NCP$SHOLIS, R7                              ;
                 56 00000000G  00  9E 00010          MOVAB   NCP$READRSP, R6                             ;
                 55 00000000'  00  9E 00017          MOVAB   NCP$GT_MSGBFR, R5                           ;
                 54 00000000G  00  9E 0001E          MOVAB   NCP$GL_EXELCB, R4                           ;
                 5E            0C  C2 00025          SUBL2   #12, SP                                     ;
                 6D      00EB  CF  DE 00028          MOVAL   11$, (FP)                                   ; 1237
                 50            64  D0 0002D          MOVL    NCP$GL_EXELCB, R0                           ; 1265
                 09            60  E8 00030          BLBS    (R0), 1$                                    ;
                               50  DD 00033          PUSHL   R0                                          ; 1267
        00000000G  00          01  FB 00035          CALLS   #1, NCP$OPENLINK                            ;
                 52        20  AC  D0 0003C 1$:      MOVL    PRM, R2                                     ; 1269
                               52  DD 00040          PUSHL   R2                                          ;
        00000000V  00          01  FB 00042          CALLS   #1, V2_REQUESTS                             ;
                 03            50  E8 00049          BLBS    R0, 2$                                      ;
                             00C4  31 0004C          BRW     10$                                         ;
        00000000G  00          00  FB 0004F 2$:      CALLS   #0, NCP$OPENSHO                             ; 1277
                         4004  8F  BB 00056          PUSHR   #^M<R2,SP>                                  ; 1279
        00000000V  00          02  FB 0005A          CALLS   #2, NCP$BLD_PROLOG                          ;
                               7E  D4 00061          CLRL    -(SP)                                       ; 1285
                           04  AE  9F 00063          PUSHAB  MSGPTR                                      ;
                           05  A2  DD 00066          PUSHL   5(R2)                                       ;
        00000000V  00          03  FB 00069          CALLS   #3, NCP$BLD_PRMS                            ;
                               55  DD 00070          PUSHL   R5                                          ; 1288
                 50            65  9E 00072          MOVAB   NCP$GT_MSGBFR, R0                           ; 1290
      7E      04  AE           50  C3 00075          SUBL3   R0, MSGPTR, -(SP)                           ;
                               64  DD 0007A          PUSHL   NCP$GL_EXELCB                               ; 1289
        00000000G  00          03  FB 0007C          CALLS   #3, NCP$SENDMSG                             ;
        00000000G  00          00  FB 00083          CALLS   #0, NCP$SHOHEAD                             ; 1294
                               01  DD 0008A          PUSHL   #1                                          ; 1297
                           08  AE  9F 0008C          PUSHAB  BFR                                         ;
                           10  AE  9F 0008F          PUSHAB  LEN                                         ;
                               64  DD 00092          PUSHL   NCP$GL_EXELCB                               ; 1298
                 66            04  FB 00094          CALLS   #4, NCP$READRSP                             ;
                 53            50  D0 00097          MOVL    R0, STATUS                                  ;
                               52  D4 0009A          CLRL    INFO_RETURNED                               ; 1304
                 02            53  D1 0009C          CMPL    STATUS, #2                                  ; 1306
                               3D  12 0009F          BNEQ    6$                                          ;
                           08  AE  D5 000A1 3$:      TSTL    LEN                                         ; 1311
                               0C  13 000A4          BEQL    4$                                          ;
                           04  AE  DD 000A6          PUSHL   BFR                                         ; 1314
                           0C  AE  DD 000A9          PUSHL   LEN                                         ;
                 67            02  FB 000AC          CALLS   #2, NCP$SHOLIS                              ;
                 52            01  D0 000AF          MOVL    #1, INFO_RETURNED                           ; 1315
                               01  DD 000B2 4$:      PUSHL   #1                                          ; 1318
                           08  AE  9F 000B4          PUSHAB  BFR                                         ;
```

```
                                  10    AE   9F 000B7          PUSHAB   LEN                                      ;
                                        64   DD 000BA          PUSHL    NCP$GL_EXELCB                            ; 1319
                         66             04   FB 000BC          CALLS    #4, NCP$READRSP                          ;
                         53             50   D0 000BF          MOVL     R0, STATUS                               ;
                         01             53   D1 000C2          CMPL     STATUS, #1                               ; 1324
                                        0C   13 000C5          BEQL     5$                                       ;
               FFFFFF80  8F             53   D1 000C7          CMPL     STATUS, #-128                            ; 1325
                                        03   13 000CE          BEQL     5$                                       ;
                         52             01   D0 000D0          MOVL     #1, INFO_RETURNED                        ; 1327
               FFFFFF80  8F             53   D1 000D3 5$:      CMPL     STATUS, #-128                            ; 1330
                                        C5   12 000DA          BNEQ     3$                                       ;
                                        19   11 000DC          BRB      8$                                       ; 1308
                                  08    AE   D5 000DE 6$:      TSTL     LEN                                      ; 1335
                                        0C   13 000E1          BEQL     7$                                       ;
                                  04    AE   DD 000E3          PUSHL    BFR                                      ; 1338
                                  0C    AE   DD 000E6          PUSHL    LEN                                      ;
                         67             02   FB 000E9          CALLS    #2, NCP$SHOLIS                           ;
                         52             01   D0 000EC          MOVL     #1, INFO_RETURNED                        ; 1339
                         01             53   D1 000EF 7$:      CMPL     STATUS, #1                               ; 1341
                                        03   13 000F2          BEQL     8$                                       ;
                         52             01   D0 000F4          MOVL     #1, INFO_RETURNED                        ; 1343
                         12             52   E8 000F7 8$:      BLBS     INFO_RETURNED, 9$                        ; 1347
                            00000000'   00   9F 000FA          PUSHAB   P.AAF                                    ; 1350
                         68             01   FB 00100          CALLS    #1, NCP$WRITESHO                         ;
                            00000000'   00   9F 00103          PUSHAB   P.AAH                                    ; 1351
                         68             01   FB 00109          CALLS    #1, NCP$WRITESHO                         ;
               00000000G 00             00   FB 0010C 9$:      CALLS    #0, NCP$CLOSESHO                         ; 1354
                         50             01   D0 00113 10$:     MOVL     #1, R0                                   ; 1356
                                             04 00116          RET                                               ; 1358
                                           0000 00117 11$:     .WORD    Save nothing                             ; 1237
                                        7E   D4 00119          CLRL     -(SP)                                    ;
                                        5E   DD 0011B          PUSHL    SP                                       ;
                         7E       04    AC   7D 0011D          MOVQ     4(AP), -(SP)                             ;
               00000000V 00             03   FB 00121          CALLS    #3, NCP$HNDL_SHOLIS                      ;
                                             04 00128          RET                                               ;

; Routine Size:  297 bytes,    Routine Base:  $CODE$ + 0579
```

```
1 %SBTTL  'NCP$HNDL_SHOLIS  Handler to Close Output'
1 ROUTINE NCP$HNDL_SHOLIS (SIG, MECH, ENBL) =     !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This is a condition handler for the show/list output routines.
1 !       It closes the output file, if it sees the unwind signal
1 !       otherwise it just resignals the condition.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SIG             Address of signal array
1 !       MECH            Address of mechanism array
1 !       ENBL            Address of mechanism array (none)
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SIG : REF VECTOR,                 ! Map the arguments to vectors
2         MECH : REF VECTOR,
2         ENBL : REF VECTOR
2         ;
2
2     IF .SIG [1] EQL SS$_UNWIND          ! Check for the unwind condition
2     THEN
3         BEGIN
3         NCP$CLOSESHO ();                ! Close the show/list output file
3         RETURN SUCCESS                  ! Return value is irrelavent
3         END
2     ELSE
2         RETURN SS$_RESIGNAL             ! Resignal the condition
2
1     END;
```

```
                                        0000 00000 NCP$HNDL_SHOLIS:
                                                           .WORD   Save nothing                          ; 1360
                      50        04    AC  D0 00002         MOVL    SIG, R0                               ; 1402
            00000920  8F        04    A0  D1 00006         CMPL    4(R0), #2336                          ;
                                      0B  12 0000E         BNEQ    1$                                    ;
            0000000G  00              00  FB 00010         CALLS   #0, NCP$CLOSESHO                      ; 1405
                      50              01  D0 00017         MOVL    #1, R0                                ; 1409
                                          04 0001A         RET                                           ;
                      50      0918    8F  3C 0001B 1$:     MOVZWL  #2328, R0                             ;
                                          04 00020         RET                                           ; 1411

; Routine Size:  33 bytes,    Routine Base:  $CODE$ + 06A2
```

```
1 %SBTTL  'ACT$VRB_LOOP  Action Routine for LOOP Command'
1 GLOBAL ROUTINE ACT$VRB_LOOP (OPT, STRCNT, STRPTR, TKNCNT, TKNPTR,
1                               CHR, NUM, PSDB) =        !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Perform the loop command.  Build the message and send it.
1 !       The loop command message is similar to other message formats
1 !       but to handle access control properly we need a special action
1 !       routine.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       Parse state table
1 !       PSDB            Address of the SDB for the loop entity
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NCP$GL_FNC_CODE
1 !       NCP$GL_OPTION
1 !       PDB$G_LOO_ACC
1 !       PDB$G_LOO_PSW
1 !       PDB$G_LOO_USR
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Success or error signalled
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         PSDB : REF BBLOCK [SDB$C_SIZE]  ! Pointer to the sdb
2         ;
2
2     LOCAL
2         STATUS,                         ! Return status
2         LEN,                            ! Length of return
2         BFR,                            ! Address of return
2         MSGPTR                          ! Pointer into message
2         ;
2
2     IF .NCP$GL_OPTION [NMA$V_OPT_ACC]   ! there can be no access control
2     THEN
2         IF .NCP$GL_OPTION [NMA$V_OPT_ENT]       ! If entity is LINE
2            EQL NMA$C_ENT_LIN
```

```
                THEN
                    SIGNAL_STOP (NCP$_ACCLIN)
                ELSE
                    IF .NCP$GL_OPTION [NMA$V_OPT_ENT]   ! If entity is CIRCUIT
                        EQL NMA$C_ENT_CIR
                    THEN
                        SIGNAL_STOP (NCP$_ACCCIR);

        !
        ! If we are dealing with a V2.0 NML, then reformat any circuit requests
        ! into the appropriate line request.
        !

            IF NOT .NCP$GL_EXELCB [LCB$B_STS]   ! If link is not open yet,
            THEN
                NCP$OPENLINK(.NCP$GL_EXELCB);   ! Open the link; signal any errors

            IF NOT V2_REQUESTS(.PSDB)           ! Handle V2.0 conversion (if any)
            THEN
                RETURN SUCCESS;                 ! If error, exit from action routine

        !
        ! Build the message prologue fields
        !

            NCP$BLD_PROLOG (.PSDB, MSGPTR);     ! Build the prolog for the message

            IF .NCP$GL_OPTION [NMA$V_OPT_ACC]   ! If there is access control
            THEN                                ! store the three strings
                BEGIN
                MSGPTR =
                    CH$MOVE
                        (
                        CH$RCHAR (PDB$G_LOO_USR + 1) + 1,
                        PDB$G_LOO_USR + 1,
                        .MSGPTR
                        );
                MSGPTR =
                    CH$MOVE
                        (
                        CH$RCHAR (PDB$G_LOO_PSW + 1) + 1,
                        PDB$G_LOO_PSW + 1,
                        .MSGPTR
                        );
                MSGPTR =
                    CH$MOVE
                        (
                        CH$RCHAR (PDB$G_LOO_ACC + 1) + 1,
                        PDB$G_LOO_ACC + 1,
                        .MSGPTR
                        );
                END
            ;

            NCP$BLD_PRMS                        ! Add parameters to end of message
                (
                .PSDB [SDB$L_PCL_ADR],          ! Address of parameter list
```

```
; 1548       1526  2          MSGPTR,                           ! Message pointer
; 1549       1527  2          FALSE                             ! No parameter check
; 1550       1528  2          );
; 1551       1529  2
; 1552       1530  2      NCP$SENDMSG                           ! And send the message and check the
; 1553       1531  2          (                                 ! Response for error
; 1554       1532  2          .NCP$GL_EXELCB,                   ! Link control block
; 1555       1533  2          .MSGPTR - NCP$GT_MSGBFR,          ! Length of message
; 1556       1534  2          NCP$GT_MSGBFR                     ! Message start
; 1557       1535  2          );
; 1558       1536  2
; 1559       1537  2      STATUS = NCP$READRSP                  ! Read the response from NML
; 1560       1538  2          (
; 1561       1539  2          .NCP$GL_EXELCB,                   ! LCB
; 1562       1540  2          LEN,                              ! Return the length here
; 1563       1541  2          BFR,                              ! Return buffer address here
; 1564       1542  2          FALSE                             ! Not show or list
; 1565       1543  2          );
; 1566       1544  2
; 1567       1545  2 !
; 1568       1546  2 !        Messages not looped is reported with errors.  The returned data
; 1569       1547  2 !        is ignored here.
; 1570       1548  2 !
; 1571       1549  2
; 1572       1550  2      RETURN SUCCESS
; 1573       1551  2
; 1574       1552  1      END;



                                        0FFC 00000        .ENTRY  ACT$VRB_LOOP, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 1413
                                                                  R10,R11                                     ;
                     5B 00000000G  00   9E 00002          MOVAB   PDB$G_LOO_ACC+1, R11                        ;
                     5A 00000000G  00   9E 00009          MOVAB   PDB$G_LOO_PSW+1, R10                        ;
                     59 00000000G  00   9E 00010          MOVAB   PDB$G_LOO_USR+1, R9                         ;
                     58 00000000G  00   9E 00017          MOVAB   NCP$GL_OPTION, R8                           ;
                     57 00000000G  00   9E 0001E          MOVAB   NCP$GL_EXELCB, R7                           ;
                     5E            0C   C2 00025          SUBL2   #12, SP                                     ;
                                   68   95 00028          TSTB    NCP$GL_OPTION                               ; 1465
                                   24   18 0002A          BGEQ    3$                                          ;
       50         68       03      00   EF 0002C          EXTZV   #0, #3, NCP$GL_OPTION, R0                   ; 1467
                           01      50   D1 00031          CMPL    R0, #1                                      ; 1468
                                   08   12 00034          BNEQ    1$                                          ;
                        00000000G  8F   DD 00036          PUSHL   #NCP$_ACCLIN                                ; 1470
                                   0B   11 0003C          BRB     2$                                          ;
                           03      50   D1 0003E 1$:      CMPL    R0, #3                                      ; 1473
                                   0D   12 00041          BNEQ    3$                                          ;
                        00000000G  8F   DD 00043          PUSHL   #NCP$_ACCCIR                                ; 1475
                 00000000G  00     01   FB 00049 2$:      CALLS   #1, LIB$STOP                                ;
                            50     67   D0 00050 3$:      MOVL    NCP$GL_EXELCB, R0                           ; 1482
                            09     60   E8 00053          BLBS    (R0), 4$                                    ;
                                   50   DD 00056          PUSHL   R0                                          ; 1484
                 00000000G  00     01   FB 00058          CALLS   #1, NCP$OPENLINK                            ;
                            56   20 AC  D0 0005F 4$:      MOVL    PSDB, R6                                    ; 1486
                                   56   DD 00063          PUSHL   R6                                          ;
```

```
                   00000000V  00           01  FB 00065          CALLS    #1, V2_REQUESTS                                ;
                              71           50  E9 0006C          BLBC     R0, 6$                                         ;
                                    4040   8F  BB 0006F          PUSHR    #^M<R6,SP>                                     ; 1494
                   00000000V  00           02  FB 00073          CALLS    #2, NCP$BLD_PROLOG                             ;
                                           68  95 0007A          TSTB     NCP$GL_OPTION                                  ; 1496
                                           27  18 0007C          BGEQ     5$                                             ;
                              50           69  9A 0007E          MOVZBL   PDB$G_LOO_USR+1, R0                            ; 1502
                                           50  D6 00081          INCL     R0                                             ;
           00   BE            69           50  28 00083          MOVC3    R0, PDB$G_LOO_USR+1, @MSGPTR                   ; 1504
                              6E           53  D0 00088          MOVL     R3, MSGPTR                                     ;
                              50           6A  9A 0008B          MOVZBL   PDB$G_LOO_PSW+1, R0                            ; 1509
                                           50  D6 0008E          INCL     R0                                             ;
           00   BE            6A           50  28 00090          MOVC3    R0, PDB$G_LOO_PSW+1, @MSGPTR                   ; 1511
                              6E           53  D0 00095          MOVL     R3, MSGPTR                                     ;
                              50           6B  9A 00098          MOVZBL   PDB$G_LOO_ACC+1, R0                            ; 1516
                                           50  D6 0009B          INCL     R0                                             ;
           00   BE            6B           50  28 0009D          MOVC3    R0, PDB$G_LOO_ACC+1, @MSGPTR                   ; 1518
                              6E           53  D0 000A2          MOVL     R3, MSGPTR                                     ;
                                           7E  D4 000A5 5$:      CLRL     -(SP)                                          ; 1524
                                      04   AE  9F 000A7          PUSHAB   MSGPTR                                         ;
                                      05   A6  DD 000AA          PUSHL    5(R6)                                          ; 1525
                   00000000V  00           03  FB 000AD          CALLS    #3, NCP$BLD_PRMS                               ;
                                 00000000' 00  9F 000B4          PUSHAB   NCP$GT_MSGBFR                                  ; 1531
                              50 00000000' 00  9E 000BA          MOVAB    NCP$GT_MSGBFR, R0                              ; 1533
                7E        04  AE           50  C3 000C1          SUBL3    R0, MSGPTR, -(SP)                              ;
                                           67  DD 000C6          PUSHL    NCP$GL_EXELCB                                  ; 1532
                   00000000G  00           03  FB 000C8          CALLS    #3, NCP$SENDMSG                                ;
                                           7E  D4 000CF          CLRL     -(SP)                                          ; 1538
                                      08   AE  9F 000D1          PUSHAB   BFR                                            ;
                                      10   AE  9F 000D4          PUSHAB   LEN                                            ;
                                           67  DD 000D7          PUSHL    NCP$GL_EXELCB                                  ; 1539
                   00000000G  00           04  FB 000D9          CALLS    #4, NCP$READRSP                                ;
                              50           01  D0 000E0 6$:      MOVL     #1, R0                                         ; 1550
                                           04     000E3          RET                                                     ; 1552

; Routine Size:  228 bytes,    Routine Base:  $CODE$ + 06C3
```

```
1 %SBTTL  'V2_REQUESTS   Handle compatibility with V2.0 NML'
1 ROUTINE V2_REQUESTS (SDB: REF BBLOCK) =
1
1 !---
1 !
1 !       This routine handles the case where we are sending a request
1 !       to a V2.0 NML, which only knows about lines, instead of lines
1 !       and circuits.  We must reformat any circuit request into the
1 !       appropriate line request.
1 !
1 ! Inputs:
1 !
1 !       sdb = Address of the SDB structure
1 !
1 ! Outputs:
1 !
1 !       Routine = True if valid converted message, False if error in message
1 !---
1
2 BEGIN
2
2 LOCAL
2     PCL: REF BBLOCKVECTOR [, PCL$C_SIZE];       ! Parameter control list
2
2 BIND
2     V2_LIST = UPLIT WORD(                   ! List of circuit -> line parameters
2                 NMA$C_PCCI_STA, NMA$C_PCLI_STA,
2                 NMA$C_PCCI_SER, NMA$C_PCLI_SER,
2                 NMA$C_PCCI_LCT, NMA$C_PCLI_LCT,
2                 NMA$C_PCCI_BLO, NMA$C_PCLI_BLO,
2                 NMA$C_PCCI_COS, NMA$C_PCLI_COS,
2                 NMA$C_PCCI_TRI, NMA$C_PCLI_TRI,
2                 -1, -1): VECTOR [,WORD,SIGNED];
2
2 IF CH$NEQ(3, NCP$GL_EXELCB [LCB$B_NMLVERS],     ! If not NML V2.0,
2         3, UPLIT BYTE(2,0,0), 0)
2 THEN
2     RETURN TRUE;                            ! then leave the message stand as is
2
2 IF .SDB [SDB$B_ENT_TYP] LSS 0               ! If not a circuit request,
2     OR .NCP$GL_OPTION [NMA$V_OPT_ENT] NEQ NMA$C_ENT_CIR
2 THEN
2     RETURN TRUE;                            ! Then leave the message stand as is
2
2 NCP$GL_OPTION [NMA$V_OPT_ENT] = NMA$C_ENT_LIN;  ! Change to line request
2
2 !
2 ! Here, we can't assume that the numbering scheme between V3.0 NICE circuit
2 ! parameters and V2.0 NICE line parameters match exactly, so we must
2 ! convert the circuit parameter code to the corresponding line parameter
2 ! code in V2.0 NICE.  If the circuit parameter doesn't appear in this table,
2 ! then reject it, as the remote NML wouldn't understand it anyway.
2 !
2
2 PCL = .SDB [SDB$L_PCL_ADR];                 ! Get address of PCL array
2
2 IF .NCP$GL_FNC_CODE EQL NMA$C_FNC_CHA   ! If its a SET or DEFINE,
; 1633          1610  2 THEN
; 1634          1611  2     INCRU I FROM 0                          ! Scan all the parameters
; 1635          1612  2     DO
; 1636          1613  3         BEGIN
; 1637          1614  3         IF .PCL [.I, PCL$B_PRM_TYP] EQL PBK$K_END ! If end of parameter list,
; 1638          1615  3         THEN
; 1639          1616  3             EXITLOOP;                       ! Then we are all done
; 1640          1617  3
; 1641          1618  3         IF .BBLOCK [.PCL [.I, PCL$L_PDB_ADR], PDB$B_STS_FLG] ! If parameter specified,
; 1642          1619  3         THEN
; 1643          1620  3             INCRU J FROM 0 BY 2             ! For each entry in conversion list,
; 1644          1621  3             DO
; 1645          1622  4                 BEGIN
; 1646          1623  4                 IF .V2_LIST [.J] EQL -1 ! If end of table,
; 1647          1624  4                 THEN
; 1648          1625  5                     BEGIN
; 1649          1626  5                     SIGNAL(NCP$_V2COMP); ! Signal conversion error
; 1650          1627  5                     RETURN FALSE;
; 1651          1628  4                     END;
; 1652          1629  4                 IF .V2_LIST [.J] EQL .PCL [.I, PCL$W_PRM_ID] ! Match found?
; 1653          1630  4                 THEN
; 1654          1631  5                     BEGIN
; 1655          1632  5                     IF .PCL [.I, PCL$W_PRM_ID] NEQ .V2_LIST [.J+1] ! If not the same,
; 1656          1633  5                     THEN                    ! (assume the codes are the same
; 1657          1634  6                         BEGIN               ! because the PCL list is read-only)
; 1658          1635  6                         SIGNAL(NCP$_V2COMP); ! Signal conversion error
; 1659          1636  6                         RETURN FALSE;
; 1660          1637  5                         END;
; 1661          1638  5                     EXITLOOP;               ! Then pronounce it OK
; 1662          1639  4                     END;
; 1663          1640  3                 END;
; 1664          1641  2         END;
; 1665          1642  2
; 1666          1643  2 RETURN TRUE;
; 1667          1644  2
; 1668          1645  1 END;


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

0384  0384  032A  032A  006E  006E  0064  0064  0000  0000  00084 P.AAJ:  .WORD   0, 0, 100, 100, 110, 110, 810, 810, 900, -      ;
                                    FFFF  FFFF  0474  0474  00098                   900, 1140, 1140, -1, -1                         ;
                                                  00  00  02  000A0 P.AAK:  .BYTE   2, 0, 0                                         ;

                                                              V2_LIST=              P.AAJ


                                                                        .PSECT  $CODE$,NOWRT,2

                                                   01FC 00000 V2_REQUESTS:
                                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8                      ; 1554
                                  58 00000000G  00  9E 00002            MOVAB   NCP$GL_OPTION, R8                              ;
                                  57 00000000'  00  9E 00009            MOVAB   P.AAK, R7                                      ;
                                  50 00000000G  00  D0 00010            MOVL    NCP$GL_EXELCB, R0                              ; 1587
                 67         06    A0                03  29 00017            CMPC3   #3, 6(R0), P.AAK                               ;
                                                    72  12 0001C            BNEQ    6$                                             ;
```

```
                                   50        04   AC  D0 0001E          MOVL     SDB, R0                                  ; 1592
                                                  60  95 00022          TSTB     (R0)                                     ;
                                                  6A  19 00024          BLSS     6$                                       ;
         03           68           03             00  ED 00026          CMPZV    #0, #3, NCP$GL_OPTION, #3                ; 1593
                                                  63  12 0002B          BNEQ     6$                                       ;
         68           03           00             01  F0 0002D          INSV     #1, #0, #3, NCP$GL_OPTION                ; 1597
                                   56        05   A0  D0 00032          MOVL     5(R0), PCL                               ; 1607
                                   13 00000000G   00  D1 00036          CMPL     NCP$GL_FNC_CODE, #19                     ; 1609
                                                  51  12 0003D          BNEQ     6$                                       ;
                                                  52  D4 0003F          CLRL     I                                        ; 1614
                      50           52             07  C5 00041 1$:      MULL3    #7, I, R0                                ;
                      53           56             50  C1 00045          ADDL3    R0, PCL, R3                              ;
                                   17             63  91 00049          CMPB     (R3), #23                                ;
                                                  42  13 0004C          BEQL     6$                                       ;
                                   3A        03   B3  E9 0004E          BLBC     @3(R3), 5$                               ; 1618
                                                  54  D4 00052          CLRL     J                                        ; 1629
                                   55      E4 A744  3E 00054 2$:        MOVAW    V2_LIST[J], R5                           ; 1623
                             FFFF  8F             65  B1 00059          CMPW     (R5), #-1                                ;
                                                  18  13 0005E          BEQL     3$                                       ;
                                   50        01   A3  3C 00060          MOVZWL   1(R3), R0                                ; 1629
         50           65           10             00  EC 00064          CMPV     #0, #16, (R5), R0                        ;
                                                  1C  12 00069          BNEQ     4$                                       ;
                                   50      E6 A744  32 0006B          CVTWL    V2_LIST+2[J], R0                         ; 1632
         50     01    A3           10             00  ED 00070          CMPZV    #0, #16, 1(R3), R0                       ;
                                                  14  13 00076          BEQL     5$                                       ;
                                      00000000G   8F  DD 00078 3$:      PUSHL    #NCP$_V2COMP                             ; 1635
                        00000000G  00             01  FB 0007E          CALLS    #1, LIB$SIGNAL                           ;
                                                  0D  11 00085          BRB      7$                                       ; 1636
                                   54             02  C0 00087 4$:      ADDL2    #2, J                                    ; 1620
                                                  C8  11 0008A          BRB      2$                                       ;
                                                  52  D6 0008C 5$:      INCL     I                                        ; 1611
                                                  B1  11 0008E          BRB      1$                                       ;
                                   50             01  D0 00090 6$:      MOVL     #1, R0                                   ; 1643
                                                      04 00093          RET                                               ;
                                                  50  D4 00094 7$:      CLRL     R0                                       ; 1645
                                                      04 00096          RET                                               ;

; Routine Size:  151 bytes,    Routine Base:  $CODE$ + 07A7
```

```
1 %SBTTL  'NCP$BLD_PROLOG  Build Prolog for a Message'
1 ROUTINE NCP$BLD_PROLOG (PSDB, MSGPTR) :NOVALUE =        !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Build the prolog for most NICE command messages.
1 !       The components are: Possible system specific code,
1 !       function code, option byte and entity.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       PSDB            Address of the SDB describing the command
1 !       MSGPTR          Address of the pointer to build the message
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NCP$GL_FNC_CODE         Function code byte
1 !       NCP$GL_OPTION           Option byte
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       PSDB : REF BBLOCK [SDB$C_SIZE]
2       ;
2
2    .MSGPTR = CH$PTR (NCP$GT_MSGBFR);    ! Set the pointer to the buffer
2
2    NCP$GL_ENTITY = .PSDB [SDB$B_ENT_TYP];  ! Copy entity type code.  If
2                                         ! negative, indicates system-specific
2
2    IF .PSDB [SDB$B_ENT_TYP] LSS 0       ! Is this a system specific entity?
2    THEN
3       BEGIN                             ! Build the system specific header
3       CH$COPY
3           (
3           2,
3           UPLIT (BYTE (NMA$C_FNC_SYS,  ! System specific
3                       NMA$C_SYS_VMS    ! VMS system
3                       )
3                  ),
3           0, 2, ..MSGPTR
; 1727          1703  3              );
; 1728          1704  3
; 1729          1705  3          .MSGPTR = CH$PLUS (..MSGPTR, 2) ! Update message pointer
; 1730          1706  3          END
; 1731          1707  2      ;
; 1732          1708  2
; 1733          1709  2 !
; 1734          1710  2 !    Write the function code and option byte to the message
; 1735          1711  2 !
; 1736          1712  2
; 1737          1713  2      CH$WCHAR_A (.NCP$GL_FNC_CODE, .MSGPTR);
; 1738          1714  2      CH$WCHAR_A (.NCP$GL_OPTION, .MSGPTR);
; 1739          1715  2
; 1740          1716  2      NCP$BLD_ENTITY (.PSDB, .MSGPTR);     ! Build the entity next
; 1741          1717  2
; 1742          1718  2      RETURN
; 1743          1719  2
; 1744          1720  1      END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                          000A3         .BLKB   1
                                04  16    000A4 P.AAL:  .BYTE   22, 4                                       ;


                                                        .PSECT  $CODE$,NOWRT,2

                                     0000 00000 NCP$BLD_PROLOG:
                                                        .WORD   Save nothing                                ; 1647
                      50       08  AC  D0 00002         MOVL    MSGPTR, R0                                  ; 1687
                      60 00000000'  00  9E 00006         MOVAB   NCP$GT_MSGBFR, (R0)                         ;
           00000000'  00       04  BC  98 0000D         CVTBL   @PSDB, NCP$GL_ENTITY                        ; 1689
                                   0B  18 00015         BGEQ    1$                                          ; 1692
                 00   B0 00000000'  00  B0 00017         MOVW    P.AAL, @0(R0)                               ; 1702
                      60           02  C0 0001F         ADDL2   #2, (R0)                                    ; 1705
                 00   B0 00000000G  00  90 00022 1$:     MOVB    NCP$GL_FNC_CODE, @0(R0)                     ; 1713
                                   60  D6 0002A         INCL    (R0)                                        ;
                      50       08  AC  D0 0002C         MOVL    MSGPTR, R0                                  ; 1714
                 00   B0 00000000G  00  90 00030         MOVB    NCP$GL_OPTION, @0(R0)                       ;
                                   60  D6 00038         INCL    (R0)                                        ;
                      7E       04  AC  7D 0003A         MOVQ    PSDB, -(SP)                                 ; 1716
           0000000V   00           02  FB 0003E         CALLS   #2, NCP$BLD_ENTITY                          ;
                                       04 00045         RET                                                 ; 1720

; Routine Size:  70 bytes,    Routine Base:  $CODE$ + 083E
```

```
%SBTTL  'NCP$BLD_ENTITY  Build Entity into Message'
ROUTINE NCP$BLD_ENTITY (PSDB, MSGPTR) :NOVALUE =

!++
! FUNCTIONAL DESCRIPTION:
!
!       Build an entity description into a NICE message.
!       An entity is a format type byte followed by a byte string.
!       If the format type is negative only it is copied.
!       If the format type is zero, the following two bytes are copied.
!       If the format type is positive, that number of bytes is copied.
!       If the entity type is logging, only the format type byte is copied
!       If it is positive since only that byte is used.
!       If the entity type is area then only one of byte is copied.
!
! FORMAL PARAMETERS:
!
!       PSDB            Address of an SDB
!       MSGPTR          Address of a character pointer to build message
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! ROUTINE VALUE:
! COMPLETION CODES:
!
!       NONE
!
! SIDE EFFECTS:
!
!       NONE
!
!--

    BEGIN

    MAP
        PSDB : REF BBLOCK [SDB$C_SIZE]
        ;

    LOCAL
        ENTFMT,
        SPTR                                        ! Pointer to source
        ;               !

    SPTR = CH$PTR (.PSDB [SDB$L_ENT_ADR], 1);   ! Make it a pointer to PDB
    ENTFMT = CH$RCHAR_A (SPTR);                  ! Read the entity

    CH$WCHAR_A (.ENTFMT, .MSGPTR);              ! Copy the entity format

    IF .PSDB [SDB$B_ENT_TYP]                    ! If the entity is logging
        EQL
```

```
; 1803          1778  2             NMA$C_ENT_LOG
; 1804          1779  2         THEN                                            ! Then only save the code byte
; 1805          1780  2             RETURN
; 1806          1781  2         ;
; 1807          1782  2
; 1808          1783  2         IF .PSDB [SDB$B_ENT_TYP]                        ! If the entity type is area
; 1809          1784  2             EQL
; 1810          1785  2             NMA$C_ENT_ARE
; 1811          1786  2         AND                                             . And the ertity format is area
; 1812          1787  2             .ENTFMT <0,8,1> EQL 0                       ! number
; 1813          1788  2         THEN                                            ! Then only save the area number
; 1814          1789  3             BEGIN
; 1815          1790  3             CH$MOVE (1, .SPTR, ..MSGPTR);               ! Move the one byte of area
; 1816          1791  3             .MSGPTR = CH$PLUS (..MSGPTR, 1);            ! Adr and update the pointer
; 1817          1792  3             RETURN
; 1818          1793  3             END
; 1819          1794  2         ;
; 1820          1795  2
; 1821          1796  2         IF .ENTFMT <0,8,1> EQL 0                        ! If the entity format is a
; 1822          1797  2         THEN                                            !  node adr
; 1823          1798  3             BEGIN
; 1824          1799  3             CH$MOVE (2, .SPTR, ..MSGPTR);               ! Move the two bytes of node
; 1825          1800  3             .MSGPTR = CH$PLUS (..MSGPTR, 2)             ! Adr and update the pointer
; 1826          1801  3             END
; 1827          1802  3
; 1828          1803  2         ELSE
; 1829          1804  2             IF .ENTFMT <0,8,1> GTR 0                    ! If the entity format is a
; 1830          1805  2             THEN                                        !  string
; 1831          1806  3                 BEGIN
; 1832          1807  3                 CH$MOVE (.ENTFMT, .SPTR, ..MSGPTR); ! Copy the string
; 1833          1808  3                 .MSGPTR = CH$PLUS (..MSGPTR, .ENTFMT); ! Update the pointer
; 1834          1809  3
; 1835          1810  4                 IF (.NCP$GL_ENTITY EQL NMA$C_ENT_MOD)
; 1836          1811  3                 THEN
; 1837          1812  4                     BEGIN
; 1838          1813  4                     NCP$GL_MODTYP = NCP$MODULE_TYPE (.ENTFMT, .SPTR);       ! discover what type of module it is
; 1839          1814  4                     END
; 1840          1815  3                 ELSE NCP$GL_MODTYP = 0;
; 1841          1816  3                 END
; 1842          1817  2         ;
; 1843          1818  2
; 1844          1819  2         RETURN
; 1845          1820  2
; 1846          1821  1         END;


                                                  03FC 00000 NCP$BLD_ENTITY:
                                                                    .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9               ; 1722
                      59 00000000'   00  9E 00002                   MOVAB    NCP$GL_MODTYP, R9
                      51        04   AC  D0 00009                   MOVL     PSDB, R1                                   ; 1771
        58       01   A1             01  C1 0000D                   ADDL3    #1, 1(R1), SPTR                            ;
                      57             88  9A 00012                   MOVZBL   (SPTR)+, ENTFMT                            ; 1772
                      50        08   AC  D0 00015                   MOVL     MSGPTR, R0                                 ; 1774
                 00   80             57  90 00019                   MOVB     ENTFMT, @0(R0)                             ;
```

```
                                    60  D6 0001D            INCL    (R0)                                    ;
                        02          61  91 0001F            CMPB    (R1), #2                                ; 1777
                                    48  13 00022            BEQL    4$                                      ;
                        05          61  91 00024            CMPB    (R1), #5                                ; 1784
                                    0F  12 00027            BNEQ    1$                                      ;
                                    57  95 00029            TSTB    ENTFMT                                  ; 1787
                                    0B  12 0002B            BNEQ    1$                                      ;
                        50     08   AC  D0 0002D            MOVL    MSGPTR, R0                              ; 1790
                   00   B0          68  90 00031            MOVB    (SPTR), @0(R0)                          ;
                        60          60  D6 00035            INCL    (R0)                                    ; 1791
                                        04 00037            RET                                             ; 1789
                                    57  95 00038 1$:        TSTB    ENTFMT                                  ; 1796
                                    0C  12 0003A            BNEQ    2$                                      ;
                        50     08   AC  D0 0003C            MOVL    MSGPTR, R0                              ; 1799
                   00   B0          68  B0 00040            MOVW    (SPTR), @0(R0)                          ;
                        60          02  C0 00044            ADDL2   #2, (R0)                                ; 1800
                                        04 00047            RET                                             ;
                                    22  15 00048 2$:        BLEQ    4$                                      ; 1804
                        56     08   AC  D0 0004A            MOVL    MSGPTR, R6                              ; 1807
         00    B6       68          57  28 0004E            MOVC3   ENTFMT, (SPTR), @0(R6)                  ;
                        66          57  C0 00053            ADDL2   ENTFMT, (R6)                            ; 1808
                        04     FC   A9  D1 00056            CMPL    NCP$GL_ENTITY, #4                       ; 1810
                                    0E  12 0005A            BNEQ    3$                                      ;
                        7E          57  7D 0005C            MOVQ    ENTFMT, -(SP)                           ; 1813
              00000000V 00          02  FB 0005F            CALLS   #2, NCP$MODULE_TYPE                     ;
                        69          50  D0 00066            MOVL    R0, NCP$GL_MODTYP                       ;
                                        04 00069            RET                                             ; 1810
                                    69  D4 0006A 3$:        CLRL    NCP$GL_MODTYP                           ; 1815
                                        04 0006C 4$:        RET                                             ; 1821

; Routine Size:  109 bytes,    Routine Base:  $CODE$ + 0884
```

```
1 %SBTTL  'NCP$MODULE_TYPE  Return code for module type'
1 ROUTINE NCP$MODULE_TYPE (LEN, PTR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Compare module type string and return a code for it
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       PTR : Pointer to counted ASCII string of module entity name
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Code for module entity or zero if no match.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2     BIND
2         ENT_CNF_DSC = $DESCRIPTOR ('CONFIGURATOR') : BBLOCK,
2         ENT_CNS_DSC = $DESCRIPTOR ('CONSOLE') : BBLOCK,
2         ENT_LOA_DSC = $DESCRIPTOR ('LOADER') : BBLOCK,
2         ENT_LOO_DSC = $DESCRIPTOR ('LOOPER') : BBLOCK,
2         ENT_ACC_DSC = $DESCRIPTOR ('X25-ACCESS') : BBLOCK,
2         ENT_PRO_DSC = $DESCRIPTOR ('X25-PROTOCOL') : BBLOCK,
2         ENT_SER_DSC = $DESCRIPTOR ('X25-SERVER') : BBLOCK,
2         ENT_TRC_DSC = $DESCRIPTOR ('X25-TRACE') : BBLOCK,
2         ENT_29S_DSC = $DESCRIPTOR ('X29-SERVER') : BBLOCK;
2
2
2     IF CH$EQL (.LEN, .PTR,
2                 .ENT_CNF_DSC [DSC$W_LENGTH], .ENT_CNF_DSC [DSC$A_POINTER])
2     THEN
2         RETURN NCP$C_ENT_MODCNF         ! module Configurator
2
2
2     ELSE
2     IF CH$EQL (.LEN, .PTR,
2                 .ENT_CNS_DSC [DSC$W_LENGTH], .ENT_CNS_DSC [DSC$A_POINTER])
2     THEN
2         RETURN NCP$C_ENT_MODCNS         ! module Console
2
2
```

```
; 1905          1879  2      ELSE
; 1906          1880  2      IF CH$EQL (.LEN, .PTR,
; 1907          1881  2                 .ENT_LOA_DSC [DSC$W_LENGTH], .ENT_LOA_DSC [DSC$A_POINTER])
; 1908          1882  2      THEN
; 1909          1883  2          RETURN NCP$C_ENT_MODLOA          ! module Loader
; 1910          1884  2
; 1911          1885  2
; 1912          1886  2      ELSE
; 1913          1887  2      IF CH$EQL (.LEN, .PTR,
; 1914          1888  2                 .ENT_LOO_DSC [DSC$W_LENGTH], .ENT_LOO_DSC [DSC$A_POINTER])
; 1915          1889  2      THEN
; 1916          1890  2          RETURN NCP$C_ENT_MODLOO          ! module Looper
; 1917          1891  2
; 1918          1892  2
; 1919          1893  2      ELSE
; 1920          1894  2      IF CH$EQL (.LEN, .PTR,
; 1921          1895  2                 .ENT_ACC_DSC [DSC$W_LENGTH], .ENT_ACC_DSC [DSC$A_POINTER])
; 1922          1896  2      THEN
; 1923          1897  2          RETURN NCP$C_ENT_MODACC          ! module X25-Access
; 1924          1898  2
; 1925          1899  2
; 1926          1900  2      ELSE
; 1927          1901  2      IF CH$EQL (.LEN, .PTR,
; 1928          1902  2                 .ENT_PRO_DSC [DSC$W_LENGTH], .ENT_PRO_DSC [DSC$A_POINTER])
; 1929          1903  2      THEN
; 1930          1904  2          RETURN NCP$C_ENT_MODPRO          ! module X25-Protocol
; 1931          1905  2
; 1932          1906  2
; 1933          1907  2      ELSE
; 1934          1908  2      IF CH$EQL (.LEN, .PTR,
; 1935          1909  2                 .ENT_SER_DSC [DSC$W_LENGTH], .ENT_SER_DSC [DSC$A_POINTER])
; 1936          1910  2      THEN
; 1937          1911  2          RETURN NCP$C_ENT_MODSER          ! module X25-Server
; 1938          1912  2
; 1939          1913  2
; 1940          1914  2      ELSE
; 1941          1915  2      IF CH$EQL (.LEN, .PTR,
; 1942          1916  2                 .ENT_TRC_DSC [DSC$W_LENGTH], .ENT_TRC_DSC [DSC$A_POINTER])
; 1943          1917  2      THEN
; 1944          1918  2          RETURN NCP$C_ENT_MODTRC          ! module X25-Trace
; 1945          1919  2
; 1946          1920  2
; 1947          1921  2      ELSE
; 1948          1922  2      IF CH$EQL (.LEN, .PTR,
; 1949          1923  2                 .ENT_29S_DSC [DSC$W_LENGTH], .ENT_29S_DSC [DSC$A_POINTER])
; 1950          1924  2      THEN
; 1951          1925  2          RETURN NCP$C_ENT_MOD29S;         ! module X29-Server
; 1952          1926  2
; 1953          1927  2      RETURN FALSE;                                 ! Not matched
; 1954          1928  1      END;          ! Routine NCP$MODULE_TYPE


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

           52  4F  54  41  52  55  47  49  46  4E  4F  43  000A6 P.AAN:   .ASCII  \CONFIGURATOR\                                  ;
                                                           000B2          .BLKB   2
```

```
                                         0000000C  000B4 P.AAM:   .LONG    12                                  :
                                         00000000' 000B8          .ADDRESS P.AAN                               :
                       45  4C  4F  53  4E  4F  43  000BC P.AAP:   .ASCII   \CONSOLE\                           :
                                                   000C3          .BLKB    1
                                         00000007  000C4 P.AAO:   .LONG    7                                   :
                                         00000000' 000C8          .ADDRESS P.AAP                               :
                           52  45  44  41  4F  4C  000CC P.AAR:   .ASCII   \LOADER\                            :
                                                   000D2          .BLKB    2
                                         00000006  000D4 P.AAQ:   .LONG    6                                   :
                                         00000000' 000D8          .ADDRESS P.AAR                               :
                           52  45  50  4F  4F  4C  000DC P.AAT:   .ASCII   \LOOPER\                            :
                                                   000E2          .BLKB    2
                                         00000006  000E4 P.AAS:   .LONG    6                                   :
                                         00000000' 000E8          .ADDRESS P.AAT                               :
           53  53  45  43  43  41  2D  35  32  58  000EC P.AAV:   .ASCII   \X25-ACCESS\                        :
                                                   000F6          .BLKB    2
                                         0000000A  000F8 P.AAU:   .LONG    10                                  :
                                         00000000' 000FC          .ADDRESS P.AAV                               :
   4C  4F  43  4F  54  4F  52  50  2D  35  32  58  00100 P.AAX:   .ASCII   \X25-PROTOCOL\                      :
                                         0000000C  0010C P.AAW.   .LONG    12                                  :
                                         00000000' 00110          .ADDRESS P.AAX                               :
           52  45  56  52  45  53  2D  35  32  58  00114 P.AAZ:   .ASCII   \X25-SERVER\                        :
                                                   0011E          .BLKB    2
                                         0000000A  00120 P.AAY:   .LONG    10                                  :
                                         00000000' 00124          .ADDRESS P.AAZ                               :
               45  43  41  52  54  2D  35  32  58  00128 P.ABB:   .ASCII   \X25-TRACE\                         :
                                                   00131          .BLKB    3
                                         00000009  00134 P.ABA:   .LONG    9                                   :
                                         00000000' 00138          .ADDRESS P.ABB                               :
           52  45  56  52  45  53  2D  39  32  58  0013C P.ABD:   .ASCII   \X29-SERVER\                        :
                                                   00146          .BLKB    2
                                         0000000A  00148 P.ABC:   .LONG    10                                  :
                                         00000000' 0014C          .ADDRESS P.ABD                               :

                                                         ENT_CNF_DSC=          P.AAM
                                                         ENT_CNS_DSC=          P.AAO
                                                         ENT_LOA_DSC=          P.AAQ
                                                         ENT_LOO_DSC=          P.AAS
                                                         ENT_ACC_DSC=          P.AAU
                                                         ENT_PRO_DSC=          P.AAW
                                                         ENT_SER_DSC=          P.AAY
                                                         ENT_TRC_DSC=          P.ABA
                                                         ENT_29S_DSC=          P.ABC


                                                                  .PSECT   $CODE$,NOWRT,2

                                            007C 00000 NCP$MODULE_TYPE:
                                                                  .WORD    Save R2,R3,R4,R5,R6                 : 1823
                              56 00000000'  00  9E 00002          MOVAB    ENT_CNF_DSC+4, R6                   :
                              55        04  AC  D0 00009          MOVL     LEN, R5                             : 1866
                              54        08  AC  D0 0000D          MOVL     PTR, R4                             :
                              50            66  D0 00011          MOVL     ENT_CNF_DSC+4, R0                   : 1867
  FC   A6            00       64            55  2D 00014          CMPC5    R5, (R4), #0, ENT_CNF_DSC, (R0)     : 1866
                                            60     0001A                                                       :
                                            04  12 0001B          BNEQ     1$                                  :
                              50            01  D0 0001D          MOVL     #1, R0                              : 1869
```

```
                                                       04 00020          RET                                              ;
                               50        10        A6  D0 00021 1$:      MOVL    ENT_CNS_DSC+4, R0                        ; 1874
      0C     A6         00     64                  55  2D 00025          CMPC5   R5, (R4), #0, ENT_CNS_DSC, (R0)          ; 1873
                                                   60     0002B                                                           ;
                                                   04  12 0002C          BNEQ    2$                                       ;
                               50                  02  D0 0002E          MOVL    #2, R0                                   ; 1876
                                                       04 00031          RET                                              ;
                               50        20        A6  D0 00032 2$:      MOVL    ENT_LOA_DSC+4, R0                        ; 1881
      1C     A6         00     64                  55  2D 00036          CMPC5   R5, (R4), #0, ENT_LOA_DSC, (R0)          ; 1880
                                                   60     0003C                                                           ;
                                                   04  12 0003D          BNEQ    3$                                       ;
                               50                  03  D0 0003F          MOVL    #3, R0                                   ; 1883
                                                       04 00042          RET                                              ;
                               50        30        A6  D0 00043 3$:      MOVL    ENT_LOO_DSC+4, R0                        ; 1888
      2C     A6         00     64                  55  2D 00047          CMPC5   R5, (R4), #0, ENT_LOO_DSC, (R0)          ; 1887
                                                   60     0004D                                                           ;
                                                   04  12 0004E          BNEQ    4$                                       ;
                               50                  04  D0 00050          MOVL    #4, R0                                   ; 1890
                                                       04 00053          RET                                              ;
                               50        44        A6  D0 00054 4$:      MOVL    ENT_ACC_DSC+4, R0                        ; 1895
      40     A6         00     64                  55  2D 00058          CMPC5   R5, (R4), #0, ENT_ACC_DSC, (R0)          ; 1894
                                                   60     0005E                                                           ;
                                                   04  12 0005F          BNEQ    5$                                       ;
                               50                  05  D0 00061          MOVL    #5, R0                                   ; 1897
                                                       04 00064          RET                                              ;
                               50        58        A6  D0 00065 5$:      MOVL    ENT_PRO_DSC+4, R0                        ; 1902
      54     A6         00     64                  55  2D 00069          CMPC5   R5, (R4), #0, ENT_PRO_DSC, (R0)          ; 1901
                                                   60     0006F                                                           ;
                                                   04  12 00070          BNEQ    6$                                       ;
                               50                  06  D0 00072          MOVL    #6, R0                                   ; 1904
                                                       04 00075          RET                                              ;
                               50        6C        A6  D0 00076 6$:      MOVL    ENT_SER_DSC+4, R0                        ; 1909
      68     A6         00     64                  55  2D 0007A          CMPC5   R5, (R4), #0, ENT_SER_DSC, (R0)          ; 1908
                                                   60     00080                                                           ;
                                                   04  12 00081          BNEQ    7$                                       ;
                               50                  07  D0 00083          MOVL    #7, R0                                   ; 1911
                                                       04 00086          RET                                              ;
                               50      0080        C6  D0 00087 7$:      MOVL    ENT_TRC_DSC+4, R0                        ; 1916
      7C     A6         00     64                  55  2D 0008C          CMPC5   R5, (R4), #0, ENT_TRC_DSC, (R0)          ; 1915
                                                   60     00092                                                           ;
                                                   04  12 00093          BNEQ    8$                                       ;
                               50                  08  D0 00095          MOVL    #8, R0                                   ; 1918
                                                       04 00098          RET                                              ;
                               50      0094        C6  D0 00099 8$:      MOVL    ENT_29S_DSC+4, R0                        ; 1923
    0090     C6         00     64                  55  2D 0009E          CMPC5   R5, (R4), #0, ENT_29S_DSC, (R0)          ; 1922
                                                   60     000A5                                                           ;
                                                   04  12 000A6          BNEQ    9$                                       ;
                               50                  09  D0 000A8          MOVL    #9, R0                                   ; 1925
                                                       04 000AB          RET                                              ;
                                                   50  D4 000AC 9$:      CLRL    R0                                       ; 1927
                                                       04 000AE          RET                                              ; 1928

; Routine Size:  175 bytes,    Routine Base:  $CODE$ + 08F1


; 1955          1929  1
; 1956          1930  1
```

```
1 %SBTTL  'NCP$BLD_PRMS   Build Parameters into Message'
1 ROUTINE NCP$BLD_PRMS (PLIST, MSGPTR, CHKFLG) :NOVALUE = !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Build a list of parameters into a NICE message.
1 !       The parameters are described by a list which gives the format
1 !       of the parameter, its two byte code and the address of the
1 !       parameter data block containing the parameter data.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       PLIST           Address of the parameter descriptor list
1 !       MSGPTR          Address of the pointer to the next byte of the
1 !                       NICE message being built.
1 !       CHKFLG          True for enable check to require at least one
1 !                       parameter or ALL
1 !                       False for disable check and allow no parameters
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       PDB$G_VRB_ALL   The all parameter
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Novalue, error signaled if no parameters saved
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       PLIST :                         ! Pointer to a parameter control list
2               REF BBLOCKVECTOR [1, PCL$C_SIZE]
2       ;
2
2    LOCAL
2       PCTR,                           ! Counter of parameters used
2       PTR                             ! Local address pointer
2       ;
2
```

```
2
2
2       PCTR = 0;                               ! No parameters used yet
2       NCP$GW_PRMTYP = 0;
2
2       INCRU IDX FROM 0                        ! Scan the list
2       DO
3           BEGIN
3           IF .PLIST [.IDX, PCL$B_PRM_TYP] ! Look for end of list
3                   EQL
3              PBK$K_END
3           THEN EXITLOOP                       ! We found it so quit
3           ;
3
4           IF .(.PLIST [.IDX, PCL$L_PDB_ADR]) ! If the parameter is present
3           THEN
4               BEGIN
4
4 !
4 !         Move the parameter id and update the message pointer
4 !
4
4               (..MSGPTR) <0, 16, 0> =                 ! Save it as a word
4                   .PLIST [.IDX, PCL$W_PRM_ID];
4               .MSGPTR = ..MSGPTR + 2;                 ! Update the pointer
4
4               IF .PCTR EQL 0                          ! Save first parameter code
4               THEN NCP$GW_PRMTYP = .PLIST [.IDX, PCL$W_PRM_ID];
4
4 !
4 !         Build an address to the data of the parameter
4 !
4
4               PTR = .PLIST [.IDX, PCL$L_PDB_ADR] + 1;
```

```
4
4 !
4 !       Dispatch for each type of parameter
4 !
4
4             CASE .PLIST [.IDX, PCL$B_PRM_TYP]
4                         FROM PBK$K_LOW
4                         TO PBK$K_HIGH
4                         OF
4             SET
4
4             [PBK$K_NUMB] :                          ! Number byte
4                 CH$WCHAR_A (..PTR, .MSGPTR)
4                 ;
4
4             [PBK$K_LITB] :                          ! Literal byte
4                 ;                                   ! Do nothing for CLEAR/PURGE
4                                                     ! We want only the ID
4
4             [PBK$K_NUMW] :                          ! Number word
5                 BEGIN
5                 (..MSGPTR) <0, 16, 0> = ..PTR;
5                 .MSGPTR = ..MSGPTR + 2
5                 END
4                 ;
4
4             [PBK$K_NUML, PBK$K_LITL, PBK$K_SAD] : ! Number long word
5                 BEGIN
5                 ..MSGPTR = ..PTR;
5                 .MSGPTR = ..MSGPTR + 4
5                 END
4                 ;
4
4             [PBK$K_RNGL] :                          ! Range lists
4         !
4         !       Move the parameter id and update the message pointer
4         !
5                 BEGIN
5                 LOCAL
5                     RNG_ELEMENTS : WORD;
5
5                 RNG_ELEMENTS = ..PTR;
5                 PTR = .PTR + 2;
5
5                 INCR INDX FROM 1 TO .RNG_ELEMENTS BY 2 DO
6                     BEGIN
6                     ..MSGPTR = ..PTR;                       ! Copy the range pair
6                     .MSGPTR = ..MSGPTR + 4;                 ! Update the pointer
6                     PTR = .PTR + 4;
6                     IF .INDX LSS .RNG_ELEMENTS - 1          ! more to come
6                     THEN
7                         BEGIN                               ! store parameter id again
7                         (..MSGPTR) <0, 16, 0> =             ! Save it as a word
7                             .PLIST [.IDX, PCL$W_PRM_ID];
7                         .MSGPTR = ..MSGPTR + 2;             ! Update the pointer
6                         END;
5                     END;
```

```
; 2102        2073  4              END;
; 2103        2074  4
; 2104        2075  4
```

```
 4
 4 !
 4 !        Any type of counted string
 4 !
 4
 4          [PBK$K_TKN, PBK$K_TKNQ, PBK$K_STRQ, PBK$K_HXPS, PBK$K_HEX,
 4              PBK$K_NIADR, PBK$K_PRVL, PBK$K_PRVC] :
 4              IF CH$RCHAR(.PTR) GTRU 127      ! If plural form (ACTIVE, KNOWN),
 4              THEN                            ! then copy plural form byte
 4                  CH$WCHAR_A (CH$RCHAR_A (PTR), .MSGPTR)
 4              ELSE                            ! else copy ascic string
 5                  BEGIN
 5                  CH$MOVE(CH$RCHAR(.PTR)+1, .PTR, ..MSGPTR);
 5                  .MSGPTR = ..MSGPTR + CH$RCHAR(.PTR) + 1;
 4                  END;
```

```
!
!       Node address or name
!

            [PBK$K_NADR] :
                BEGIN
                IF CH$RCHAR (.PTR) EQL 0        ! If its an address
                THEN                            ! Copy the address
                    BEGIN
                    CH$MOVE (3, .PTR, ..MSGPTR);
                    .MSGPTR = CH$PLUS (..MSGPTR, 3)
                    END
                ELSE                            ! Copy the name otherwise
                    BEGIN
                    IF CH$RCHAR (.PTR) GTRU 127 ! Is it a plural form?
                    THEN                        ! Copy the plural byte code
                        CH$WCHAR_A (CH$RCHAR_A (PTR), .MSGPTR)
                    ELSE                        ! Copy the string
                        BEGIN
                        CH$MOVE
                            (.(.PTR) <0,8,0> + 1,
                            .PTR, ..MSGPTR)
                        ;
                        .MSGPTR =
                            CH$PLUS
                                (..MSGPTR,
                                .(.PTR) <0,8,0> + 1)
                        END
                    END
                END
                ;



            [PBK$K_AADR] :                      ! Node area and address in a word
                BEGIN
                (..MSGPTR) <0, 16, 0> = ..PTR;
                .MSGPTR = ..MSGPTR + 2
                END;
```

```
; 2163        2131  4
; 2164        2132  4 !
; 2165        2133  4 !       Object number or name
; 2166        2134  4 !
; 2167        2135  4
; 2168        2136  4             [PBK$K_OBJ]:
; 2169        2137  4                 IF CH$RCHAR (.PTR) EQL 0        ! If its an address
; 2170        2138  4                 THEN                            ! Copy the number (byte)
; 2171        2139  5                     BEGIN
; 2172        2140  5                     CH$MOVE (2, .PTR, ..MSGPTR);
; 2173        2141  5                     .MSGPTR = CH$PLUS (..MSGPTR, 2);
; 2174        2142  5                     END
; 2175        2143  4                 ELSE                            ! Copy the name otherwise
; 2176        2144  5                     BEGIN
; 2177        2145  5                     CH$MOVE(CH$RCHAR(.PTR)+1, .PTR, ..MSGPTR);
; 2178        2146  5                     .MSGPTR = ..MSGPTR + CH$RCHAR(.PTR) + 1;
; 2179        2147  4                     END;
```

```
4
4 !
4 !      Entity type and ID
4 !
4 !      Byte of entity type code, followed by:
4 !              If node:          word(address) OR ascic(name)
4 !              Any other:        ascic(name)
4 !
4
4            [PBK$K_ENT]:
5                BEGIN
5                CH$WCHAR_A (CH$RCHAR_A (PTR), .MSGPTR); ! Copy entity type code
5                IF CH$RCHAR(.PTR-1) EQL NMA$C_ENT_NOD   ! If node entity
5                    OR CH$RCHAR (.PTR) EQL 0           ! and if its a node address
5                THEN                                    ! Copy the address
6                    BEGIN
6                    CH$MOVE (3, .PTR, ..MSGPTR);
6                    .MSGPTR = ..MSGPTR + 3;
6                    END
5                ELSE                                    ! Copy the name otherwise
6                    BEGIN
6                    IF CH$RCHAR (.PTR) GTRU 127 ! Is it a plural form?
6                    THEN                        ! Copy the plural byte code
6                        CH$WCHAR_A (CH$RCHAR_A (PTR), .MSGPTR)
6                    ELSE                        ! Copy the string
7                        BEGIN
7                        CH$MOVE(CH$RCHAR(.PTR)+1, .PTR, ..MSGPTR);
7                        .MSGPTR = ..MSGPTR + CH$RCHAR(.PTR) + 1;
6                        END;
5                    END;
4                END;
```

```
!
!       Event type codes
!

!
!       Event Parameter Format in PDB
!
!       offset  size            data
!
!       0       word            event class
!                               bits 14,15 = 3 for wild event class and events
!                               bits 14,15 = 2 for wild events
!       2       byte            size of next field
!       3       8 bytes         event type mask
!       11      byte            source type (-1 none, 0 node, 1 line, 3 circuit)
!       12      byte            source code
!       13      bytes           source entity
!

!
!       Event Parameter Format in Message
!
!       size    data
!
!       byte    Source type
!       byte    Source code
!       bytes   Source entity
!       word    Event class
!       byte    Size of event mask
!       bytes   Event mask, not present if wild class or events
!
```

```
; 2246          2211  4
; 2247          2212  4 !
; 2248          2213  4 !       Build Event Code Into a Message
; 2249          2214  4 !
; 2250          2215  4
; 2251          2216  4             [PBK$K_ESET TO PBK$K_ESEX, PBK$K_ESCI] :
; 2252          2217  5                 BEGIN
; 2253          2218  5                                         ! Write the source type
; 2254          2219  5                 CH$WCHAR_A (.(.PTR + 11), .MSGPTR);
; 2255          2220  5                 IF .(.PTR + 11) <0, 8, 1>! Look at the source type
; 2256          2221  5                     NEQ
; 2257          2222  5                     -1                  ! If its present
; 2258          2223  5                 THEN
; 2259          2224  6                     BEGIN               ! If the source type is node
; 2260          2225  7                     IF  (CH$RCHAR (.PTR + 11)
; 2261          2226  7                         EQL
; 2262          2227  7                         0)
; 2263          2228  6                         AND             ! If the node is an address
; 2264          2229  7                         (CH$RCHAR (.PTR + 12)
; 2265          2230  7                         EQL
; 2266          2231  7                         0)
; 2267          2232  6                     THEN
; 2268          2233  7                         BEGIN
; 2269          2234  7                         CH$MOVE         ! Move the address
; 2270          2235  7                             (3, .PTR + 12, ..MSGPTR)
; 2271          2236  7                         ;
; 2272          2237  7                         .MSGPTR =       ! Update the message pointer too
; 2273          2238  7                             CH$PLUS
; 2274          2239  7                                 (..MSGPTR, 3)
; 2275          2240  7                         END
; 2276          2241  6                     ELSE
; 2277          2242  7                         BEGIN           ! If the source is a token
; 2278          2243  7                         CH$MOVE         ! Move the token and its count
; 2279          2244  7                             (.(.PTR + 12) <0, 8, 0> + 1,
; 2280          2245  7                             .PTR + 12, ..MSGPTR)
; 2281          2246  7                         ;
; 2282          2247  7                         .MSGPTR =       ! And update the message pointer
; 2283          2248  7                             CH$PLUS
; 2284          2249  7                                 (..MSGPTR, .(.PTR+12) <0, 8, 0> + 1)
; 2285          2250  7                         END
; 2286          2251  6                     END
; 2287          2252  5                 ;
; 2288          2253  5 !
; 2289          2254  5 !       Write the Event class and Mask to the message last
; 2290          2255  5 !
; 2291          2256  5
; 2292          2257  5
; 2293          2258  5                 IF .(.PTR) <14, 2, 0> NEQ 0     ! Look at the wild bits
; 2294          2259  5                 THEN
; 2295          2260  6                     BEGIN                       ! Something is wild,
; 2296          2261  6                     (..MSGPTR) <0, 16, 0> = ..PTR; ! copy class only
; 2297          2262  6                     .MSGPTR = ..MSGPTR + 2
; 2298          2263  6                     END
; 2299          2264  5                 ELSE
; 2300          2265  6                     BEGIN                       ! Nothing wild, take all
; 2301          2266  6                     CH$MOVE (11, .PTR, ..MSGPTR);
; 2302          2267  6                     .MSGPTR = CH$PLUS (..MSGPTR, 11)
; 2303          2268  6                         END
; 2304          2269  5                 END
; 2305          2270  4                 ;
; 2307          2271 4
; 2308          2272 4              [PBK$K_AREA]:                       ! Node area
; 2309          2273 5                  BEGIN
; 2310          2274 5                  (..MSGPTR) <0, 16, 0> = ..PTR;
; 2311          2275 5                  .MSGPTR = ..MSGPTR + 2
; 2312          2276 5                  END
; 2313          2277 4                  ;
; 2314          2278 4
; 2315          2279 4              [OUTRANGE,                          ! For anything strange
; 2316          2280 4                PBK$K_END,
; 2317          2281 4                PBK$K_TRIPL,                      ! These parameter types are for read only parameters which n
; 2318          2282 4                PBK$K_DELTIM,                     !  Formatting in NCPSHOLIS, but which don't need to be saved
; 2319          2283 4                PBK$K_DAYTIM,
; 2320          2284 4                PBK$K_LITLST,
; 2321          2285 4
; 2322          2286 4                PBK$K_MODPRM] :                   ! Used only for saving module names
; 2323          2287 4                                                  !  as parameters, not for building messages
; 2324          2288 4                  EXITLOOP;
; 2325          2289 4
; 2326          2290 4              TES
; 2327          2291 4              ;
; 2328          2292 4              PCTR = .PCTR + 1;                   ! Count one more parameter
; 2329          2293 4              END
; 2330          2294 4
; 2331          2295 2          END;
; 2332          2296 2
; 2333          2297 2
; 2334          2298 2      IF .CHKFLG AND                              ! check is enabled
; 2335          2299 2          (NOT .PDB$G_VRB_ALL) AND                ! and ALL was not specified
; 2336          2300 3          (NOT .NCP$GL_NOPARMS)                   ! and it should have parameters
; 2337          2301 2      THEN
; 2338          2302 3          IF ((.NCP$GL_QUALPRS EQL 0) AND         ! If there are no qualifiers
; 2339          2303 3              (.PCTR EQL 0))                      ! and no parameters
; 2340          2304 2            OR
; 2341          2305 3            ((.NCP$GL_QUALPRS NEQ 0) AND          ! If there is a qualifier
; 2342          2306 3              (.PCTR LEQ 1))                      ! and only the qualifier parameter
; 2343          2307 3
; 2344          2308 2          THEN SIGNAL_STOP (NCP$_NOPARM);         ! Signal an error
; 2345          2309 2
; 2346          2310 2      RETURN
; 2347          2311 2
; 2348          2312 1      END;



                                        0FFC 00000 NCP$BLD_PRMS:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        : 1932
                   5B 00000000'   00  9E 00002     MOVAB   NCP$GW_PRMTYP, R11
                             5A   D4 00009         CLRL    PCTR                                        : 1984
                             6B   D4 0000B         CLRL    NCP$GW_PRMTYP                               : 1985
                             56   D4 0000D         CLRL    IDX                                         : 1990
         50          56      07   C5 0000F 1$:     MULL3   #7, IDX, R0                                 :
                     50      04   AC C0 00013      ADDL2   PLIST, R0                                   :
                     17      60   91 00017         CMPB    (R0), #23                                   : 1991
                             6A   13 0001A         BEQL    5$                                          :
                                03      03  B0  E8 0001C        BLBS    @3(R0), 2$                          : 1996
                                          0157  31 00020        BRW     26$                                 :
                                57      08  AC  D0 00023 2$:    MOVL    MSGPTR, R7                          : 2004
                          00    B7      01  A0  B0 00027        MOVW    1(R0), @0(R7)                       : 2005
                                67          02  C0 0002C        ADDL2   #2, (R7)                            : 2006
                                            5A  D5 0002F        TSTL    PCTR                                : 2008
                                            04  12 00031        BNEQ    3$                                  :
                                6B      01  A0  3C 00033        MOVZWL  1(R0), NCP$GW_PRMTYP                : 2009
                  58      03    A0          01  C1 00037 3$:    ADDL3   #1, 3(R0), PTR                      : 2015
                  22            01          60  8F 0003C        CASEB   (R0), #1, #34                       : 2028
      004F      0131          0049        0138     00040 4$:    .WORD   25$-4$,-                            :
      00C1      0083          00C1        00C1     00048                6$-4$,-                             :
      00C1      004F          013F        00C1     00050                24$-4$,-                            :
      00E5      00E5          00E5        00C1     00058                7$-4$,-                             :
      00E5      00E5          00E5        00E5     00060                15$-4$,-                            :
      004F      013F          00A3        00E5     00068                15$-4$,-                            :
      00C1      0058          00E5        009C     00070                11$-4$,-                            :
      013F      00C1          0131        0131     00078                15$-4$,-                            :
                013F          013F        013F     00080                15$-4$,-                            :
                                                                        27$-4$,-                            :
                                                                        7$-4$,-                             :
                                                                        15$-4$,-                            :
                                                                        15$-4$,-                            :
                                                                        21$-4$,-                            :
                                                                        21$-4$,-                            :
                                                                        21$-4$,-                            :
                                                                        21$-4$,-                            :
                                                                        21$-4$,-                            :
                                                                        21$-4$,-                            :
                                                                        21$-4$,-                            :
                                                                        21$-4$,-                            :
                                                                        13$-4$,-                            :
                                                                        27$-4$,-                            :
                                                                        7$-4$,-                             :
                                                                        12$-4$,-                            :
                                                                        21$-4$,-                            :
                                                                        8$-4$,-                             :
                                                                        15$-4$,-                            :
                                                                        24$-4$,-                            :
                                                                        24$-4$,-                            :
                                                                        15$-4$,-                            :
                                                                        27$-4$,-                            :
                                                                        27$-4$,-                            :
                                                                        27$-4$,-                            :
                                                                        27$-4$                              :
                                          00F6  31 00086 5$:    BRW     27$                                 : 2288
                          00    B7          68  90 00089 6$:    MOVB    (PTR), @0(R7)                       : 2028
                                            7C  11 0008D        BRB     17$                                 :
                          00    B7          68  D0 0008F 7$:    MOVL    (PTR), @0(R7)                       : 2044
                                67          04  C0 00093        ADDL2   #4, (R7)                            : 2045
                                            75  11 00096        BRB     18$                                 :
                                51          88  B0 00098 8$:    MOVW    (PTR)+, RNG_ELEMENTS                : 2057
                                52          51  3C 0009B        MOVZWL  RNG_ELEMENTS, R2                    : 2060
                                53      FF  A2  9E 0009E        MOVAB   -1(R2), R3                          : 2065
                                51          01  CE 000A2        MNEGL   #1, INDX                            :
                                            14  11 000A5        BRB     10$                                 :
                          00    B7          88  D0 000A7 9$:    MOVL    (PTR)+, @0(R7)                      : 2062
```

```
                              67          04  C0 000AB        ADDL2   #4, (R7)                         ; 2063
                              53          51  D1 000AE        CMPL    INDX, R3                         ; 2065
                                          08  18 000B1        BGEQ    10$                              ;
                      00      B7     01   A0  B0 000B3        MOVW    1(R0), @0(R7)                    ; 2069
                              67          02  C0 000B8        ADDL2   #2, (R7)                         ; 2070
    FFE6         51           02          52  F1 000BB 10$:   ACBL    R2, #2, INDX, 9$                 ; 2060
                                          60  11 000C1        BRB     20$                              ; 2028
                                          68  95 000C3 11$:   TSTB    (PTR)                            ; 2098
                                          2F  13 000C5        BEQL    14$                              ;
                      7F      8F          68  91 000C7        CMPB    (PTR), #127                      ; 2106
                                          3A  1A 000CB        BGTRU   16$                              ;
                              59          68  9A 000CD        MOVZBL  (PTR), R9                        ; 2112
                                          59  D6 000D0        INCL    R9                               ;
                 00      B7   68          59  28 000D2        MOVC3   R9, (PTR), @0(R7)                ; 2113
                              67          59  C0 000D7        ADDL2   R9, (R7)                         ; 2118
                                          47  11 000DA        BRB     20$                              ; 2097
                                          68  95 000DC 12$:   TSTB    (PTR)                            ; 2137
                                          2F  12 000DE        BNEQ    19$                              ;
                                        008E  31 000E0        BRW     24$                              ; 2140
                      00      B7          88  90 000E3 13$:   MOVB    (PTR)+, @0(R7)                   ; 2159
                                          67  D6 000E7        INCL    (R7)                             ;
                              57     08   AC  D0 000E9        MOVL    MSGPTR, R7                       ; 2164
                                     FF   A8  95 000ED        TSTB    -1(PTR)                          ; 2160
                                          04  13 000F0        BEQL    14$                              ;
                                          68  95 000F2        TSTB    (PTR)                            ; 2161
                                          0B  12 000F4        BNEQ    15$                              ;
  00      B7     18           00          68  F0 000F6 14$:   INSV    (PTR), #0, #24, @0(R7)           ; 2164
                              67          03  C0 000FC        ADDL2   #3, (R7)                         ; 2165
                                          77  11 000FF        BRB     25$                              ; 2160
                      7F      8F          68  91 00101 15$:   CMPB    (PTR), #127                      ; 2169
                                          08  1B 00105        BLEQU   19$                              ;
                      00      B7          88  90 00107 16$:   MOVB    (PTR)+, @0(R7)                   ; 2171
                                          67  D6 0010B 17$:   INCL    (R7)                             ;
                                          69  11 0010D 18$:   BRB     25$                              ;
                              50          68  9A 0010F 19$:   MOVZBL  (PTR), R0                        ; 2174
                                          50  D6 00112        INCL    R0                               ;
                 00      B7   68          50  28 00114        MOVC3   R0, (PTR), @0(R7)                ;
                              50          68  9A 00119        MOVZBL  (PTR), R0                        ; 2175
                              50          67  C0 0011C        ADDL2   (R7), R0                         ;
                              67     01   A0  9E 0011F        MOVAB   1(R0), (R7)                      ;
                                          53  11 00123 20$:   BRB     25$                              ; 2028
                      00      B7     0B   A8  90 00125 21$:   MOVB    11(PTR), @0(R7)                  ; 2219
                                          67  D6 0012A        INCL    (R7)                             ;
                      FF      8F     0B   A8  91 0012C        CMPB    11(PTR), #-1                     ; 2222
                                          29  13 00131        BEQL    23$                              ;
                              57     08   AC  D0 00133        MOVL    MSGPTR, R7                       ; 2235
                                     0B   A8  95 00137        TSTB    11(PTR)                          ; 2226
                                          11  12 0013A        BNEQ    22$                              ;
                                     0C   A8  95 0013C        TSTB    12(PTR)                          ; 2230
                                          0C  12 0013F        BNEQ    22$                              ;
  00      B7     18           00     0C   A8  F0 00141        INSV    12(PTR), #0, #24, @0(R7)         ; 2235
                              67          03  C0 00148        ADDL2   #3, (R7)                         ; 2239
                                          0F  11 0014B        BRB     23$                              ; 2237
                              59     0C   A8  9A 0014D 22$:   MOVZBL  12(PTR), R9                      ; 2244
                                          59  D6 00151        INCL    R9                               ;
                 00      B7   0C     A8   59  28 00153        MOVC3   R9, 12(PTR), @0(R7)              ; 2245
                              67          59  C0 00159        ADDL2   R9, (R7)                         ; 2249
```

```
                                   57         08   AC   D0 0015C 23$:      MOVL     MSGPTR, R7                              ; 2261
                         C0        8F         01   A8   93 00160           BITB     1(PTR), #192                            ; 2258
                                                   0A   12 00165           BNEQ     24$                                     ;
           00     B7               68              0B   28 00167           MOVC3    #11, (PTR), @0(R7)                      ; 2266
                                   67              0B   C0 0016C           ADDL2    #11, (R7)                               ; 2267
                                                   07   11 0016F           BRB      25$                                     ; 2258
                         00        B7              68   B0 00171 24$:      MOVW     (PTR), @0(R7)                           ; 2274
                                   67              02   C0 00175           ADDL2    #2, (R7)                                ; 2275
                                                   5A   D6 00178 25$:      INCL     PCTR                                    ; 2292
                                                   56   D6 0017A 26$:      INCL     IDX                                     ; 1996
                                                 FE90   31 0017C           BRW      1$                                      ;
                                   2B         0C   AC   E9 0017F 27$:      BLBC     CHKFLG, 30$                             ; 2298
                                   24  00000000G   00   E8 00183           BLBS     PDB$G_VRB_ALL, 30$                      ; 2299
                                   20         08   AB   E8 0018A           BLBS     NCP$GL_NOPARMS, 30$                     ; 2300
                                   50         04   AB   D0 0018E           MOVL     NCP$GL_QUALPRS, R0                      ; 2302
                                                   04   12 00192           BNEQ     28$                                     ;
                                                   5A   D5 00194           TSTL     PCTR                                    ; 2303
                                                   09   13 00196           BEQL     29$                                     ;
                                                   50   D5 00198 28$:      TSTL     R0                                      ; 2305
                                                   12   13 0019A           BEQL     30$                                     ;
                                   01              5A   D1 0019C           CMPL     PCTR, #1                                ; 2306
                                                   0D   14 0019F           BGTR     30$                                     ;
                                       00000000G   8F   DD 001A1 29$:      PUSHL    #NCP$_NOPARM                            ; 2308
                    00000000G      00              01   FB 001A7           CALLS    #1, LIB$STOP                            ;
                                                   04   04 001AE 30$:      RET                                              ; 2312

; Routine Size:  431 bytes,    Routine Base:  $CODE$ + 09A0
; 2350           2313  1 END                                 .End of module
; 2351           2314  0 ELUDOM


                                                                         .EXTRN  LIB$SIGNAL, LIB$STOP

;                                    PSECT SUMMARY
;
;        Name                      Bytes                          Attributes
;
;   $GLOBAL$                         1036  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                               4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                            336  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                           2895  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                             Library Statistics
;
;                                                -------- Symbols --------      Pages        Processing
;        File                                    Total     Loaded   Percent     Mapped       Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1             9776          8         0       581         00:01.0
;   _$255$DUA28:[NCP.OBJ]NCPLIBRY.L32;1            373         72        19        52         00:00.3
;   _$255$DUA28:[NCP.OBJ]NMALIBRY.L32;1            887         26         2        47         00:00.8



;                                    COMMAND QUALIFIERS

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

; Size:          2895 code + 1376 data bytes
; Run Time:         00:53.2
; Elapsed Time:     02:44.2
; Lines/CPU Min:      2608
; Lexemes/CPU-Min: 12198
; Memory Used:  320 pages
; Compilation Complete
```

0271 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

NCPSTANOD
LIS

NCPSTAVRB
LIS

NCPSTASHL
LIS

NCPSTAOBJ
LIS

NCPSTATRI
LIS

NCPSTAZER
LIS

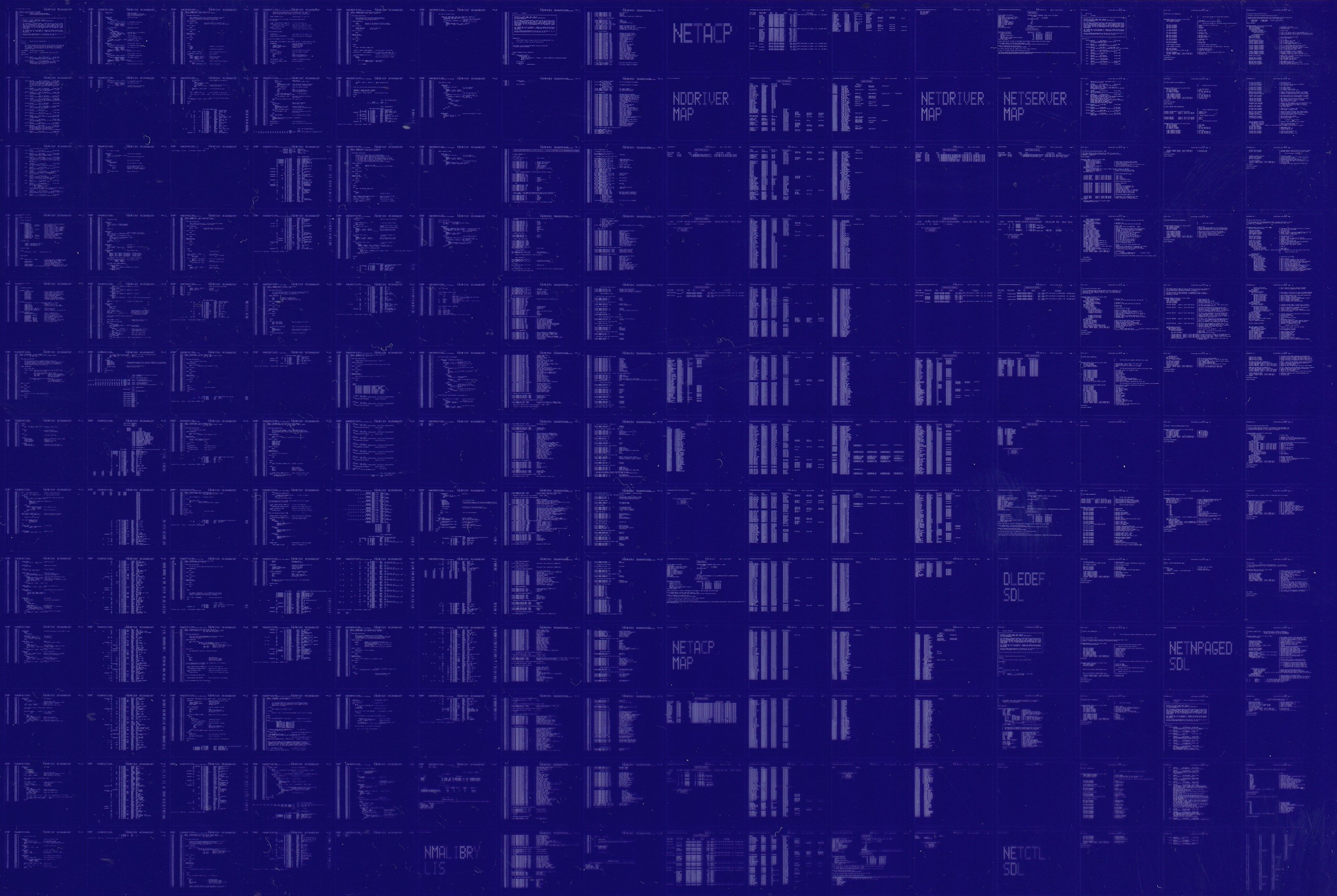
NCPTERMIO
LIS

NCPTABLES
LIS

NCPVRBACT
LIS

0272 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


NETACP

NDDRIVER
MAP

NETDRIVER
MAP

NETSERVER
MAP

DLEDEF
SDL

NETACP
MAP

NETNPAGED
SDL

NMALIBRY
LIS

NETCTL
SDL