I 3

```
   CCCCCCCCCCCC  LLL                  IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC  LLL                  IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC  LLL                  IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
CCC              LLL                     III        UUU         UUU        TTT        LLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL      IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL      IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL      IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**SHONET                                  C  8





  SSSSSSSS  HH      HH    000000    NN      NN  EEEEEEEEEE  TTTTTTTTTT
  SSSSSSSS  HH      HH    000000    NN      NN  EEEEEEEEEE  TTTTTTTTTT
SS          HH      HH  00      00  NN      NN  EE              TT
SS          HH      HH  00      00  NN      NN  EE              TT
SS          HH      HH  00      00  NNNN    NN  EE              TT
SS          HH      HH  00      00  NNNN    NN  EE              TT
  SSSSSS    HHHHHHHHHH  00      00  NN  NN  NN  EEEEEEEE        TT
  SSSSSS    HHHHHHHHHH  00      00  NN  NN  NN  EEEEEEEE        TT
        SS  HH      HH  00      00  NN    NNNN  EE              TT
        SS  HH      HH  00      00  NN    NNNN  EE              TT
        SS  HH      HH  00      00  NN      NN  EE              TT           ....
        SS  HH      HH  00      00  NN      NN  EE              TT           ....
SSSSSSSS    HH      HH    000000    NN      NN  EEEEEEEEEE      TT           ....
SSSSSSSS    HH      HH    000000    NN      NN  EEEEEEEEEE      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
```

```
0 MODULE show$network (IDENT = 'V04-000') =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 ! FACILITY:  SHOW Command
1 !
1 ! ABSTRACT:
1 !
1 !       This module processes the SHOW NETWORK command
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX/VMS operating system. unprivileged user mode,
1 !
1 ! AUTHOR:  Tim Halvorsen, August 1981
1 !
1 ! Modified by:
1 !
1 !       V03-010 TMH0010         Tim Halvorsen   27-Jun-1983
1 !               Make endnode display look better.
1 !
1 !       V03-009 TMH0009         Tim Halvorsen   17-May-1983
1 !               Fix bug in routine which obtains the next node name
1 !               in the area display.  It was accidentally sending
1 !               a binary count to the terminal.
1 !
1 !       V03-008 TMH0008         Tim Halvorsen   13-Mar-1983
1 !               Do not display loop nodes, and add new area display.
1 !
1 !       V03-007 GAS0105         Gerry Smith     20-Jan-1983
1 !               Fix output display.
1 !
1 !       V03-006 GAS0100         Gerry Smith     11-Jan-1983
```

```
1 !                     Remove reference to SHOW$L_STATUS, since all
1 !                     errors are signaled.
1 !
1 !     V03-005 GAS0099          Gerry Smith      7-Jan-1983
1 !             Minor modifications to fit new SHOW image.
1 !
1 !     V004    MKP0001          Kathy Perko      14-Dec-1982
1 !             Add capability to get multiple nodes in one QIO to NETACP.
1 !
1 !     V003    TMH0003          Tim Halvorsen    28-Nov-1982
1 !             Add formatting of area node addresses.
1 !
1 !     V002    TMH0002          Tim Halvorsen    24-Jun-1982
1 !             Fix failure to initialize an NFB field.
1 !
1 !     V001    TMH0001          Tim Halvorsen    03-Jun-1982
1 !             Modify to use new NETACP control QIO interface.
1 !--
1
1 !
1 ! Include files
1 !
1
1 LIBRARY 'SYS$LIBRARY:STARLET';               ! VAX/VMS common definitions
1
1 LIBRARY 'SHRLIB$:NET';                       ! NETACP control QIO definitions
1
1 REQUIRE 'SYS$LIBRARY:UTILDEF';               ! Common BLISS definitions
```

```
1 !
1 ! Table of contents
1 !
1
1 FORWARD ROUTINE
1     show$network:         NOVALUE,          ! Process SHOW NETWORK
1     display_nodes:        NOVALUE,          ! Produce reachable node display
1     format_area_info,                       ! Write area info to the display
1     format_node_info,                       ! Write node info to the display
1     get_node_name,                          ! Get node name given node address
1     write_line:           NOVALUE,          ! Write line to output
1     format_nodeadr;                         ! Format a node address
1
1 !
1 ! OWN storage
1 !
1
1 OWN
1     channel:    WORD;                       ! Channel to ACP
1
1 !
1 ! Status codes
1 !
1
1 EXTERNAL LITERAL
1     show$_nonet;                            ! Network not available
1
1 !
1 ! External routine
1 !
1
1 EXTERNAL ROUTINE
1     show$write_line:      NOVALUE;          ! General SHOW FAO output routine
```

```
  1 GLOBAL ROUTINE show$network : NOVALUE =
  1
  1 !---
  1 !
  1 !       This routine processes the SHOW NETWORK command
  1 !
  1 ! Inputs:
  1 !
  1 !       None
  1 !
  1 ! Outputs:
  1 !
  1 !       None
  1 !---
  1
  2 BEGIN
  2
  2 LITERAL
  2     buffer_size = 512;                  ! Size of return buffer.
  2
  2 LOCAL
  2     nfb:        BBLOCK [nfb$c_length+20*4],     ! Network function block
  2                                                 ! (room for 20 field requests)
  2     nfb_desc:   VECTOR [2],             ! Descriptor of NFB
  2     iosb:       BBLOCK [8],             ! I/O status block
  2     total_count,                        ! Number of entries displayed
  2     buffer_count,                       ! Number of entries returned in buffer
  2     buffer:     BBLOCK [buffer_size],   ! Return buffer
  2     buffer_desc: VECTOR [2],            ! Descriptor of above buffer
  2     buffer_ptr,                         ! Pointer to return buffer
  2     keys:       BBLOCK [4+8+nfb$c_ctx_size], ! Buffer for search keys & context
  2     key_desc:   VECTOR [2],             ! Descriptor of above buffer
  2     node_name_buffer: VECTOR [32,BYTE], ! Node name buffer
  2     node_name:  VECTOR [2],             ! Descriptor of above buffer
  2     exec_type,                          ! Executor node type
  2     exec_addr,                          ! Executor address
  2     exec_name_buffer: VECTOR [32,BYTE], ! Executor name buffer
  2     exec_name:  VECTOR [2],             ! Executor node name descriptor
  2     status;
  2
  2 !
  2 ! Assign a channel to the network ACP
  2 !
  2
P 2 status = $ASSIGN(CHAN=channel,          ! Assign channel to NETACP
  2                 DEVNAM=%ASCID '_NET:');
  2
  2 IF NOT .status                          ! If error detected,
  2 THEN
  3     BEGIN
  3     IF .status EQL ss$_nosuchdev        ! If network not yet up,
  3     THEN SIGNAL(show$_nonet)            ! then tell user
  3     ELSE SIGNAL(.status);               ! Else, report the status
  3     RETURN;
  2     END;
  2
  2 !
```

```
   2 ! Get our executor node name, address and type
   2 !
   2
   2 key_desc [0] = 4 + nfb$c_ctx_size;        ! Longword overhead, NO search values
   2 key_desc [1] = keys;                      ! and fixed context area
   2
   2 keys [0,0,32,0] = 0;                      ! Zero count of fields in P4 (unused)
   2 keys [4,0,16,0] = 0;                      ! Start key = at beginning
   2
   2 buffer_desc [0] = buffer_size;            ! Setup descriptor of P4 buffer
   2 buffer_desc [1] = buffer;
   2
   2 CH$FILL(0,nfb$c_length,nfb);              ! Pre-zero NFB fields
   2
   2 nfb [nfb$b_fct] = nfb$c_fc_show;          ! Request "show" function
   2 nfb [nfb$b_database] = nfb$c_db_lni;      ! of executor database
   2
   2 nfb_desc [0] = $BYTEOFFSET(nfb$l_fldid) + 3*4;  ! Construct descriptor of NFB
   2 nfb_desc [1] = nfb;
   2
   2 CH$MOVE(3*4, UPLIT LONG(                  ! Request the following fields:
   2                 nfb$c_lni_add,            ! Executor address
   2                 nfb$c_lni_ety,            ! Executor type
   2                 nfb$c_lni_nam),           ! Executor name
   2                 nfb [nfb$l_fldid]);
   2
P  2 status = $QIOW(FUNC = IO$_ACPCONTROL,     ! Issue control function
P  2                 CHAN = .channel,
P  2                 IOSB = iosb,
P  2                 P1 = nfb_desc,            ! Address of NDB descriptor
P  2                 P2 = key_desc,            ! Address of key buffer descriptor
   2                 P4 = buffer_desc);        ! Address of return buffer descriptor
   2
   2 IF NOT .status                            ! If error detected,
   3     OR NOT (status = .iosb [0,0,16,0])
   2 THEN
   3     BEGIN
   3     IF .status EQL ss$_devnotmount        ! If ACP not yet started,
   3     THEN SIGNAL(show$_nonet)              ! then indicate network not up
   3     ELSE SIGNAL(.status);                 ! Else, report the status
   3     RETURN;
   2     END;
   2
   2 exec_addr = .buffer [0,0,32,0];           ! Save our node address
   2 exec_type = .buffer [4,0,32,0];           ! Save our node type
   2 exec_name [0] = .buffer [8,0,16,0];       ! Construct descriptor of executor name
   2 exec_name [1] = exec_name_buffer;
   2 CH$MOVE(.exec_name [0], buffer+10, .exec_name [1]);
   2
   2 !
   2 ! Display title lines
   2 !
   2
   2 write_line(%ASCID 'VAX/VMS Network status for local node !AS !AS on !%D',
   2                 format_nodeadr(.exec_addr),
   2                 exec_name,
   2                 0);
```

```
  2 write_line(%ASCID '');
  2
  2 !
  2 ! If we are a level 2 (area) router, then display cost/hops information
  2 ! for all areas in the network.
  2 !
  2 ! If we are a level 1 router, then the area database will display the
  2 ! "nearest level 2 router".
  2 !
  2
  2 buffer_desc [0] = buffer_size;          ! Construct descriptor of return buffer
  2 buffer_desc [1] = buffer;
  2
  2 key_desc [0] = 4 + 4 + nfb$c_ctx_size;  ! Longword overhead, ONE search value
  2 key_desc [1] = keys;                    ! and fixed context area
  2
  2 keys [0,0,32,0] = 0;                    ! Zero count of fields in P4 (unused)
  2 keys [4,0,32,0] = true;                 ! REA search value EQL TRUE
  2 keys [8,0,16,0] = 0;                    ! Start key = at beginning
  2
  2 CH$FILL(0,nfb$c_length,nfb);            ! Pre-zero NFB fields
  2
  2 nfb [nfb$b_fct] = nfb$c_fc_show;        ! Request "show" function
  2 nfb [nfb$b_database] = nfb$c_db_ari;    ! of area database
  2 nfb [nfb$b_flags] = nfb$m_mult;         ! Request multiple entries per QIO
  2 nfb [nfb$l_srch_key] = nfb$c_ari_rea;   ! Only return reachable areas
  2 nfb [nfb$b_oper] = nfb$c_op_eql;        ! by checking if field EQL P2 value
  2
  2 nfb_desc [0] = $BYTEOFFSET(nfb$l_fldid) + 5*4;  ! Construct descriptor of NFB
  2 nfb_desc [1] = nfb;
  2
  2 CH$MOVE(5*4, UPLIT LONG(                ! Request the following fields:
  2                 nfb$c_ari_add,          ! Area number
  2                 nfb$c_ari_dco,          ! Destination cost
  2                 nfb$c_ari_dho,          ! Destination hops
  2                 nfb$c_ari_nnd,          ! Next node to destination
  2                 nfb$c_ari_dli),         ! Destination circuit name
  2                 nfb [nfb$l_fldid]);
  2
  2 total_count = 0;                        ! Initialize area count
  2
  2 WHILE true
  2 DO
  3     BEGIN
P 3     status = $QIOW(FUNC = IO$_ACPCONTROL,       ! Issue control function
P 3                 CHAN = .channel,
P 3                 IOSB = iosb,
P 3                 P1 = nfb_desc,          ! Address of NDB descriptor
P 3                 P2 = key_desc,          ! Address of key buffer descriptor
  3                 P4 = buffer_desc);      ! Address of return buffer descriptor
  3
  3     IF NOT .status                      ! If error detected,
  4         OR NOT (status = .iosb [0,0,16,0])
  3     THEN
  3         EXITLOOP;                       ! then stop looping
  3
  3     IF .exec_type NEQ adj$c_pty_area    ! If we are not an area router,
3          THEN
4              BEGIN
4              BIND
4                  next_hop_addr = buffer [3*4,0,32,0];
4
4              node_name [0] = 32;                     ! Make descriptor of output buffer
4              node_name [1] = node_name_buffer;
4              node_name [0] =                         ! Get node name of next hop
4                  get_node_name(.next_hop_addr, node_name);
4
4              SELECTONEU .exec_type OF
4                  SET
4                  [adj$c_pty_ph4n,adj$c_pty_ph3n]:
5                      BEGIN
5                      write_line(%ASCID 'This is a nonrouting node, and does not have any network information.');
5                      IF .next_hop_addr NEQ -1
5                      THEN
5                          write_line(%ASCID 'The designated router for !AS is node !AS !AS.',
5                              exec_name,
5                              format_nodeadr(.next_hop_addr),
5                              node_name);
4                      END;
4                  [OTHERWISE]:
5                      BEGIN
5                      IF .next_hop_addr NEQ -1
5                      THEN
5                          write_line(%ASCID 'The next hop to the nearest area router is node !AS !AS.',
5                              format_nodeadr(.next_hop_addr),
5                              node_name);
4                      END;
4                  TES;
4              total_count = 1;                        ! Force some spacing afterwards
4              EXITLOOP;                               ! Do not display area database
3              END;
3
3          IF .total_count EQL 0                       ! If first time through,
3          THEN                                        ! Print header line
3              write_line(%ASCID '!/!13* Area   Cost  Hops    Next Hop to Area!/');
3
3          buffer_ptr = buffer;                        ! Point to first node in buffer.
3          buffer_count = .keys [0,0,32,0];            ! Get number of nodes returned in the
3                                                      !       buffer.
3          WHILE .buffer_count GTR 0
3          DO
4              BEGIN
4              buffer_ptr = format_area_info (.buffer_ptr);
4              total_count = .total_count + 1;  ! Increment # areas reachable
4              buffer_count = .buffer_count - 1;
3              END;
2          END;
2
2      !
2      ! As long as we aren't an endnode, display reachable nodes
2      !
2
2      IF .exec_type NEQ adj$c_pty_ph4n          ! If we aren't an endnode,
2          AND .exec_type NEQ adj$c_pty_ph3n
```

```
2 THEN
3     BEGIN
3     IF .total_count GTR 0                          ! If displayed at least 1 area,
3     THEN
3         write_line(%ASCID '');                     ! put 1 blank line here
3
3     display_nodes();                               ! Display reachable nodes in our area
2     END;
2
2 !
2 ! Cleanup channel to ACP
2 !
2
2 $DASSGN(CHAN = .channel);                          ! Deassign the ACP channel
2
2 RETURN;                                            ! Return to CLI dispatcher
2
1 END;

                                                                          .TITLE  SHOW$NETWORK
                                                                          .IDENT  \V04-000\

                                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

                        00  00  00  3A  54  45  4E  5F  00000 P.AAB:   .ASCII  \_NET:\<0><0><0>
                                              010E0005  00008 P.AAA:   .LONG   17694725
                                             00000000'  0000C          .ADDRESS P.AAB
                        01020041  0101001A  01010010  00010 P.AAC:   .LONG   16842768, 16842778, 16908353
6B  72  6F  77  74  65  4E  20  53  4D  56  2F  58  41  56  0001C P.AAE:   .ASCII  \VAX/VMS Network status for local node !A\
63  6F  6C  20  72  6F  66  20  73  75  74  61  74  73  20  0002B
                        41  21  20  65  64  6F  6E  20  6C  61  0003A
            44  25  21  20  6E  6F  20  53  41  21  20  53  00044          .ASCII  \S !AS on !%D\
                                              010E0034  00050 P.AAD:   .LONG   17694772
                                             00000000'  00054          .ADDRESS P.AAE
                                                        00058 P.AAG:   .BLKB   0
                                              010E0000  00058 P.AAF:   .LONG   17694720
                                             00000000'  0005C          .ADDRESS P.AAG
          14020041  14010013  14010012  14010011  14010010  00060 P.AAH:   .LONG   335609872, 335609873, 335609874, -
                                                                                  335609875, 335675457
6F  72  6E  6F  6E  20  61  20  73  69  20  73  69  68  54  00074 P.AAJ:   .ASCII  \This is a nonrouting node, and does not \
64  6E  61  20  2C  65  64  6F  6E  20  67  6E  69  74  75  00083
                    20  74  6F  6E  20  73  65  6F  64  20  00092
72  6F  77  74  65  6E  20  79  6E  61  20  65  76  61  68  0009C          .ASCII  \have any network information.\<0><0>
00  2E  6E  6F  69  74  61  6D  72  6F  66  6E  69  20  6B  000AB
                                                        00  000BA
                                                        00  000BB          .ASCII  <0>
                                              010E0045  000BC P.AAI:   .LONG   17694789
                                             00000000'  000C0          .ADDRESS P.AAJ
20  64  65  74  61  6E  67  69  73  65  64  20  65  68  54  000C4 P.AAL:   .ASCII  \The designated router for !AS is node !A\
20  53  41  21  20  72  6F  66  20  72  65  74  75  6F  72  000D3
                    41  21  20  65  64  6F  6E  20  73  69  000E2
                            00  00  2E  53  41  21  20  53  000EC          .ASCII  \S !AS.\<0><0>
                                              010E002E  000F4 P.AAK:   .LONG   17694766
                                             00000000'  000F8          .ADDRESS P.AAL
6F  74  20  70  6F  68  20  74  78  65  6E  20  65  68  54  000FC P.AAN:   .ASCII  \The next hop to the nearest area router \
72  61  20  74  73  65  72  61  65  6E  20  65  68  74  20  0010B
```

```
                                                                   L  8
SHOW$NETWORK                                                      16-Sep-1984 00:39:09     VAX-11 Bliss-32 V4.0-742             Page   9
V04-000                                                           14-Sep-1984 12:09:32     [CLIUTL.SRC]SHONET.B32;1                  (3)

                    20  72  65  74  75  6F  72  20  61  65    0011A                                                               ;
53  41  21  20  53  41  21  20  65  64  6F  6E  20  73  69    00124         .ASCII  \is node !AS !AS.\                            ;
                                                        2E    00133                                                               ;
                                                  010E0038.   00134 P.AAM:  .LONG   17694776                                      ;
                                                  00000000'   00138         .ADDRESS P.AAN                                        ;
43  20  20  20  61  65  72  41  20  2A  33  31  21  2F  21    0013C P.AAP:  .ASCII  \!/!13* Area   Cost  Hops    Next Hop to \    ;
65  4E  20  20  20  20  73  70  6F  48  20  20  74  73  6F    0014B                                                               ;
                    20  6F  74  20  70  6F  48  20  74  78    0015A                                                               ;
                            00  00  2F  21  61  65  72  41    00164         .ASCII  \Area!/\<0><0>                                ;
                                                  010E002E.   0016C P.AAO:  .LONG   17694766                                      ;
                                                  00000000'   00170         .ADDRESS P.AAP                                        ;
                                                              00174 P.AAR:  .BLKB   0                                             ;
                                                  010E0000.   00174 P.AAQ:  .LONG   17694720                                      ;
                                                  00000000'   00178         .ADDRESS P.AAR                                        ;

                                                                            .PSECT  $OWN$,NOEXE,2

                                                              00000 CHANNEL:.BLKB   2

                                                                            .EXTRN  SHOW$_NONET, SHOW$WRITE_LINE
                                                                            .EXTRN  SYS$ASSIGN, SYS$QIOW
                                                                            .EXTRN  SYS$DASSGN

                                                                            .PSECT  $CODE$,NOWRT,2

                                                         0FFC 00000         .ENTRY  SHOW$NETWORK, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0294
                                                                                    R10,R11                                       ;
                                        5B 00000000G   00 9E  00002         MOVAB   SYS$QIOW, R11                                 ;
                                        5A     0000V   CF 9E  00009         MOVAB   WRITE_LINE, R10                               ;
                                        59     0000'   CF 9E  0000E         MOVAB   P.AAA, R9                                     ;
                                        5E      FCE4   CE 9E  00013         MOVAB   -796(SP), SP                                  ;
                                                       7E 7C  00018         CLRQ    -(SP)                                         ; 0339
                                               0000'   CF 9F  0001A         PUSHAB  CHANNEL                                       ;
                                                       59 DD  0001E         PUSHL   R9                                            ;
                             00000000G  00             04 FB  00020         CALLS   #4, SYS$ASSIGN                                ;
                                        56             50 D0  00027         MOVL    R0, STATUS                                    ;
                                        11             56 E8  0002A         BLBS    STATUS, 2$                                    ; 0341
                             00000908   8F             56 D1  0002D         CMPL    STATUS, #2312                                 ; 0344
                                                       79 12  00034 1$:     BNEQ    4$                                            ;
                                           00000000G   8F DD  00036         PUSHL   #SHOW$_NONET                                  ; 0345
                                                       73 11  0003C         BRB     5$                                            ; 0346
                                    50  AE        44   8F 9A  0003E 2$:     MOVZBL  #68, KEY_DESC                                 ; 0354
                                    54  AE        58   AE 9E  00043         MOVAB   KEYS, KEY_DESC+4                              ; 0355
                                                  58   AE D4  00048         CLRL    KEYS                                          ; 0357
                                                  5C   AE B4  0004B         CLRW    KEYS+4                                        ; 0358
                                  00A4  CE      0200   8F 3C  0004E         MOVZWL  #512, BUFFER_DESC                             ; 0360
                                  00A8  CE      00AC   CE 9E  00055         MOVAB   BUFFER, BUFFER_DESC+4                         ; 0361
            10            00            6E             00 2C  0005C         MOVC5   #0, (SP), #0, #16, NFB                        ; 0363
                                                  A0   AD     00061                                                               ;
                                    A0  AD             22 90  00063         MOVB    #34, NFB                                      ; 0365
                                    A2  AD             01 90  00067         MOVB    #1, NFB+2                                     ; 0366
                                    98  AD             1C D0  0006B         MOVL    #28, NFB_DESC                                 ; 0368
                                    9C  AD        A0   AD 9E  0006F         MOVAB   NFB, NFB_DESC+4                               ; 0369
                      B0  AD        08  A9             0C 28  00074         MOVC3   #12, P.AAC, NFB+16                            ; 0375
                                                       7E 7C  0007A         CLRQ    -(SP)                                         ; 0382
                                                00AC   CE 9F  0007C         PUSHAB  BUFFER_DESC                                   ;
                                                       7E D4  00080         CLRL    -(SP)                                         ;
```

```
                                                            M  8
SHOW$NETWORK                                          16-Sep-1984 00:39:09    VAX-11 Bliss-32 V4.0-742          Page  10
V04-000                                               14-Sep-1984 12:09:32    [CLIUTL.SRC]SHONET.B32;1                (3)

                                        60  AE  9F 00082        PUSHAB  KEY_DESC                            ;
                                        98  AD  9F 00085        PUSHAB  NFB_DESC                            ;
                                            7E  7C 00088        CLRQ    -(SP)                               ;
                                        90  AD  9F 0008A        PUSHAB  IOSB                                ;
                                            38  DD 0008D        PUSHL   #56                                 ;
                              7E     0000'  CF  3C 0008F        MOVZWL  CHANNEL, -(SP)                      ;
                                            7E  D4 00094        CLRL    -(SP)                               ;
                              6B            0C  FB 00096        CALLS   #12, SYS$QIOW                       ;
                              56            50  D0 00099        MOVL    R0, STATUS                          ;
                              07            56  E9 0009C        BLBC    STATUS, 3$                          ; 0384
                              56        90  AD  3C 0009F        MOVZWL  IOSB, STATUS                        ; 0385
                              13            56  E8 000A3        BLBS    STATUS, 6$                          ;
                   0000007C   8F            56  D1 000A6  3$:   CMPL    STATUS, #124                        ; 0388
                                            85  11 000AD        BRB     1$                                  ;
                                            56  DD 000AF  4$:   PUSHL   STATUS                              ; 0390
                  00000000G   00            01  FB 000B1  5$:   CALLS   #1, LIB$SIGNAL                      ;
                                                04 000B8        RET                                         ; 0387
                              57      00AC  CE  7D 000B9  6$:   MOVQ    BUFFER, EXEC_ADDR                   ; 0394
                              6E      00B4  CE  3C 000BE        MOVZWL  BUFFER+8, EXEC_NAME                 ; 0396
                         04   AE        08  AE  9E 000C3        MOVAB   EXEC_NAME_BUFFER, EXEC_NAME+4       ; 0397
        04    BE       00B6   CE            6E  28 000C8        MOVC3   EXEC_NAME, BUFFER+10, @EXEC_NAME+4  ; 0398
                                            7E  D4 000CF        CLRL    -(SP)                               ; 0404
                                        04  AE  9F 000D1        PUSHAB  EXEC_NAME                           ;
                                            57  DD 000D4        PUSHL   EXEC_ADDR                           ; 0405
                      0000V   CF            01  FB 000D6        CALLS   #1, FORMAT_NODEADR                  ;
                                            50  DD 000DB        PUSHL   R0                                  ;
                                        48  A9  9F 000DD        PUSHAB  P.AAD                               ; 0404
                              6A            04  FB 000E0        CALLS   #4, WRITE_LINE                      ;
                                        50  A9  9F 000E3        PUSHAB  P.AAF                               ; 0408
                              6A            01  FB 000E6        CALLS   #1, WRITE_LINE                      ;
                       00A4   CE      0200  8F  3C 000E9        MOVZWL  #512, BUFFER_DESC                   ; 0418
                       00A8   CE      00AC  CE  9E 000F0        MOVAB   BUFFER, BUFFER_DESC+4               ; 0419
                         50   AE        48  8F  9A 000F7        MOVZBL  #72, KEY_DESC                       ; 0421
                         54   AE        58  AE  9E 000FC        MOVAB   KEYS, KEY_DESC+4                    ; 0422
                                        58  AE  D4 00101        CLRL    KEYS                                ; 0424
                         5C   AE            01  D0 00104        MOVL    #1, KEYS+4                          ; 0425
                                        60  AE  B4 00108        CLRW    KEYS+8                              ; 0426
10            00              6E            00  2C 0010B        MOVC5   #0, (SP), #0, #16, NFB              ; 0428
                                        A0  AD     00110                                                    ;
                         A2   AD            14  90 00112        MOVB    #20, NFB+2                          ; 0431
                         A0   AD      0222  8F  B0 00116        MOVW    #546, NFB                           ; 0430
                         A4   AD  14000002  8F  D0 0011C        MOVL    #335544322, NFB+4                   ; 0433
                                        A3  AD  94 00124        CLRB    NFB+3                               ; 0434
                         98   AD            24  D0 00127        MOVL    #36, NFB_DESC                       ; 0436
                         9C   AD        A0  AD  9E 0012B        MOVAB   NFB, NFB_DESC+4                     ; 0437
        B0    AD         58   A9            14  28 00130        MOVC3   #20, P.AAH, NFB+16                  ; 0445
                                            53  D4 00136        CLRL    TOTAL_COUNT                         ; 0447
                                            7E  7C 00138  7$:   CLRQ    -(SP)                               ; 0457
                                      00AC  CE  9F 0013A        PUSHAB  BUFFER_DESC                         ;
                                            7E  D4 0013E        CLRL    -(SP)                               ;
                                        60  AE  9F 00140        PUSHAB  KEY_DESC                            ;
                                        98  AD  9F 00143        PUSHAB  NFB_DESC                            ;
                                            7E  7C 00146        CLRQ    -(SP)                               ;
                                        90  AD  9F 00148        PUSHAB  IOSB                                ;
                                            38  DD 0014B        PUSHL   #56                                 ;
                              7E     0000'  CF  3C 0014D        MOVZWL  CHANNEL, -(SP)                      ;
                                            7E  D4 00152        CLRL    -(SP)                               ;
```

```
                        6B           0C  FB 00154          CALLS   #12, SYS$QIOW                                  ;
                        56           50  D0 00157          MOVL    R0, STATUS                                     ;
                        79           56  E9 0015A          BLBC    STATUS, 11$                                    ; 0459
                        56      90   AD  3C 0015D          MOVZWL  IOSB, STATUS                                   ; 0460
                        72           56  E9 00161          BLBC    STATUS, 11$                                    ;
                        03           58  D1 00164          CMPL    EXEC_TYPE, #3                                  ; 0464
                                     6F  13 00167          BEQL    12$                                            ;
                 28     AE           20  D0 00169          MOVL    #32, NODE_NAME                                 ; 0470
                 2C     AE      30   AE  9E 0016D          MOVAB   NODE_NAME_BUFFER, NODE_NAME+4                  ; 0471
                                28   AE  9F 00172          PUSHAB  NODE_NAME                                      ; 0473
                        52    00BC   CE  D0 00175          MOVL    NEXT_HOP_ADDR, R2                              ;
                                     52  DD 0017A          PUSHL   R2                                             ;
              0000V     CF           02  FB 0017C          CALLS   #2, GET_NODE_NAME                              ;
                 28     AE           50  D0 00181          MOVL    R0, NODE_NAME                                  ;
                        01           58  D1 00185          CMPL    EXEC_TYPE, #1                                  ; 0477
                                     05  13 00188          BEQL    8$                                             ;
                        05           58  D1 0018A          CMPL    EXEC_TYPE, #5                                  ;
                                     28  12 0018D          BNEQ    9$                                             ;
                              00B4   C9  9F 0018F  8$:     PUSHAB  P.AAI                                          ; 0479
                        6A           01  FB 00193          CALLS   #1, WRITE_LINE                                 ;
           FFFFFFFF     8F           52  D1 00196          CMPL    R2, #-1                                        ; 0480
                                     34  13 0019D          BEQL    10$                                            ;
                                28   AE  9F 0019F          PUSHAB  NODE_NAME                                      ; 0482
                                     52  DD 001A2          PUSHL   R2                                             ; 0484
              0000V     CF           01  FB 001A4          CALLS   #1, FORMAT_NODEADR                             ;
                                     50  DD 001A9          PUSHL   R0                                             ;
                                08   AE  9F 001AB          PUSHAB  EXEC_NAME                                      ; 0482
                              00EC   C9  9F 001AE          PUSHAB  P.AAR                                          ;
                        6A           04  FB 001B2          CALLS   #4, WRITE_LINE                                 ;
                                     1C  11 001B5          BRB     10$                                            ; 0475
           FFFFFFFF     8F           52  D1 001B7  9$:     CMPL    R2, #-1                                        ; 0489
                                     13  13 001BE          BEQL    10$                                            ;
                                28   AE  9F 001C0          PUSHAB  NODE_NAME                                      ; 0491
                                     52  DD 001C3          PUSHL   R2                                             ; 0492
              0000V     CF           01  FB 001C5          CALLS   #1, FORMAT_NODEADR                             ;
                                     50  DD 001CA          PUSHL   R0                                             ;
                              012C   C9  9F 001CC          PUSHAB  P.AAM                                          ; 0491
                        6A           03  FB 001D0          CALLS   #3, WRITE_LINE                                 ;
                        53           01  D0 001D3  10$:    MOVL    #1, TOTAL_COUNT                                ; 0496
                                     29  11 001D6  11$:    BRB     16$                                            ; 0466
                                     53  D5 001D8  12$:    TSTL    TOTAL_COUNT                                    ; 0500
                                     07  12 001DA          BNEQ    13$                                            ;
                              0164   C9  9F 001DC          PUSHAB  P.AAO                                          ; 0502
                        6A           01  FB 001E0          CALLS   #1, WRITE_LINE                                 ;
                        55    00AC   CE  9E 001E3  13$:    MOVAB   BUFFER, BUFFER_PTR                             ; 0504
                        54      58   AE  D0 001E8          MOVL    KEYS, BUFFER_COUNT                             ; 0505
                                     03  14 001EC  14$:    BGTR    15$                                            ; 0507
                                   FF47  31 001EE          BRW     7$                                             ;
                                     55  DD 001F1  15$:    PUSHL   BUFFER_PTR                                     ; 0510
              0000V     CF           01  FB 001F3          CALLS   #1, FORMAT_AREA_INFO                           ;
                        55           50  D0 001F8          MOVL    R0, BUFFER_PTR                                 ;
                                     53  D6 001FB          INCL    TOTAL_COUNT                                    ; 0511
                                     54  D7 001FD          DECL    BUFFER_COUNT                                   ; 0512
                                     EB  11 001FF          BRB     14$                                            ; 0507
                        05           58  D1 00201  16$:    CMPL    EXEC_TYPE, #5                                  ; 0520
                                     15  13 00204          BEQL    18$                                            ;
                        01           58  D1 00206          CMPL    EXEC_TYPE, #1                                  ; 0521
```

```
                                  10   13 00209            BEQL    18$                                 ;
                                  53   D5 0020B            TSTL    TOTAL_COUNT                         ; 0524
                                  07   15 0020D            BLEQ    17$                                 ;
                          016C    C9   9F 0020F            PUSHAB  P.AAQ                               ; 0526
                  6A              01   FB 00213            CALLS   #1, WRITE_LINE                      ;
           0000V  CF              00   FB 00216 17$:       CALLS   #0, DISPLAY_NODES                   ; 0528
                  7E      0000'   CF   3C 0021B 18$:       MOVZWL  CHANNEL, -(SP)                      ; 0535
       00000000G  00              01   FB 00220            CALLS   #1, SYS$DASSGN                      ;
                                       04 00227            RET                                         ; 0539

; Routine Size:  552 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 ROUTINE display_nodes: NOVALUE =
1
1 !---
1 !
1 !       This routine displays all reachable nodes in our area.
1 !
1 ! Inputs:
1 !
1 !       None
1 !
1 ! Outputs:
1 !
1 !       None
1 !---
1
2 BEGIN
2
2 LITERAL
2     buffer_size = 512;                      ! Size of return buffer.
2
2 LOCAL
2     nfb:        BBLOCK [nfb$c_length+20*4],     ! Network function block
2                                                 ! (room for 20 field requests)
2     nfb_desc:   VECTOR [2],                 ! Descriptor of NFB
2     iosb:       BBLOCK [8],                 ! I/O status block
2     total_node_count,                       ! Number of nodes displayed
2     buffer_node_count,                      ! Number of nodes returned in buffer
2     buffer:     BBLOCK [buffer_size],       ! Return buffer
2     buffer_desc: VECTOR [2],                ! Descriptor of above buffer
2     buffer_ptr,                             ! Pointer to return buffer
2     keys:       BBLOCK [4+8+nfb$c_ctx_size], ! Buffer for search keys & context
2     key_desc:   VECTOR [2],                 ! Descriptor of above buffer
2     status;
2
2 !
2 ! Display the cost/hops information for all nodes in this area
2 !
2
2 buffer_desc [0] = buffer_size;              ! Construct descriptor of return buffer
2 buffer_desc [1] = buffer;
2
2 key_desc [0] = 4 + 8 + nfb$c_ctx_size;      ! Longword overhead, TWO search values
2 key_desc [1] = keys;                        ! and fixed context area
2
2 keys [0,0,32,0] = 0;                        ! Zero count of fields in P4 (unused)
2 keys [4,0,32,0] = true;                     ! REA search value EQL TRUE
2 keys [8,0,32,0] = true;                     ! LOO search value NEQ true
2 keys [12,0,16,0] = 0;                       ! Start key = at beginning
2
2 CH$FILL(0,nfb$c_length,nfb);                ! Pre-zero NFB fields
2
2 nfb [nfb$b_fct] = nfb$c_fc_show;            ! Request "show" function
2 nfb [nfb$b_database] = nfb$c_db_ndi;        ! of node database
2 nfb [nfb$b_flags] = nfb$m_mult;             ! Request multiple entries per QIO
2 nfb [nfb$l_srch_key] = nfb$c_ndi_rea;       ! Only return reachable nodes
2 nfb [nfb$b_oper] = nfb$c_op_eql;            ! by checking if field EQL P2 value
2 nfb [nfb$l_srch2_key] = nfb$c_ndi_loo;      ! Do not return "loop nodes"
```

```
nfb [nfb$b_oper2] = nfb$c_op_neq;            ! by checking if field NEQ P2 value

nfb_desc [0] = $BYTEOFFSET(nfb$l_fldid) + 8*4;  ! Construct descriptor of NFB
nfb_desc [1] = nfb;

CH$MOVE(8*4, UPLIT LONG(                     ! Request the following fields:
                nfb$c_ndi_tad,               ! Translated node address
                nfb$c_ndi_acl,               ! Active links
                nfb$c_ndi_dco,               ! Destination cost
                nfb$c_ndi_dho,               ! Destination hops
                nfb$c_ndi_nnd,               ! Next hop node address
                nfb$c_ndi_nna,               ! Node name
                nfb$c_ndi_nnn,               ! Next hop node name
                nfb$c_ndi_dli),              ! Destination circuit name
                nfb [nfb$l_fldid]);

total_node_count = 0;                        ! Initialize node count

WHILE true
DO
    BEGIN
    status = $QIOW(FUNC = IO$_ACPCONTROL,        ! Issue control function
                CHAN = .channel,
                IOSB = iosb,
                P1 = nfb_desc,               ! Address of NDB descriptor
                P2 = key_desc,               ! Address of key buffer descriptor
                P4 = buffer_desc);           ! Address of return buffer descriptor

    IF NOT .status                           ! If error detected,
        OR NOT (status = .iosb [0,0,16,0])
    THEN
        EXITLOOP;                            ! then stop looping

    IF .total_node_count EQL 0               ! If first time through,
    THEN                                     ! Print header line
        write_line(%ASCID '!/!8* Node!9*  Links  Cost  Hops   Next Hop to Node!/');

    buffer_ptr = buffer;                     ! Point to first node in buffer.
    buffer_node_count = .keys [0,0,32,0];    ! Get number of nodes returned in the
                                             !         buffer.
    WHILE .buffer_node_count GTR 0
    DO
        BEGIN
        buffer_ptr = format_node_info (.buffer_ptr);
        total_node_count =                   ! Increment # nodes reachable
                .total_node_count + 1;
        buffer_node_count = .buffer_node_count - 1;
        END;

    END;

IF .status EQL ss$_endoffile                 ! If normal termination,
THEN
    BEGIN
    IF .total_node_count GTR 1               ! If more than local node found,
    THEN                                     ! Write the total
        write_line(%ASCID '!/!16* Total of !UL node!%S.',
;   482              0654  3                          .total_node_count);
;   483              0655  3              END
;   484              0656  2      ELSE
;   485              0657  3              BEGIN
;   486              0658  3              IF .status EQL ss$_devnotmount          ! If ACP not yet started,
;   487              0659  3              THEN SIGNAL(show$_nonet)                ! then indicate network not up
;   488              0660  3              ELSE SIGNAL(.status);                   ! Else, report the status
;   489              0661  3              END;
;   490              0662  2
;   491              0663  1      END;


                                                                             .PSECT  $PLIT$,NOWRT,NOEXE,2

02020043  02010022  02010018  02010017  02010014  02010010  0017C P.AAS:  .LONG   33619984, 33619988, 33619991, 33619992, -    ;
                                        0202004D  02020059  00194                 33620002, 33685571, 33685593, 33685581       ;
20  20  2A  39  21  65  64  6F  4E  20  2A  38  21  2F  21  0019C P.AAU:  .ASCII  \!/!8* Node!9*  Links  Cost  Hops    Next \   ;
6F  48  20  20  74  73  6F  43  20  20  73  6B  6E  69  4C  001AB                                                              ;
                    20  74  78  65  4E  20  20  20  73  70  001BA                                                              ;
00  00  2F  21  65  64  6F  4E  20  6F  74  20  70  6F  48  001C4         .ASCII  \Hop to Node!/\<0><0><0>                     ;
                                                        00  001D3                                                              ;
                                                  010E0035  001D4 P.AAT:  .LONG   17694773                                     ;
                                                 00000000'  001D8         .ADDRESS P.AAU                                       ;
66  6F  20  6C  61  74  6F  54  20  2A  36  31  21  2F  21  001DC P.AAW:  .ASCII  \!/!16* Total of !UL node!%S.\               ;
        2E  53  25  21  65  64  6F  6E  20  4C  55  21  20  001EB                                                              ;
                                                  010E001C  001F8 P.AAV:  .LONG   17694748                                     ;
                                                 00000000'  001FC         .ADDRESS P.AAW                                       ;


                                                                             .PSECT  $CODE$,NOWRT,2

                                               003C 00000 DISPLAY_NODES:
                                                                          .WORD   Save R2,R3,R4,R5                             ; 0540
                         5E        FD38    CE   9E 00002                  MOVAB   -712(SP), SP                                 ;
                 50      AE        0200    8F   3C 00007                  MOVZWL  #512, BUFFER_DESC                            ; 0578
                 54      AE          58    AE   9E 0000D                  MOVAB   BUFFER, BUFFER_DESC+4                        ; 0579
                         7E          4C    8F   9A 00012                  MOVZBL  #76, KEY_DESC                                ; 0581
                 04      AE          08    AE   9E 00016                  MOVAB   KEYS, KEY_DESC+4                             ; 0582
                                     08    AE   D4 0001B                  CLRL    KEYS                                         ; 0584
                 0C      AE                01   D0 0001E                  MOVL    #1, KEYS+4                                   ; 0585
                 10      AE                01   D0 00022                  MOVL    #1, KEYS+8                                   ; 0586
                                     14    AE   B4 00026                  CLRW    KEYS+12                                      ; 0587
     10            00                6E    00   2C 00029                  MOVC5   #0, (SP), #0, #16, NFB                       ; 0589
                                     A0    AD      0002E                                                               ;
                 A2      AD                02   90 00030                  MOVB    #2, NFB+2                                    ; 0592
                 A0      AD        0222    8F   B0 00034                  MOVW    #546, NFB                                    ; 0591
                 A4      AD    02000003    8F   D0 0003A                  MOVL    #33554435, NFB+4                             ; 0594
                                     A3    AD   94 00042                  CLRB    NFB+3                                        ; 0595
                 A8      AD    02000002    8F   D0 00045                  MOVL    #33554434, NFB+8                             ; 0596
                 AC      AD                03   90 0004D                  MOVB    #3, NFB+12                                   ; 0597
                 98      AD                30   D0 00051                  MOVL    #48, NFB_DESC                                ; 0599
                 9C      AD          A0    AD   9E 00055                  MOVAB   NFB, NFB_DESC+4                              ; 0600
          B0      AD      0000'    CF      20   28 0005A                  MOVC3   #32, P.AAS, NFB+16                           ; 0611
                                           53   D4 00061                  CLRL    TOTAL_NODE_COUNT                             ; 0613
                                           7E   7C 00063 1$:              CLRQ    -(SP)                                        ; 0623
```

```
                                  5C   AE  9F 00065        PUSHAB  BUFFER_DESC                                 ;
                                       7E  D4 00068        CLRL    -(SP)                                       ;
                                  10   AE  9F 0006A        PUSHAB  KEY_DESC                                    ;
                                  98   AD  9F 0006D        PUSHAB  NFB_DESC                                    ;
                                       7E  7C 00070        CLRQ    -(SP)                                       ;
                                  90   AD  9F 00072        PUSHAB  IOSB                                        ;
                                       38  DD 00075        PUSHL   #56                                         ;
                      7E       0000'   CF  3C 00077        MOVZWL  CHANNEL, -(SP)                              ;
                                       7E  D4 0007C        CLRL    -(SP)                                       ;
             00000000G  00             0C  FB 0007E        CALLS   #12, SYS$QIOW                               ;
                        52             50  D0 00085        MOVL    R0, STATUS                                  ;
                        2E             52  E9 00088        BLBC    STATUS, 4$                                  ; 0625
                        52        90   AD  3C 0008B        MOVZWL  IOSB, STATUS                                ; 0626
                        27             52  E9 0008F        BLBC    STATUS, 4$                                  ;
                                       53  D5 00092        TSTL    TOTAL_NODE_COUNT                            ; 0630
                                       09  12 00094        BNEQ    2$                                          ;
                               0000'   CF  9F 00096        PUSHAB  P.AAT                                       ; 0632
                  0000V  CF            01  FB 0009A        CALLS   #1, WRITE_LINE                              ;
                         55       5C   AE  9E 0009F  2$:   MOVAB   BUFFER, BUFFER_PTR                          ; 0634
                         54       08   AE  D0 000A3        MOVL    KEYS, BUFFER_NODE_COUNT                     ; 0635
                                       BA  15 000A7  3$:   BLEQ    1$                                          ; 0637
                                       55  DD 000A9        PUSHL   BUFFER_PTR                                  ; 0640
                  0000V  CF            01  FB 000AB        CALLS   #1, FORMAT_NODE_INFO                        ;
                         55            50  D0 000B0        MOVL    R0, BUFFER_PTR                              ;
                                       53  D6 000B3        INCL    TOTAL_NODE_COUNT                            ; 0642
                                       54  D7 000B5        DECL    BUFFER_NODE_COUNT                           ; 0643
                                       EE  11 000B7        BRB     3$                                          ; 0637
             00000870   8F             52  D1 000B9  4$:   CMPL    STATUS, #2160                               ; 0648
                                       11  12 000C0        BNEQ    5$                                          ;
                        01             53  D1 000C2        CMPL    TOTAL_NODE_COUNT, #1                        ; 0651
                                       26  15 000C5        BLEQ    8$                                          ;
                                       53  DD 000C7        PUSHL   TOTAL_NODE_COUNT                            ; 0654
                               0000'   CF  9F 000C9        PUSHAB  P.AAV                                       ; 0653
                  0000V  CF            02  FB 000CD        CALLS   #2, WRITE_LINE                              ;
                                       04  04 000D2        RET                                                 ; 0648
             0000007C   8F             52  D1 000D3  5$:   CMPL    STATUS, #124                                ; 0658
                                       08  12 000DA        BNEQ    6$                                          ;
                          00000000G    8F  DD 000DC        PUSHL   #SHOW$_NONET                                ; 0659
                                       02  11 000E2        BRB     7$                                          ;
                                       52  DD 000E4  6$:   PUSHL   STATUS                                      ; 0660
             00000000G  00             01  FB 000E6  7$:   CALLS   #1, LIB$SIGNAL                              ;
                                          04 000ED  8$:    RET                                                 ; 0663

; Routine Size:  238 bytes,    Routine Base:  $CODE$ + 0228
```

```
1 ROUTINE format_area_info (info_ptr: REF VECTOR) =
1
1 !--
1 !
1 !       This routine accepts a pointer to one area's information in the buffer
1 !       returned by NETACP.  It formats this information and writes it to the
1 !       output stream.
1 !
1 ! Inputs:
1 !
1 !       info_ptr = Address of the beginning of the area's information in
1 !               the buffer returned by NETACP.
1 !
1 ! Outputs:
1 !
1 !       Routine value = Address of next byte beyond area's information.
1 !--
1
2 BEGIN
2
2 LOCAL
2     ptr:        REF BBLOCK,                 ! Pointer into area information.
2     circ_name:  VECTOR [2],                 ! Descriptor of circuit name
2     next_hop_name_buffer: VECTOR [32,BYTE], ! Buffer to hold next hop name
2     next_hop_name: VECTOR [2];              ! Descriptor of next hop node name
2
2 next_hop_name [0] = 32;                     ! Make descriptor of output buffer
2 next_hop_name [1] = next_hop_name_buffer;
2 next_hop_name [0] =                         ! Get node name of next hop
2     get_node_name(.info_ptr [3], next_hop_name);
2
2 ptr = info_ptr [4];                         ! Point to word-counted circuit name
2
2 circ_name [0] = .ptr [0,0,16,0];            ! Construct descriptor of circuit name
2 circ_name [1] = .ptr + 2;
2 ptr = .ptr + 2 + .ptr [0,0,16,0];           ! Skip by string in buffer
2
2 !
2 ! Output the line
2 !
2
2 write_line(%ASCID '!13* !3UL   !4UL  !4UL      !10AS-> !6AS !AS',
2             .info_ptr [0],                  ! Area number
2             .info_ptr [1],                  ! Least cost to area
2             .info_ptr [2],                  ! Actual hops to area
2             (IF .circ_name [0] EQL 0 then %ASCID '(Local)' ELSE circ_name), ! Circuit name
2             format_nodeadr(.info_ptr [3]),  ! Next hop node address
2             next_hop_name);                 ! Next hop node name
2
2 RETURN .ptr;                                ! Return updated pointer
2
1 END;
```

```
                                               .PSECT  $PLIT$,NOWRT,NOEXE,2
```

```
55  34  21  20  20  20  4C  55  33  21  20  2A  33  31  21  00200 P.AAY:  .ASCII  \!13* !3UL   !4UL  !4UL        !10AS-> !6AS\   ;
31  21  20  20  20  20  20  20  4C  55  34  21  20  20  4C  0020F                                                                 ;
                        53  41  36  21  20  3E  2D  53  41  30  0021E                                                             ;
                                                53  41  21  20  00228         .ASCII  \ !AS\                                      ;
                                                   010E002C'  0022C P.AAX:  .LONG   17694764                                      ;
                                                   00000000'  00230         .ADDRESS P.AAY                                        ;
                        00  29  6C  61  63  6F  4C  28  00234 P.ABA:  .ASCII  \(Local)\<0>                                        ;
                                                   010E0007   0023C P.AAZ:  .LONG   17694727                                      ;
                                                   00000000'  00240         .ADDRESS P.ABA                                        ;


                                                                            .PSECT  $CODE$,NOWRT,2

                                                       000C 00000 FORMAT_AREA_INFO:
                                                                            .WORD   Save R2,R3                                    ; 0664
                               5E              2C  C2 00002                 SUBL2   #44, SP                                       ;
                                               20  DD 00005                 PUSHL   #32                                           ; 0690
                     04        AE        08    AE  9E 00007                 MOVAB   NEXT_HOP_NAME_BUFFER, NEXT_HOP_NAME+4         ; 0691
                                               5E  DD 0000C                 PUSHL   SP                                            ; 0693
                               53        04    AC  D0 0000E                 MOVL    INFO_PTR, R3                                  ;
                                         0C    A3  DD 00012                 PUSHL   12(R3)                                        ;
                  0000V        CF              02  FB 00015                 CALLS   #2, GET_NODE_NAME                             ;
                               6E              50  D0 0001A                 MOVL    R0, NEXT_HOP_NAME                             ;
                               52        10    A3  9E 0001D                 MOVAB   16(R3), PTR                                   ; 0695
                               50              62  3C 00021                 MOVZWL  (PTR), R0                                     ; 0697
                     28        AE              50  D0 00024                 MOVL    R0, CIRC_NAME                                 ;
                     2C        AE        02    A2  9E 00028                 MOVAB   2(R2), CIRC_NAME+4                            ; 0698
                               52        02  A042  9E 0002D                 MOVAB   2(R0)[PTR], PTR                               ; 0699
                                               5E  DD 00032                 PUSHL   SP                                            ; 0705
                                         0C    A3  DD 00034                 PUSHL   12(R3)                                        ; 0710
                  0000V        CF              01  FB 00037                 CALLS   #1, FORMAT_NODEADR                            ;
                                               50  DD 0003C                 PUSHL   R0                                            ;
                                         30    AE  D5 0003E                 TSTL    CIRC_NAME                                     ; 0709
                                               07  12 00041                 BNEQ    1$                                            ;
                               50     0000'    CF  9E 00043                 MOVAB   P.AAZ, R0                                     ;
                                               04  11 00048                 BRB     2$                                            ;
                               50        30    AE  9E 0004A 1$:             MOVAB   CIRC_NAME, R0                                 ;
                                               50  DD 0004E 2$:             PUSHL   R0                                            ;
                               7E        04    A3  7D 00050                 MOVQ    4(R3), -(SP)                                  ; 0707
                                               63  DD 00054                 PUSHL   (R3)                                          ; 0706
                                      0000'    CF  9F 00056                 PUSHAB  P.AAX                                         ; 0705
                  0000V        CF              07  FB 0005A                 CALLS   #7, WRITE_LINE                                ;
                               50              52  D0 0005F                 MOVL    PTR, R0                                       ; 0713
                                               04     00062                 RET                                                   ; 0715

; Routine Size:  99 bytes,    Routine Base:  $CODE$ + 0316
```

```
1 ROUTINE format_node_info (info_ptr: REF VECTOR) =
1
1 !--
1 !
1 !       This routine accepts a pointer to one node's information in the buffer
1 !       returned by NETACP.  It formats this information and writes it to the
1 !       output stream.
1 !
1 ! Inputs:
1 !
1 !       info_ptr = Address of the beginning of the node's information in
1 !               the buffer returned by NETACP.
1 !
1 ! Outputs:
1 !
1 !       Routine value = Address of next byte beyond node's information.
1 !--
1
2 BEGIN
2
2 LOCAL
2     ptr:            REF BBLOCK,                ! Pointer into node information.
2     node_name:  VECTOR [2],                    ! Descriptor of node name
2     circ_name:  VECTOR [2],                    ! Descriptor of circuit name
2     next_hop_name: VECTOR [2],                 ! Descriptor of next hop node name
2     next_hop_ptr: REF VECTOR [2],              ! Ptr to formatted next hop descriptor
2     next_hop_addr_buffer: VECTOR [32,BYTE], ! Buffer to hold next hop address
2     next_hop_addr: VECTOR [2];                 ! Descriptor of next hop node address
2
2 ptr = info_ptr [5];                           ! Point to word-counted node name
2
2 node_name [0] = .ptr [0,0,16,0];              ! Construct descriptor of node name
2 node_name [1] = .ptr + 2;
2 ptr = .ptr + 2 + .ptr [0,0,16,0];             ! Skip by string in buffer
2
2 next_hop_name [0] = .ptr [0,0,16,0];          ! Construct descriptor of next hop
2 next_hop_name [1] = .ptr + 2;
2 ptr = .ptr + 2 + .ptr [0,0,16,0];             ! Skip by string in buffer
2
2 circ_name [0] = .ptr [0,0,16,0];              ! Construct descriptor of circuit name
2 circ_name [1] = .ptr + 2;
2 ptr = .ptr + 2 + .ptr [0,0,16,0];             ! Skip by string in buffer
2
2 next_hop_ptr = format_nodeadr(.info_ptr [4]);   ! Format next hop address
2 next_hop_addr [0] = .next_hop_ptr [0];   ! Save descriptor of formatted string
2 next_hop_addr [1] = next_hop_addr_buffer;
2 CH$MOVE(.next_hop_ptr [0], .next_hop_ptr [1], .next_hop_addr [1]);
2
2 !
2 ! Output the line
2 !
2
2 write_line(%ASCID '!4* !15<!6AS !AS!> !6UL  !4UL  !4UL     !10AS-> !6AS !AS',
2           format_nodeadr(.info_ptr [0]),      ! Node address
2           node_name,                          ! Node name
2           (IF .info_ptr [1] GEQ 0 THEN .info_ptr [1] ELSE 0), ! Active links
2           .info_ptr [2],                      ! Destination cost
;  603         0773  2               .info_ptr [3],                      ! Destination hops
;  604         0774  2               (IF .circ_name [0] EQL 0 then %ASCID '(Local)' ELSE circ_name), ! Circuit name
;  605         0775  2               next_hop_addr,                      ! Next hop node address
;  606         0776  2               next_hop_name);                     ! Next hop node name
;  607         0777  2
;  608         0778  2  RETURN .ptr;                                     ! Return updated pointer
;  609         0779  2
;  610         0780  1  END;



                                                                             .PSECT  $PLIT$,NOWRT,NOEXE,2

41 21 20 53 41 36 21 3C 35 31 21 20 2A 34 21 00244 P.ABC:   .ASCII  \!4* !15<!6AS !AS!> !6UL  !4UL  !4UL      \  ;
20 4C 55 34 21 20 20 4C 55 36 21 20 3E 21 53 00253                                                              ;
               20 20 20 20 20 4C 55 34 21 20 00262                                                              ;
41 21 20 53 41 36 21 20 3E 2D 53 41 30 31 21 0026C          .ASCII  \!10AS-> !6AS !AS\                          ;
                                          53 0027B                                                              ;
                                    010E0038 0027C P.ABB:   .LONG   17694776                                    ;
                                    00000000' 00280          .ADDRESS P.ABC                                     ;
                     00 29 6C 61 63 6F 4C 28 00284 P.ABE:   .ASCII  \(Local)\<0>                                ;
                                    010E0007 0028C P.ABD:   .LONG   17694727                                    ;
                                    00000000' 00290          .ADDRESS P.ABE                                     ;



                                                                             .PSECT  $CODE$,NOWRT,2

                                    00FC 00000 FORMAT_NODE_INFO:
                                                            .WORD    Save R2,R3,R4,R5,R6,R7                       ; 0716
                    5E      C0   AE   9E 00002              MOVAB    -64(SP), SP                                  ;
                    57      04   AC   D0 00006              MOVL     INFO_PTR, R7                                 ; 0745
                    56      14   A7   9E 0000A              MOVAB    20(R7), PTR                                  ;
                    50           66   3C 0000E              MOVZWL   (PTR), R0                                    ; 0747
               38   AE           50   D0 00011              MOVL     R0, NODE_NAME                                ;
               3C   AE      02   A6   9E 00015              MOVAB    2(R6), NODE_NAME+4                           ; 0748
                    56   02 A046      9E 0001A              MOVAB    2(R0)[PTR], PTR                              ; 0749
                    50           66   3C 0001F              MOVZWL   (PTR), R0                                    ; 0751
               28   AE           50   D0 00022              MOVL     R0, NEXT_HOP_NAME                            ;
               2C   AE      02   A6   9E 00026              MOVAB    2(R6), NEXT_HOP_NAME+4                       ; 0752
                    56   02 A046      9E 0002B              MOVAB    2(R0)[PTR], PTR                              ; 0753
                    50           66   3C 00030              MOVZWL   (PTR), R0                                    ; 0755
               30   AE           50   D0 00033              MOVL     R0, CIRC_NAME                                ;
               34   AE      02   A6   9E 00037              MOVAB    2(R6), CIRC_NAME+4                           ; 0756
                    56   02 A046      9E 0003C              MOVAB    2(R0)[PTR], PTR                              ; 0757
                            10   A7   DD 00041              PUSHL    16(R7)                                       ; 0759
            0000V   CF           01   FB 00044              CALLS    #1, FORMAT_NODEADR                           ;
                    6E           60   D0 00049              MOVL     (NEXT_HOP_PTR), NEXT_HOP_ADDR                ; 0760
               04   AE      08   AE   9E 0004C              MOVAB    NEXT_HOP_ADDR_BUFFER, NEXT_HOP_ADDR+4        ; 0761
     04   BE   04   B0           60   28 00051              MOVC3    (NEXT_HOP_PTR), @4(NEXT_HOP_PTR), -          ; 0762
                                                                     @NEXT_HOP_ADDR+4                             ;
                            28   AE   9F 00057              PUSHAB   NEXT_HOP_NAME                                ; 0768
                            04   AE   9F 0005A              PUSHAB   NEXT_HOP_ADDR                                ;
                            38   AE   D5 0005D              TSTL     CIRC_NAME                                    ; 0774
                                 07   12 00060              BNEQ     1$                                           ;
                    50    0000'  CF   9E 00062              MOVAB    P.ABD, R0                                    ;
                                 04   11 00067              BRB      2$                                           ;
```

```
                  50        38    AE  9E 00069 1$:        MOVAB   CIRC_NAME, R0                              ;
                                  50  DD 0006D 2$:        PUSHL   R0                                         ;
                  7E        08    A7  7D 0006F            MOVQ    8(R7), -(SP)                               ; 0772
                            04    A7  D5 00073            TSTL    4(R7)                                      ; 0771
                                  05  19 00076            BLSS    3$                                         ;
                            04    A7  DD 00078            PUSHL   4(R7)                                      ;
                                  02  11 0007B            BRB     4$                                         ;
                                  7E  D4 0007D 3$:        CLRL    -(SP)                                      ;
                            50    AE  9F 0007F 4$:        PUSHAB  NODE_NAME                                  ; 0768
                                  67  DD 00082            PUSHL   (R7)                                       ; 0769
           0000V  CF              01  FB 00084            CALLS   #1, FORMAT_NODEADR                         ;
                                  50  DD 00089            PUSHL   R0                                         ;
                         0000'    CF  9F 0008B            PUSHAB  P.ABB                                      ; 0768
           0000V  CF              09  FB 0008F            CALLS   #9, WRITE_LINE                             ;
                  50              56  D0 00094            MOVL    PTR, R0                                    ; 0778
                                  04     00097            RET                                                ; 0780

; Routine Size:  152 bytes,    Routine Base:  $CODE$ + 0379
```

```
  1 ROUTINE get_node_name (addr, buffer_desc: REF VECTOR) =
  1
  1 !---
  1 !
  1 !       This routine returns the node name associated with a given node
  1 !       address.
  1 !
  1 ! Inputs:
  1 !
  1 !       addr = Node address
  1 !       buffer_desc = Address of descriptor of output buffer
  1 !
  1 ! Outputs:
  1 !
  1 !       Routine Value = Length of returned string
  1 !-
  1
  2 BEGIN
  2
  2 LOCAL
  2     nfb:        BBLOCK [nfb$c_length+1*4], ! Network function block
  2     nfb_desc:   VECTOR [2],            ! Descriptor of NFB
  2     iosb:       BBLOCK [8],            ! I/O status block
  2     keys:       BBLOCK [4+4+nfb$c_ctx_size], ! Buffer for search keys & context
  2     key_desc:   VECTOR [2],            ! Descriptor of above buffer
  2     buffer:     BBLOCK [16],           ! P4 buffer (for node name)
  2     p4_desc:    VECTOR [2],            ! Descriptor of above buffer
  2     status;
  2
  2 key_desc [0] = 4 + 4 + nfb$c_ctx_size;  ! Longword overhead, ONE search value
  2 key_desc [1] = keys;                    ! and fixed context area
  2
  2 keys [0,0,32,0] = 0;                    ! Zero count of fields in P4 (unused)
  2 keys [4,0,32,0] = .addr;                ! Insert desired node address
  2 keys [8,0,16,0] = 0;                    ! Start key = at beginning
  2
  2 p4_desc [0] = 16;                       ! Setup descriptor of P4 buffer
  2 p4_desc [1] = buffer;
  2
  2 CH$FILL(0,nfb$c_length,nfb);            ! Pre-zero NFB fields
  2
  2 nfb [nfb$b_fct] = nfb$c_fc_show;        ! Request "show" function
  2 nfb [nfb$b_database] = nfb$c_db_ndi;    ! of node database
  2 nfb [nfb$l_srch_key] = nfb$c_ndi_tad;   ! Search for matching address
  2 nfb [nfb$b_oper] = nfb$c_op_eql;        ! using "EQL" comparision
  2
  2 nfb_desc [0] = $BYTEOFFSET(nfb$l_fldid) + 1*4;  ! Construct descriptor of NFB
  2 nfb_desc [1] = nfb;
  2
  2 CH$MOVE(1*4, UPLIT LONG(                ! Request the following fields:
  2             nfb$c_ndi_nna),             ! Node name
  2             nfb [nfb$l_fldid]);
  2
P 2 status = $QIOW(FUNC = IO$_ACPCONTROL,   ! Issue control function
P 2             CHAN = .channel,
P 2             IOSB = iosb,
P 2             P1 = nfb_desc,              ! Address of NDB descriptor
;  669      P 0838  2                  P2 = key_desc,              ! Address of key buffer descriptor
;  670        0839  2                  P4 = p4_desc);              ! Address of return buffer descriptor
;  671        0840  2
;  672        0841  2 IF NOT .status                                 ! If error detected,
;  673        0842  3     OR NOT (status = .iosb [0,0,16,0])
;  674        0843  2 THEN
;  675        0844  2     RETURN 0                                   ! Return null string
;  676        0845  2 ELSE
;  677        0846  3     BEGIN
;  678        0847  3     CH$MOVE(.buffer [0,0,16,0], buffer [2,0,0,0], .buffer_desc [1]);
;  679        0848  3     RETURN .buffer [0,0,16,0];                 ! Return length of string
;  680        0849  2     END;
;  681        0850  2
;  682        0851  1 END;



                                                                           .PSECT  $PLIT$,NOWRT,NOEXE,2

                                              02020043  00294 P.ABF:      .LONG   33685571                        ;


                                                                           .PSECT  $CODE$,NOWRT,2

                                                    003C 00000 GET_NODE_NAME:
                                                                           .WORD   Save R2,R3,R4,R5                ; 0781
                          5E        FF78   CE   9E 00002                   MOVAB   -136(SP), SP                    ;
                   14     AE          48   8F   9A 00007                   MOVZBL  #72, KEY_DESC                   ; 0810
                   18     AE          1C   AE   9E 0000C                   MOVAB   KEYS, KEY_DESC+4                ; 0811
                                      1C   AE   D4 00011                   CLRL    KEYS                            ; 0813
                   20     AE          04   AC   D0 00014                   MOVL    ADDR, KEYS+4                    ; 0814
                                      24   AE   B4 00019                   CLRW    KEYS+8                          ; 0815
                                           10   DD 0001C                   PUSHL   #16                             ; 0817
                   04     AE          08   AE   9E 0001E                   MOVAB   BUFFER, P4_DESC+4               ; 0818
    10             00     6E               00   2C 00023                   MOVC5   #0, (SP), #0, #16, NFB          ; 0820
                                      78   AE      00028                                                           ;
                   78     AE               22   90 0002A                   MOVB    #34, NFB                        ; 0822
                   7A     AE               02   90 0002E                   MOVB    #2, NFB+2                       ; 0823
                   7C     AE  02010010     8F   D0 00032                   MOVL    #33619984, NFB+4                ; 0824
                                      7B   AE   94 0003A                   CLRB    NFB+3                           ; 0825
                   70     AE               14   D0 0003D                   MOVL    #20, NFB_DESC                   ; 0827
                   74     AE          78   AE   9E 00041                   MOVAB   NFB, NFB_DESC+4                 ; 0828
                   FC     AD       0000'   CF   D0 00046                   MOVL    P.ABF, NFB+16                   ; 0830
                                           7E   7C 0004C                   CLRQ    -(SP)                           ; 0839
                                      08   AE   9F 0004E                   PUSHAB  P4_DESC                         ;
                                           7E   D4 00051                   CLRL    -(SP)                           ;
                                      28   AE   9F 00053                   PUSHAB  KEY_DESC                        ;
                                      E4   AD   9F 00056                   PUSHAB  NFB_DESC                        ;
                                           7E   7C 00059                   CLRQ    -(SP)                           ;
                                      DC   AD   9F 0005B                   PUSHAB  IOSB                            ;
                                           38   DD 0005E                   PUSHL   #56                             ;
                   7E              0000'   CF   3C 00060                   MOVZWL  CHANNEL, -(SP)                  ;
                                           7E   D4 00065                   CLRL    -(SP)                           ;
             00000000G    00               0C   FB 00067                   CALLS   #12, SYS$QIOW                   ;
                          07               50   E9 0006E                   BLBC    STATUS, 1$                      ; 0841
                          50          68   AE   3C 00071                   MOVZWL  IOSB, STATUS                    ; 0842
                          03          50  E8 00075       BLBS    STATUS, 2$                              ;
                                      50  D4 00078  1$:  CLRL    R0                                      ; 0846
                                          04 0007A       RET                                             ;
                          50      08  AC  D0 0007B  2$:  MOVL    BUFFER_DESC, R0                         ; 0847
            04    B0    0A  AE      08  AE  28 0007F       MOVC3   BUFFER, BUFFER+2, @4(R0)                ;
                          50      08  AE  3C 00086       MOVZWL  BUFFER, R0                              ; 0848
                                          04 0008A       RET                                             ; 0851

; Routine Size:  139 bytes,    Routine Base:  $CODE$ + 0411
;  684          0852   1 ROUTINE write_line (message, args): NOVALUE =
;  685          0853   1
;  686          0854   1 !---
;  687          0855   1 !
;  688          0856   1 !        This routine accepts a control string and a series of FAO
;  689          0857   1 !        arguments, and writes the resulting line to the output stream.
;  690          0858   1 !
;  691          0859   1 ! Inputs:
;  692          0860   1 !
;  693          0861   1 !        message = Message control string
;  694          0862   1 !        args = First FAO argument (any number of arguments may follow)
;  695          0863   1 !
;  696          0864   1 ! Outputs:
;  697          0865   1 !
;  698          0866   1 !        None
;  699          0867   1 !---
;  700          0868   1
;  701          0869   2 BEGIN
;  702          0870   2
;  703          0871   2 show$write_line(.message, args);          ! Use standard SHOW output routine
;  704          0872   2
;  705          0873   1 END;



                                        0000 00000 WRITE_LINE:
                                                              .WORD    Save nothing                          ; 0852
                              08   AC  9F 00002               PUSHAB   ARGS                                  ; 0871
                              04   AC  DD 00005               PUSHL    MESSAGE                               ;
                  0000G  CF        02  FB 00008               CALLS    #2, SHOW$WRITE_LINE                   ;
                                       04 0000D               RET                                            ; 0873

; Routine Size:  14 bytes,    Routine Base:  $CODE$ + 049C
```

```
  1 ROUTINE format_nodeadr(address) =
  1
  1 !---
  1 !
  1 !       This routine formats a 16-bit node address into an
  1 !       formatted ASCII string of the form <area>.<node>.
  1 !       If the area number is zero, then the area portion
  1 !       is omitted.
  1 !
  1 ! Inputs:
  1 !
  1 !       address = 16-bit node address
  1 !
  1 ! Outputs:
  1 !
  1 !       Routine = Address of descriptor of string describing address
  1 !
  1 !       Since the string & descriptor is stored in OWN storage, it must
  1 !       be copied immediately after returning (with a standard routine
  1 !       such as "append").
  1 !---
  1
  2 BEGIN
  2
  2 OWN
  2     string:     VECTOR [40,BYTE],       ! Formatted node address string
  2     desc:       VECTOR [2];             ! FAO result string descriptor
  2
  2 desc [0] = 40;                          ! Setup descriptor for FAO
  2 desc [1] = string;
  2
  2 IF .address <10,6,0> EQL 0             ! If area = 0,
  2 THEN
P 2     $FAO(%ASCID '   !UL',              ! Format only node
P 2         desc, desc,
  3         .address)
  2 ELSE
P 2     $FAO(%ASCID '!2UL.!UL',            ! Format area and node
P 2         desc, desc,
P 2         .address <10,6,0>,
  2         .address <0,10,0>);
  2
  2 RETURN desc;
  2
  1 END;


                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

 00  00  4C  55  21  20  20  20  00298 P.ABH:   .ASCII  \   !UL\<0><0>
                      010E0006   002A0 P.ABG:   .LONG   17694726
                      00000000'  002A4          .ADDRESS P.ABH
 4C  55  21  2E  4C  55  32  21  002A8 P.ABJ:   .ASCII  \!2UL.!UL\
                      010E0008   002B0 P.ABI:   .LONG   17694728
                      00000000'  002B4          .ADDRESS P.ABJ
```

```
                                                                  .PSECT  $OWN$,NOEXE,2

                                                    00002         .BLKB   2
                                                    00004 STRING: .BLKB   40
                                                    0002C DESC:   .BLKB   8

                                                                  .EXTRN  SYS$FAO

                                                                  .PSECT  $CODE$,NOWRT,2

                                              000C 00000 FORMAT_NODEADR:
                                                                  .WORD   Save R2,R3                              ; 0874
                            53 00000000G  00   9E 00002           MOVAB   SYS$FAO, R3                             ;
                            52     0000'  CF   9E 00009           MOVAB   DESC, R2                                ;
                            62            28   D0 0000E           MOVL    #40, DESC                               ; 0902
                    04      A2      D8    A2   9E 00011           MOVAB   STRING, DESC+4                          ; 0903
                    FC      8F      05    AC   93 00016           BITB    ADDRESS+1, #252                         ; 0905
                                          10   12 0001B           BNEQ    1$                                      ;
                                    04    AC   DD 0001D           PUSHL   ADDRESS                                 ; 0909
                                          52   DD 00020           PUSHL   R2                                      ;
                                          52   DD 00022           PUSHL   R2                                      ;
                                   0000'  CF   9F 00024           PUSHAB  P.ABG                                   ;
                            63            04   FB 00028           CALLS   #4, SYS$FAO                             ;
                                          17   11 0002B           BRB     2$                                      ;
     7E        04    AC     0A            00   EF 0002D 1$:       EXTZV   #0, #10, ADDRESS, -(SP)                 ; 0914
     7E        05    AC     06            02   EF 00033           EXTZV   #2, #6, ADDRESS+1, -(SP)                ;
                                          52   DD 00039           PUSHL   R2                                      ;
                                          52   DD 0003B           PUSHL   R2                                      ;
                                   0000'  CF   9F 0003D           PUSHAB  P.ABI                                   ;
                            63            05   FB 00041           CALLS   #5, SYS$FAO                             ;
                            50            62   9E 00044 2$:       MOVAB   DESC, R0                                ; 0916
                                               04 00047           RET                                             ; 0918

; Routine Size:  72 bytes,    Routine Base:  $CODE$ + 04AA
;  753          0919  1 END
;  754          0920  0 ELUDOM


                                                                   .EXTRN  LIB$SIGNAL

;                                     PSECT SUMMARY
;
;       Name                        Bytes                      Attributes
;
;  $OWN$                               52  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                             696  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                            1266  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        15        0          581        00:01.0
;  _$255$DUA28:[SHRLIB]NET.L32;1                1279        39        3           63        00:00.9



;                                     COMMAND QUALIFIERS

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

; Size:           1266 code + 748 data bytes
; Run Time:          00:24.7
; Elapsed Time:      01:20.3
; Lines/CPU Min:       2237
; Lexemes/CPU-Min: 21745
; Memory Used:  183 pages
; Compilation Complete
```

0056 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SHOMSGUTL
LIS

SHONET
LIS

SHOWAUDIT
LIS

SHOWIO
LIS

SHOWLOG
LIS

SHOWERROR
LIS

SHOWFILES
LIS

SHOMEMORY
LIS