```
NNN          NNN     IIIIIIIII        CCCCCCCCCCCC  NNN          NNN  FFFFFFFFFFFFFFF
NNN          NNN     IIIIIIIII        CCCCCCCCCCCC  NNN          NNN  FFFFFFFFFFFFFFF
NNN          NNN     IIIIIIIII        CCCCCCCCCCCC  NNN          NNN  FFFFFFFFFFFFFFF
NNN          NNN        III        CCC             NNN          NNN  FFF
NNN          NNN        III        CCC             NNN          NNN  FFF
NNN          NNN        III        CCC             NNN          NNN  FFF
NNNNNN       NNN        III        CCC             NNNNNN       NNN  FFF
NNNNNN       NNN        III        CCC             NNNNNN       NNN  FFF
NNNNNN       NNN        III        CCC             NNNNNN       NNN  FFF
NNN    NNN   NNN        III        CCC             NNN    NNN   NNN  FFFFFFFFFFF
NNN    NNN   NNN        III        CCC             NNN    NNN   NNN  FFFFFFFFFFF
NNN    NNN   NNN        III        CCC             NNN    NNN   NNN  FFFFFFFFFFF
NNN       NNNNNN        III        CCC             NNN       NNNNNN  FFF
NNN       NNNNNN        III        CCC             NNN       NNNNNN  FFF
NNN       NNNNNN        III        CCC             NNN       NNNNNN  FFF
NNN          NNN        III        CCC             NNN          NNN  FFF
NNN          NNN        III        CCC             NNN          NNN  FFF
NNN          NNN        III        CCC             NNN          NNN  FFF
NNN          NNN     IIIIIIIII        CCCCCCCCCCCC  NNN          NNN  FFF
NNN          NNN     IIIIIIIII        CCCCCCCCCCCC  NNN          NNN  FFF
NNN          NNN     IIIIIIIII        CCCCCCCCCCCC  NNN          NNN  FFF
NNN          NNN     IIIIIIIII        CCCCCCCCCCCC  NNN          NNN  FFF
```

FILEID**CNFSEND

```
  CCCCCCCC  NN      NN  FFFFFFFFFF    SSSSSSSS  EEEEEEEEEE  NN      NN  DDDDDDDD
  CCCCCCCC  NN      NN  FFFFFFFFFF    SSSSSSSS  EEEEEEEEEE  NN      NN  DDDDDDDD
CC          NN      NN  FF          SS          EE          NN      NN  DD      DD
CC          NN      NN  FF          SS          EE          NN      NN  DD      DD
CC          NNNN    NN  FF          SS          EE          NNNN    NN  DD      DD
CC          NNNN    NN  FF          SS          EE          NNNN    NN  DD      DD
CC          NN  NN  NN  FFFFFFFF      SSSSSS    EEEEEEEE    NN  NN  NN  DD      DD
CC          NN  NN  NN  FFFFFFFF      SSSSSS    EEEEEEEE    NN  NN  NN  DD      DD
CC          NN    NNNN  FF                  SS  EE          NN    NNNN  DD      DD
CC          NN    NNNN  FF                  SS  EE          NN    NNNN  DD      DD
CC          NN      NN  FF                  SS  EE          NN      NN  DD      DD      ....
CC          NN      NN  FF                  SS  EE          NN      NN  DD      DD      ....
  CCCCCCCC  NN      NN  FF          SSSSSSSS    EEEEEEEEEE  NN      NN  DDDDDDDD        ....
  CCCCCCCC  NN      NN  FF          SSSSSSSS    EEEEEEEEEE  NN      NN  DDDDDDDD        ....

LL              IIIIII      SSSSSSSS
LL              IIIIII    SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;  1     0001  0 %TITLE  'DECnet Ethernet Configurator Module'
;  2     0002  0 MODULE CNFSEND                  (
;  3     0003  0                                 LANGUAGE (BLISS32),
;  4     0004  0                                 IDENT = 'V04-000'
;  5     0005  0                                 ) =
;  6     0006  1 BEGIN
;  7     0007  1
;  8     0008  1 !
;  9     0009  1 !*****************************************************************************
; 10     0010  1 !*                                                                           *
; 11     0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 12     0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
; 13     0013  1 !*  ALL RIGHTS RESERVED.                                                      *
; 14     0014  1 !*                                                                           *
; 15     0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 16     0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 17     0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 18     0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 19     0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 20     0020  1 !*  TRANSFERRED.                                                           *
; 21     0021  1 !*                                                                        *
; 22     0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 23     0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 24     0024  1 !*  CORPORATION.                                                           *
; 25     0025  1 !*                                                                        *
; 26     0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 27     0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
; 28     0028  1 !*                                                                        *
; 29     0029  1 !*                                                                        *
; 30     0030  1 !*****************************************************************************
; 31     0031  1 !
; 32     0032  1
; 33     0033  1 !++
; 34     0034  1 ! FACILITY:     DECnet Configurator Module (NICONFIG)
; 35     0035  1 !
; 36     0036  1 ! ABSTRACT:
; 37     0037  1 !
; 38     0038  1 !       This module contains the routines to buffer and send NICE
; 39     0039  1 !       response messages to the processes which interrupt with requests.
; 40     0040  1 !
; 41     0041  1 ! ENVIRONMENT:  VAX/VMS Operating System
; 42     0042  1 !
; 43     0043  1 ! AUTHOR:       Bob Grosso,     CREATION DATE:  18-Jan-1982
; 44     0044  1 !
; 45     0045  1 ! MODIFIED BY:
; 46     0046  1 !
; 47     0047  1 !--
```

```
;    49          0048  1 %SBTTL  'Definitions'
;    50          0049  1
;    51          0050  1 !
;    52          0051  1 ! INCLUDE FILES:
;    53          0052  1 !
;    54          0053  1
;    55          0054  1 LIBRARY 'SYS$LIB  RY:STARLET';          ! VMS common definitions
;    56          0055  1
;    57          0056  1 REQUIRE 'LIB$:CNFDEF.R32';
;    58          0147  1
;    59          0148  1 REQUIRE 'SRC$:CNFPREFIX.REQ';
;    60          0245  1
;    61          0246  1
;    62          0247  1 !
;    63          0248  1 ! BUILTIN functions
;    64          0249  1 !
;    65          0250  1
;    66          0251  1 BUILTIN
;    67          0252  1     INSQUE,                           ! INSQUE instruction
;    68          0253  1     REMQUE;                           ! REMQUE instruction
;    69          0254  1
;    70          0255  1 !
;    71          0256  1 ! TABLE OF CONTENTS:
;    72          0257  1 !
;    73          0258  1
;    74          0259  1 FORWARD ROUTINE
;    75          0260  1
;    76          0261  1         CNF$BUFR_NICE_MSG,            ! Buffer NICE messages into IRB
;    77          0262  1         CNF$SEND_NICE_MSG;            ! Send the NICE message stored in IRB
;    78          0263  1
;    79          0264  1 !
;    80          0265  1 ! EXTERNAL REFERENCES:
;    81          0266  1 !
;    82          0267  1
;    83          0268  1 EXTERNAL ROUTINE
;    84          0269  1
;    85          0270  1 !    Module CNFMAIN
;    86          0271  1
;    87          0272  1         CNF$TRACE,                    ! Log messages to log file
;    88          0273  1         CNF$LOG_DATA,                 ! Log formatted data to log file
;    89          0274  1         CNF$GET_ZVM,                  ! Get zeroed virtual memory
;    90          0275  1         CNF$FREE_VM,                  ! Free virtual memory
;    91          0276  1
;    92          0277  1 !    Module CNFINTRPT
;    93          0278  1
;    94          0279  1         CNF$CLOSE_REQUEST_LINK,       ! After an unsuccessful IO shut down the link and deallocate control
;    95          0280  1         CNF$SOLICIT_REQUEST,
;    96          0281  1
;    97          0282  1 !    Module CNFWORKQ
;    98          0283  1
;    99          0284  1         WKQ$ADD_WORK_ITEM;            ! Add work to work queue
;   100          0285  1
;   101          0286  1
;   102          0287  1 EXTERNAL LITERAL
;   103          0288  1
;   104          0289  1         CNF$_LINK,                    ! Error on logical link
;   105          0290  1
```

```
;  106          0291  1          CNFSC_ASYNCH_EFN;
;  107          0292  1
;  108          0293  1
;  109          0294  1 EXTERNAL
;  110          0295  1
;  111          0296  1          CNF$GL_LOGMASK : BITVECTOR [32];
;  112          0297  1
```

```
;  114          0298  1 %SBTTL  'CNF$BUFR_NICE_MSG  Buffer the response message'
;  115          0299  1 GLOBAL ROUTINE CNF$BUFR_NICE_MSG (IRB, MSG, DEALLOCATE_LEN) =
;  116          0300  1
;  117          0301  1 !++
;  118          0302  1 ! FUNCTIONAL DESCRIPTION:
;  119          0303  1 !
;  120          0304  1 !  Place the NICE message onto a linked list of messages stored
;  121          0305  1 !  in the IRB for later transmission to the connectee.
;  122          0306  1 !
;  123          0307  1 ! FORMAL PARAMETERS:
;  124          0308  1 !
;  125          0309  1 !       irb             Interrupt Request Block, contains context for
;  126          0310  1 !                       I/O with connectee.
;  127          0311  1 !
;  128          0312  1 !       msg             address of buffer containing NICE message to be
;  129          0313  1 !                       stored in the IRB.
;  130          0314  1 !
;  131          0315  1 !       deallocate_len  Length of message to be deallocated after transmission.
;  132          0316  1 !                       Some messages are stored in buffers allocated in VM
;  133          0317  1 !                       and must be deallocated after transmission.  Others
;  134          0318  1 !                       reside on the stack or in OWN storage and shouldn't
;  135          0319  1 !                       be deallocated.
;  136          0320  1 !
;  137          0321  1 ! IMPLICIT INPUTS:
;  138          0322  1 !       NONE
;  139          0323  1 !
;  140          0324  1 ! IMPLICIT OUTPUTS:
;  141          0325  1 !       NONE
;  142          0326  1 !
;  143          0327  1 ! ROUTINE VALUE:
;  144          0328  1 ! COMPLETION CODES:
;  145          0329  1 !       Success
;  146          0330  1 !
;  147          0331  1 !--
;  148          0332  1
;  149          0333  2     BEGIN
;  150          0334  2     MAP
;  151          0335  2         IRB : REF BBLOCK,
;  152          0336  2         MSG : REF BBLOCK;
;  153          0337  2
;  154          0338  2     LOCAL
;  155          0339  2         BNR : REF BBLOCK,
;  156          0340  2         STATUS;
;  157          0341  2
;  158          0342  2
;  159          0343  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
;  160          0344  2         $DESCRIPTOR ('cnf$bufr_nice_msg'));
;  161          0345  2
;  162          0346  2
;  163          0347  2     EXECUTE (CNF$GET_ZVM (%REF (BNR$C_LENGTH), BNR) );          ! Get space to store header and message
;  164          0348  2     BNR [BNR$L_ADDRESS] = .MSG [DSC$A_POINTER];                  ! Record message buffer pointer
;  165          0349  2     BNR [BNR$W_LENGTH] = .MSG [DSC$W_LENGTH];                    ! Record message length
;  166          0350  2
;  167          0351  2     BNR [BNR$W_FREE_LEN] = .DEALLOCATE_LEN;
;  168          0352  2     INSQUE (.BNR, .IRB [IRB$L_BNR_BLINK]);                       ! Queue message onto IRB
;  169          0353  2
;  170          0354  2     RETURN TRUE;
```

```
;  171          0355  1     END;                                                  ! Routine cnf$bufr_nice_msg


                                                                                 .TITLE  CNFSEND DECnet Ethernet Configurator Module
                                                                                 .IDENT  \V04-000\

                                                                                 .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                  45  43  41  52  54  00000 P.AAB:  .ASCII  \TRACE\                                ;
                                                                      00005      .BLKB   3
                                                              00000005  00008 P.AAA:  .LONG   5                                      ;
                                                              00000000' 0000C        .ADDRESS P.AAB                                 ;
6D  5F  65  63  69  6E  5F  72  66  75  62  24  66  6E  63  00010 P.AAD:  .ASCII  \cnf$bufr_nice_msg\                               ;
                                                          67  73  0001F              .BLKB   3                                      ;
                                                                  00021
                                                              00000011  00024 P.AAC:  .LONG   17                                     ;
                                                              00000000' 00028        .ADDRESS P.AAD                                 ;

                                                                                 .EXTRN  CNF$TRACE, CNF$LOG_DATA
                                                                                 .EXTRN  CNF$GET_ZVM, CNF$FREE_VM
                                                                                 .EXTRN  CNF$CLOSE_REQUEST_LINK
                                                                                 .EXTRN  CNF$SOLICIT_REQUEST
                                                                                 .EXTRN  WKQ$ADD_WORK_ITEM
                                                                                 .EXTRN  CNF$_LINK, CNF$C_ASYNCH_EFN
                                                                                 .EXTRN  CNF$GL_LOGMASK

                                                                                 .PSECT  $CODE$,NOWRT,2

                                                      0000 00000                 .ENTRY  CNF$BUFR_NICE_MSG, Save nothing        ; 0299
                                                  5E          08  C2  00002     SUBL2   #8, SP                                 ;
                                              0000' CF        9F  00005     PUSHAB  P.AAC                                  ; 0344
                                              0000' CF        9F  00009     PUSHAB  P.AAA                                  ; 0343
                                                              01  DD  0000D     PUSHL   #1                                     ;
                                          0000G   CF              03  FB  0000F     CALLS   #3, CNF$TRACE                          ;
                                                      04      AE  9F  00014     PUSHAB  BNR                                    ; 0347
                                                04    AE          10  D0  00017     MOVL    #16, 4(SP)                             ;
                                                      04      AE  9F  0001B     PUSHAB  4(SP)                                  ;
                                          0000G   CF              02  FB  0001E     CALLS   #2, CNF$GET_ZVM                        ;
                                                  21              50  E9  00023     BLBC    STATUS, 1$                             ;
                                                  51          04  AE  D0  00026     MOVL    BNR, R1                                ; 0348
                                                  50          08  AC  D0  0002A     MOVL    MSG, R0                                ;
                                                0C    A1        04  A0  D0  0002E     MOVL    4(R0), 12(R1)                          ;
                                                08    A1            60  B0  00033     MOVW    (R0), 8(R1)                            ; 0349
                                                0A    A1        0C  AC  B0  00037     MOVW    DEALLOCATE_LEN, 10(R1)                 ; 0351
                                                  50          04  AC  D0  0003C     MOVL    IRB, R0                                ; 0352
                                                18    B0            61  0E  00040     INSQUE  (R1), @24(R0)                          ;
                                                  50              01  D0  00044     MOVL    #1, R0                                 ; 0354
                                                                      04  00047 1$:  RET                                            ; 0355

; Routine Size:  72 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  173     0356  1 %SBTTL  'CNF$BUFR_ERR_MSG  Buffer the error response message'
;  174     0357  1 GLOBAL ROUTINE CNF$BUFR_ERR_MSG
;  175     0358  1         (IRB, ERR_CODE, ERR_DETAIL, ERR_TXT_COD, ERR_TXT_DSC) =
;  176     0359  1
;  177     0360  1 !++
;  178     0361  1 ! FUNCTIONAL DESCRIPTION:
;  179     0362  1 !
;  180     0363  1 !  Build the error response message and buffer it for later return to
;  181     0364  1 !  the connectee.
;  182     0365  1 !
;  183     0366  1 ! FORMAL PARAMETERS:
;  184     0367  1 !
;  185     0368  1 !       irb             Interrupt Request Block, contains context for
;  186     0369  1 !                       I/O with connectee.
;  187     0370  1 !
;  188     0371  1 !       err_code        The error code is returned in the first byte
;  189     0372  1 !                       of the NICE response message.
;  190     0373  1 !
;  191     0374  1 !       err_detail      The error detail is returned in second and third bytes
;  192     0375  1 !                       of the NICE response message.
;  193     0376  1 !
;  194     0377  1 !       err_txt_cod     An optional error status, for which the error text
;  195     0378  1 !                       will be obtained and buffered
;  196     0379  1 !
;  197     0380  1 !       err_txt_dsc     An optional error text which will be buffered
;  198     0381  1 !
;  199     0382  1 ! IMPLICIT INPUTS:
;  200     0383  1 !       NONE
;  201     0384  1 !
;  202     0385  1 ! IMPLICIT OUTPUTS:
;  203     0386  1 !       NONE
;  204     0387  1 !
;  205     0388  1 ! ROUTINE VALUE:
;  206     0389  1 ! COMPLETION CODES:
;  207     0390  1 !       Always return success
;  208     0391  1 !
;  209     0392  1 ! SIDE EFFECTS:
;  210     0393  1 !
;  211     0394  1 !       Error message is built and buffered and stored in the IRB
;  212     0395  1 !
;  213     0396  1 !--
;  214     0397  1
;  215     0398  2     BEGIN
;  216     0399  2     BUILTIN
;  217     0400  2         NULLPARAMETER;                  ! To check for optional parameters
;  218     0401  2
;  219     0402  2     LITERAL
;  220     0403  2         DECODED_TXT_BUFLEN = 256;       ! Maximum size of text string for decoded error messages
;  221     0404  2
;  222     0405  2     MAP
;  223     0406  2         ERR_TXT_DSC : REF BBLOCK;
;  2       0407  2
;  22      0408  2     LOCAL
;  226     0409  2         ERR_TXTLEN,             ! Either the length of the text decoded from the ERR_TXT_COD
;  227     0410  2                                 !  or the length of optional text in ERR_TXT_DSC
;  228     0411  2         MSG :                   ! Descriptor of message being built
;  229     0412  2           BBLOCK [DSC$C_S_BLN],
```

```
;   230          0413  2          STATUS,
;   231          0414  2          DECODED_TXT_LEN,        ! Length of message text decoded from ERR_TXT_COD
;   232          0415  2          DECODED_TXT_BUFDSC :    ! Descriptor of message text decoded from ERR_TXT_COD
;   233          0416  2            BBLOCK [DSC$C_S_BLN],
;   234          0417  2          DECODED_TXT_BUF :       ! Buffer for message text decoded from ERR_TXT_COD
;   235          0418  2            BBLOCK [DECODED_TXT_BUFLEN];
;   236          0419  2
;   237          0420  2
;   238          0421  2      CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
;   239          0422  2          $DESCRIPTOR ('cnf$bufr_err_msg'));
;   240          0423  2
;   241          0424  2      MSG = 0;                        ! Zero descriptor length and type fields
;   242          0425  2      ERR_TXTLEN = 0;
;   243          0426  2
;   244          0427  2      !
;   245          0428  2      !   Set up descriptor and buffer for decoding the optional error code
;   246          0429  2      !
;   247          0430  2      DECODED_TXT_LEN = 0;
;   248          0431  2      DECODED_TXT_BUFDSC = 0;
;   249          0432  2      DECODED_TXT_BUFDSC [DSC$W_LENGTH] = DECODED_TXT_BUFLEN;
;   250          0433  2      DECODED_TXT_BUFDSC [DSC$A_POINTER] = DECODED_TXT_BUF;
;   251          0434  2
;   252          0435  2      IF NOT NULLPARAMETER (4)
;   253          0436  2      THEN
;   254          0437  2          !
;   255          0438  2          !   Parameter ERR_TXT_COD was provided so decode it
;   256          0439  2          !
;   257          0440  3          BEGIN
;   258        P 0441  3          $GETMSG (MSGID = .ERR_TXT_COD,
;   259        P 0442  3                   MSGLEN = DECODED_TXT_LEN,
;   260          0443  3                   BUFADR = DECODED_TXT_BUFDSC);
;   261          0444  3          ERR_TXTLEN = .DECODED_TXT_LEN;
;   262          0445  3          END
;   263          0446  2      ELSE
;   264          0447  2          !
;   265          0448  2          !   Optional parameter ERR_TXT_COD was not provided so see if
;   266          0449  2          !   ERR_TXT_DSC was and use it instead.
;   267          0450  2          !
;   268          0451  3          BEGIN
;   269          0452  3          IF NOT NULLPARAMETER (5)
;   270          0453  3          THEN
;   271          0454  3              ERR_TXTLEN = .ERR_TXT_DSC [DSC$W_LENGTH];
;   272          0455  2          END;
;   273          0456  2
;   274          0457  2      MSG [DSC$W_LENGTH] = 4 + .ERR_TXTLEN;
;   275          0458  2      EXECUTE (CNF$GET_ZVM (MSG [DSC$W_LENGTH], MSG [DSC$A_POINTER]) );   ! Get space to store message
;   276          0459  2
;   277          0460  2      (.MSG [DSC$A_POINTER]) <0, 8> = .ERR_CODE;          ! First byte is error code
;   278          0461  2      (.MSG [DSC$A_POINTER]) <8, 16> = .ERR_DETAIL;       ! Second and third bytes are error detail
;   279          0462  2      (.MSG [DSC$A_POINTER]) <24, 8> = .ERR_TXTLEN;       ! Fourth byte is length of optional error text
;   280          0463  2
;   281          0464  2      IF .ERR_TXTLEN GTR 0
;   282          0465  2      THEN
;   283          0466  2          !
;   284          0467  2          !   Ooptional text was provided either by decoding ERR_TXT_COD
;   285          0468  2          !   or in ERR_TXT_DSC, so append it to error message being built
;   286          0469  2          !
```

```
;   287           0470  2          CH$MOVE (.ERR_TXTLEN,
;   288           0471  3                   (IF .DECODED_TXT_LEN GTR 0
;   289           0472  3                    THEN
;   290           0473  3                        DECODED_TXT_BUF
;   291           0474  3                    ELSE
;   292           0475  3                        .ERR_TXT_DSC [DSC$A_POINTER]
;   293           0476  2                   ),
;   294           0477  2                   (.MSG [DSC$A_POINTER]) + 4);
;   295           0478  2
;   296           0479  2      CNF$BUFR_NICE_MSG (.IRB, MSG, .MSG [DSC$W_LENGTH]); ! Place the error message in the IRB for later trans
;   297           0480  2      RETURN TRUE;
;   298           0481  1      END;                                                ! Routine cnf$bufr_err_msg


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                  45  43  41  52  54  0002C P.AAF:  .ASCII  \TRACE\                          ;
                                                                      00031         .BLKB   3
                                                        00000005  00034 P.AAE:  .LONG   5                                   ;
                                                        00000000' 00038         .ADDRESS P.AAF                              ;
73  6D  5F  72  72  65  5F  72  66  75  62  24  66  6E  63  0003C P.AAH:  .ASCII  \cnf$bufr_err_msg\                         ;
                                                              67  0004B                                                     ;
                                                        00000010  0004C P.AAG:  .LONG   16                                  ;
                                                        00000000' 00050         .ADDRESS P.AAH                              ;

                                                                .EXTRN  SYS$GETMSG

                                                                .PSECT  $CODE$,NOWRT,2

                                                    003C 00000          .ENTRY  CNF$BUFR_ERR_MSG, Save R2,R3,R4,R5          ; 0357
                                              5E    FEEC   CE   9E 00002  MOVAB   -276(SP), SP                               ;
                                                    0000'  CF   9F 00007  PUSHAB  P.AAG                                     ; 0422
                                                    0000'  CF   9F 0000B  PUSHAB  P.AAE                                     ; 0421
                                                           01   DD 0000F  PUSHL   #1                                        ;
                                           0000G    CF            03   FB 00011  CALLS   #3, CNF$TRACE                      ;
                                                           F8     AD   D4 00016  CLRL    MSG                                 ; 0424
                                                                  52   D4 00019  CLRL    ERR_TXTLEN                          ; 0425
                                                                  6E   D4 0001B  CLRL    DECODED_TXT_LEN                     ; 0430
                                                           F0     AD   D4 0001D  CLRL    DECODED_TXT_BUFDSC                  ; 0431
                                                F0    AD   0100   8F   B0 00020  MOVW    #256, DECODED_TXT_BUFDSC            ; 0432
                                                F4    AD     04   AE   9E 00026  MOVAB   DECODED_TXT_BUF, DECODED_TXT_BUFDSC+4 ; 0433
                                                      04           6C   91 0002B  CMPB    (AP), #4                            ; 0435
                                                                   1D   1F 0002E  BLSSU   1$                                  ;
                                                           10      AC   D5 00030  TSTL    16(AP)                              ;
                                                                   18   13 00033  BEQL    1$                                  ;
                                                      7E           0F   7D 00035  MOVQ    #15, -(SP)                          ; 0443
                                                           F0      AD   9F 00038  PUSHAB  DECODED_TXT_BUFDSC                  ;
                                                           0C      AE   9F 0003B  PUSHAB  DECODED_TXT_LEN                     ;
                                                           10      AC   DD 0003E  PUSHL   ERR_TXT_COD                         ;
                                           00000000G       00           05   FB 00041  CALLS   #5, SYS$GETMSG                 ;
                                                      52           6E   D0 00048  MOVL    DECODED_TXT_LEN, ERR_TXTLEN         ; 0444
                                                                   0E   11 0004B  BRB     2$                                  ; 0435
                                                      05           6C   91 0004D 1$:  CMPB    (AP), #5                        ; 0452
                                                                   09   1F 00050  BLSSU   2$                                  ;
                                                           14      AC   D5 00052  TSTL    20(AP)                              ;
                                                                   04   13 00055  BEQL    2$                                  ;
```

```
                                52        14    BC   3C 00057          MOVZWL  @ERR_TXT_DSC, ERR_TXTLEN                ; 0454
                     F8   AD            52          04   A1 0005B 2$:  ADDW3   #4, ERR_TXTLEN, MSG                    ; 0457
                                                FC   AD   9F 00060     PUSHAB  MSG+4                                   ; 0458
                                                F8   AD   9F 00063     PUSHAB  MSG                                     ;
                           0000G   CF                   02   FB 00066  CALLS   #2, CNF$GET_ZVM                       ;
                                   3E                   50   E9 0006B  BLBC    STATUS, 6$                           ;
                                   51           FC      AD   D0 0006E  MOVL    MSG+4, R1                            ; 0460
                                   61           08      AC   90 00072  MOVB    ERR_CODE, (R1)                       ;
                             01    A1           0C      AC   B0 00076  MOVW    ERR_DETAIL, 1(R1)                    ; 0461
                             03    A1                   52   90 0007B  MOVB    ERR_TXTLEN, 3(R1)                    ; 0462
                                                        52   D5 0007F  TSTL    ERR_TXTLEN                           ; 0464
                                                        17   15 00081  BLEQ    5$                                   ;
                                                        6E   D5 00083  TSTL    DECODED_TXT_LEN                      ; 0471
                                                        06   15 00085  BLEQ    3$                                   ;
                                   50           04      AE   9E 00087  MOVAB   DECODED_TXT_BUF, R0                  ;
                                                        08   11 0008B  BRB     4$                                   ;
                                   50           14      AC   D0 0008D 3$:     MOVL    ERR_TXT_DSC, R0              ; 0475
                                   50           04      A0   D0 00091  MOVL    4(R0), R0                            ;
                     04    A1              60            52   28 00095 4$:     MOVC3   ERR_TXTLEN, (R0), 4(R1)      ; 0477
                                   7E           F8      AD   3C 0009A 5$:     MOVZWL  MSG, -(SP)                   ; 0479
                                                F8   AD   9F 0009E     PUSHAB  MSG                                     ;
                                                04   AC   DD 000A1     PUSHL   IRB                                     ;
                           FF0F    CF                   03   FB 000A4  CALLS   #3, CNF$BUFR_NICE_MSG                 ;
                                   50                   01   D0 000A9  MOVL    #1, R0                               ; 0480
                                                            04 000AC 6$:  RET                                       ; 0481

; Routine Size:  173 bytes,    Routine Base:  $CODE$ + 0048
```

```
;  300     0482  1 %SBTTL  'CNF$SEND_NICE_MSG  send the response message'
;  301     0483  1 GLOBAL ROUTINE CNF$SEND_NICE_MSG (IRB) =
;  302     0484  1
;  303     0485  1 !++
;  304     0486  1 ! FUNCTIONAL DESCRIPTION:
;  305     0487  1 !
;  306     0488  1 !       Called first from CNF$PROCESS_REQUEST, a routine executed off
;  307     0489  1 !       the work queue.  There will be an assumption at this point that
;  308     0490  1 !       the IOSB contains a success from a previous interaction over the
;  309     0491  1 !       channel.  The first NICE message in the IRB is QIO'd and from
;  310     0492  1 !       then on CNF$SEND_NICE_MSG is executed as an AST routine upon QIO
;  311     0493  1 !       completion.  The IOSB is checked before another NICE message
;  312     0494  1 !       is removed and QIO'd.
;  313     0495  1 !
;  314     0496  1 !       When the list is empty then a CNF$SOLICIT_REQUEST is placed on
;  315     0497  1 !       the work queue.
;  316     0498  1 !
;  317     0499  1 ! FORMAL PARAMETERS:
;  318     0500  1 !
;  319     0501  1 !       irb             Interrupt Request Block, contains context for
;  320     0502  1 !                       I/O with connectee.
;  321     0503  1 !
;  322     0504  1 ! IMPLICIT INPUTS:
;  323     0505  1 !
;  324     0506  1 !
;  325     0507  1 ! IMPLICIT OUTPUTS:
;  326     0508  1 !
;  327     0509  1 !       NONE
;  328     0510  1 !
;  329     0511  1 ! ROUTINE VALUE:
;  330     0512  1 ! COMPLETION CODES:
;  331     0513  1 !
;  332     0514  1 !       NONE
;  333     0515  1 !
;  334     0516  1 ! SIDE EFFECTS:
;  335     0517  1 !
;  336     0518  1 !       NONE
;  337     0519  1 !
;  338     0520  1 !--
;  339     0521  1
;  340     0522  2     BEGIN
;  341     0523  2     MAP
;  342     0524  2         IRB : REF BBLOCK;
;  343     0525  2
;  344     0526  2     LOCAL
;  345     0527  2         BNR : REF BBLOCK,
;  346     0528  2         STATUS;
;  347     0529  2
;  348     0530  2
;  349     0531  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
;  350     0532  2         $DESCRIPTOR ('cnf$send_nice_msg'));
;  351     0533  2
;  352     0534  2     !
;  353     0535  2     !   The first time thru, the IOSB should contain a success status
;  354     0536  2     !   from a previous I/O on the channel.  For subsequent passes,
;  355     0537  2     !   CNF$SEND_NICE_MSG will be called to send the next NICE message.
;  356     0538  2     !   Then the IOSB will contain the status for the previous send,
```

```
;  357     0539  2      !   and then if there was an error on the channel, the channel will
;  358     0540  2      !   will be closed.
;  359     0541  2      !
;  360     0542  2      STATUS = .IRB [IRB$W_IOSB];
;  361     0543  2      IF NOT .STATUS
;  362     0544  2      THEN
;  363     0545  3          BEGIN
;  364     0546  3          IF (.STATUS NEQ SS$_LINKABORT) AND
;  365     0547  4             (.STATUS NEQ SS$_LINKEXIT)
;  366     0548  3          THEN
;  367     0549  3              SIGNAL (CNF$_LINK, 0, .STATUS);
;  368     0550  3          WKQ$ADD_WORK_ITEM (CNF$CLOSE_REQUEST_LINK, .IRB);
;  369     0551  3          RETURN TRUE;
;  370     0552  2          END;
;  371     0553  2
;  372     0554  2      !
;  373     0555  2      !   Check to see if this call of the routine follows a call in which a
;  374     0556  2      !   buffered message was sent.  In that case it should now be
;  375     0557  2      !   deallocated.  This would not be the case if this was the first
;  376     0558  2      !   call to this routine.
;  377     0559  2      !
;  378     0560  2      IF .IRB [IRB$W_FREE_LEN] NEQ 0
;  379     0561  2      THEN
;  380     0562  3          BEGIN
;  381     0563  3          EXECUTE (CNF$FREE_VM (%REF(.IRB [IRB$W_FREE_LEN]), IRB [IRB$L_NICE_ADR]) );
;  382     0564  3          IRB [IRB$W_FREE_LEN] = 0;          ! Keep it clean to avoid confusion
;  383     0565  2          END;                               !  when another set of messages are buffered.
;  384     0566  2
;  385     0567  2      !
;  386     0568  2      !   If there are any Buffered NICE Responses in the linked list
;  387     0569  2      !   then remove the next and set it up for sending.  Deallocate the header.
;  388     0570  2      !
;  389     0571  2      IF .IRB [IRB$L_BNR_FLINK] NEQ IRB [IRB$L_BNR_FLINK]
;  390     0572  2      THEN                                   ! There are messages to send
;  391     0573  3          BEGIN                              ! Get the next buffered message ready for sending
;  392     0574  3          BNR = .IRB [IRB$L_BNR_FLINK];
;  393     0575  3          REMQUE (.BNR, STATUS);             ! Remove the next message
;  394     0576  3          IRB [IRB$W_NICE_LEN] = .BNR [BNR$W_LENGTH];
;  395     0577  3          IRB [IRB$L_NICE_ADR] = .BNR [BNR$L_ADDRESS];
;  396     0578  3          IRB [IRB$W_FREE_LEN] = .BNR [BNR$W_FREE_LEN];
;  397     0579  3          EXECUTE (CNF$FREE_VM (%REF(BNR$C_LENGTH), BNR) );
;  398     0580  3          END
;  399     0581  2      ELSE
;  400     0582  2          !
;  401     0583  2          !   No more NICE messages buffered
;  402     0584  2          !   Last request has been completed, solicit another.
;  403     0585  2          !
;  404     0586  3          BEGIN
;  405     0587  3          WKQ$ADD_WORK_ITEM (CNF$SOLICIT_REQUEST, .IRB);
;  406     0588  3          RETURN TRUE;
;  407     0589  2          END;
;  408     0590  2
;  409     0591  2      !
;  410     0592  2      !   If NICE debug logging is enabled, print the NICE message about
;  411     0593  2      !   to be sent.
;  412     0594  2      !
;  413     0595  2      IF .CNF$GL_LOGMASK [DBG$C_NICE]
```

```
;  414          0596  2     THEN
;  415          0597  3         BEGIN
;  416          0598  3         LOCAL DATA_DSC : BBLOCK [DSC$C_S_BLN];
;  417          0599  3         DATA_DSC = 0;
;  418          0600  3         DATA_DSC [DSC$W_LENGTH] = .IRB [IRB$W_NICE_LEN];
;  419          0601  3         DATA_DSC [DSC$A_POINTER] = .IRB [IRB$L_NICE_ADR];
;  420          0602  3         CNF$LOG_DATA (DBG$C_NICE, $DESCRIPTOR ('NICE transmitted'),0, DATA_DSC);
;  421          0603  2         END;
;  422          0604  2
;  423          0605  2     !
;  424          0606  2     !    Send the NICE message
;  425          0607  2     !
;  426        P 0608  2     STATUS = $QIO
;  427        P 0609  2                 (
;  428        P 0610  2                 FUNC = IO$_WRITEVBLK,
;  429        P 0611  2                 CHAN = .IRB [IRB$W_CHAN],
;  430        P 0612  2                 EFN = CNF$C_ASYNCH_EFN,
;  431        P 0613  2                 IOSB = IRB [IRB$W_IOSB],
;  432        P 0614  2                 ASTADR = CNF$SEND_NICE_MSG,
;  433        P 0615  2                 ASTPRM = .IRB,
;  434        P 0616  2                 P1 = .IRB [IRB$L_NICE_ADR],
;  435        P 0617  2                 P2 = .IRB [IRB$W_NICE_LEN]
;  436          0618  2                 );
;  437          0619  2
;  438          0620  2     IF NOT .STATUS
;  439          0621  2     THEN SIGNAL (CNF$_LINK, 0, .STATUS);
;  440          0622  2
;  441          0623  2     RETURN TRUE;
;  442          0624  1     END;                          ! Routine cnf$send_nice_msg


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                        45  43  41  52  54  00054 P.AAJ:  .ASCII  \TRACE\                          ;
                                                        00059           .BLKB   3
                                                    00000005    0005C P.AAI:  .LONG   5                             ;
                                                    00000000'   00060         .ADDRESS P.AAJ                        ;
6D  5F  65  63  69  6E  5F  64  6E  65  73  24  66   6E  63  00064 P.AAL:  .ASCII  \cnf$send_nice_msg\               ;
                                                        67  73  00073                                               ;
                                                                00075           .BLKB   3                           ;
                                                    00000011    00078 P.AAK:  .LONG   17                            ;
                                                    00000000'   0007C         .ADDRESS P.AAL                        ;
65  74  74  69  6D  73  6E  61  72  74  20  45  43   49  4E  00080 P.AAN:  .ASCII  \NICE transmitted\               ;
                                                            64  0008F                                               ;
                                                    00000010    00090 P.AAM:  .LONG   16                            ;
                                                    00000000'   00094         .ADDRESS P.AAN                        ;

                                                        .EXTRN  SYS$QIO

                                                        .PSECT  $CODE$,NOWRT,2

                                                003C    00000          .ENTRY  CNF$SEND_NICE_MSG, Save R2,R3,R4,R5  ; 0483
                                55 00000000G    00  9E  00002          MOVAB   LIB$SIGNAL, R5                       ;
                                54 00000000G    8F  D0  00009          MOVL    #CNF$_LINK, R4                       ;
                                5E                  10  C2  00010      SUBL2   #16, SP                              ;
                                        0000'   CF  9F  00013          PUSHAB  P.AAK                                ; 0532
```

```
                                0000'   CF   9F 00017         PUSHAB  P.AAI                                    ; 0531
                                        01   DD 0001B         PUSHL   #1                                      ;
                      0000G     CF             03   FB 0001D         CALLS   #3, CNF$TRACE                   ;
                                52             04   AC   D0 00022    MOVL    IRB, R2                         ; 0542
                                53             0C   A2   32 00026    CVTWL   12(R2), STATUS                  ;
                                23                  53   E8 0002A    BLBS    STATUS, 2$                      ; 0543
                000020E4        8F                  53   D1 0002D    CMPL    STATUS, #8420                   ; 0546
                                                    12   13 00034    BEQL    1$                              ;
                000020F4        8F                  53   D1 00036    CMPL    STATUS, #8436                   ; 0547
                                                    09   13 0003D    BEQL    1$                              ;
                                                    53   DD 0003F    PUSHL   STATUS                          ; 0549
                                                    7E   D4 00041    CLRL    -(SP)                           ;
                                                    54   DD 00043    PUSHL   R4                              ;
                                65                  03   FB 00045    CALLS   #3, LIB$SIGNAL                  ;
                                                    52   DD 00048 1$:    PUSHL   R2                          ; 0550
                                0000G   CF   9F 0004A         PUSHAB  CNF$CLOSE_REQUEST_LINK                  ;
                                        55   11 0004E         BRB     6$                                      ;
                                  1E    A2   B5 00050 2$:     TSTW    30(R2)                                  ; 0560
                                        17   13 00053         BEQL    4$                                      ;
                                  20    A2   9F 00055         PUSHAB  32(R2)                                  ; 0563
                          04    AE      1E   A2   32 00058    CVTWL   30(R2), 4(SP)                           ;
                                  04    AE   9F 0005D         PUSHAB  4(SP)                                   ;
                      0000G     CF             02   FB 00060         CALLS   #2, CNF$FREE_VM                 ;
                                01             50   E8 00065         BLBS    STATUS, 3$                      ;
                                                    04   00068         RET                                     ;
                                  1E    A2   B4 00069 3$:     CLRW    30(R2)                                  ; 0564
                                50      14   A2   9E 0006C 4$:     MOVAB   20(R2), R0                      ; 0571
                                50      14   A2   D1 00070         CMPL    20(R2), R0                      ;
                                        29   13 00074         BEQL    5$                                      ;
                          04    AE      14   A2   D0 00076         MOVL    20(R2), BNR                     ; 0574
                                53      04   BE   0F 0007B         REMQUE  @BNR, STATUS                    ; 0575
                                51      04   AC   D0 0007F         MOVL    IRB, R1                         ; 0576
                                50      04   AE   D0 00083         MOVL    BNR, R0                         ;
                          1C    A1      08   A0   7D 00087         MOVQ    8(R0), 28(R1)                   ;
                                  04    AE   9F 0008C         PUSHAB  BNR                                     ; 0579
                          04    AE              10   D0 0008F         MOVL    #16, 4(SP)                      ;
                                  04    AE   9F 00093         PUSHAB  4(SP)                                   ;
                      0000G     CF             02   FB 00096         CALLS   #2, CNF$FREE_VM                 ;
                                0E             50   E8 0009B         BLBS    STATUS, 7$                      ;
                                                    04   0009E         RET                                     ;
                                                    52   DD 0009F 5$:     PUSHL   R2                          ; 0587
                                0000G   CF   9F 000A1         PUSHAB  CNF$SOLICIT_REQUEST                     ;
                      0000G     CF             02   FB 000A5 6$:     CALLS   #2, WKQ$ADD_WORK_ITEM           ;
                                        60   11 000AA         BRB     9$                                      ; 0588
                                21      0000G   CF   E9 000AC 7$:     BLBC    CNF$GL_LOGMASK, 8$              ; 0595
                                  08    AE   D4 000B1         CLRL    DATA_DSC                                ; 0599
                                50      04   AC   D0 000B4         MOVL    IRB, R0                         ; 0600
                          08    AE      1C   A0   B0 000B8         MOVW    28(R0), DATA_DSC                ;
                          0C    AE      20   A0   D0 000BD         MOVL    32(R0), DATA_DSC+4              ; 0601
                                  08    AE   9F 000C2         PUSHAB  DATA_DSC                                ; 0602
                                        7E   D4 000C5         CLRL    -(SP)                                   ;
                                0000'   CF   9F 000C7         PUSHAB  P.AAM                                   ;
                                        7E   D4 000CB         CLRL    -(SP)                                   ;
                      0000G     CF             04   FB 000CD         CALLS   #4, CNF$LOG_DATA                ;
                                        7E   7C 000D2 8$:     CLRQ    -(SP)                                   ; 0618
                                        7E   7C 000D4         CLRQ    -(SP)                                   ;
                                50      04   AC   D0 000D6         MOVL    IRB, R0                         ;
```

```
                                7E        1C    A0  32 000DA          CVTWL   28(R0), -(SP)                    ;
                                          20    A0  DD 000DE          PUSHL   32(R0)                           ;
                                                50  DD 000E1          PUSHL   R0                               ;
                                              FF19  CF  9F 000E3      PUSHAB  CNF$SEND_NICE_MSG                ;
                                                0C  A0  9F 000E7      PUSHAB  12(R0)                           ;
                                                30  DD 000EA          PUSHL   #48                              ;
                                7E              0A  A0  32 000EC      CVTWL   10(R0), -(SP)                    ;
                                          00000000G 8F  DD 000F0      PUSHL   #CNF$C_ASYNCH_EFN                ;
                    00000000G   00                  0C  FB 000F6      CALLS   #12, SYS$QIO                     ;
                                53                  50  D0 000FD      MOVL    R0, STATUS                       ;
                                09                  53  E8 00100      BLBS    STATUS, 9$                       ; 0620
                                                    53  DD 00103      PUSHL   STATUS                           ; 0621
                                                    7E  D4 00105      CLRL    -(SP)                            ;
                                                    54  DD 00107      PUSHL   R4                               ;
                                65                  03  FB 00109      CALLS   #3, LIB$SIGNAL                   ;
                                50                  01  D0 0010C 9$:  MOVL    #1, R0                           ; 0623
                                                        04 0010F      RET                                      ; 0624

; Routine Size:  272 bytes,    Routine Base:  $CODE$ + 00F5
```

```
;  444          0625  1 END                                ! End of module CNFSEND
;  445          0626  0 ELUDOM


                                                                .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY
;
;       Name                      Bytes                            Attributes
;
;  $PLIT$                          152  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                          517  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        11         0         581        00:01.1


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

; Size:          517 code + 152 data bytes
; Run Time:          00:12.4
; Elapsed Time:      00:28.1
; Lines/CPU Min:      3031
; Lexemes/CPU-Min: 22300
; Memory Used:  120 pages
; Compilation Complete
```

CNFSEND
LIS

CNFSTORE
LIS

NML.SHR
MAP

CNFSHOW
LIS

CNFWQDEF
LIS

NPADEF
MDL

NMAHEAD
B32

NPAMAC
MAR

NMATAIL
B32

NML

NML
MAP

NMLDDL
B32

NMLDEF
MDL

NMAFILES
LIS

NMAFIELDS
LIS

CNFWORKQ
LIS