```
RRRRRRRRRRR      MMM          MMM      SSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM      SSSSSSSSSSSS
RRRRRRRRRRRR     MMM          MMM      SSSSSSSSSSSS
RRR          RRR MMMMMMM    MMMMMMM   SSS
RRR          RRR MMMMMMM    MMMMMMM   SSS
RRR          RRR MMMMMMM    MMMMMMM   SSS
RRR          RRR MMM   MMM   MMM   MMM SSS
RRR          RRR MMM   MMM   MMM   MMM SSS
RRR          RRR MMM   MMM   MMM   MMM SSS
RRRRRRRRRRR      MMM          MMM      SSSSSSSSS
RRRRRRRRRRRR     MMM          MMM      SSSSSSSSS
RRRRRRRRRRRR     MMM          MMM      SSSSSSSSS
RRR    RRR       MMM          MMM               SSS
RRR    RRR       MMM          MMM               SSS
RRR    RRR       MMM          MMM               SSS
RRR       RRR    MMM          MMM               SSS
RRR       RRR    MMM          MMM               SSS
RRR       RRR    MMM          MMM               SSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
RRR       RRR    MMM          MMM               SSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
RRR          RRR MMM          MMM  SSSSSSSSSSS
```

```
NN       NN  TTTTTTTTTT    000000    DDDDDDDD      IIIIII      SSSSSSSS    CCCCCCCC    000000    NN       NN
NN       NN  TTTTTTTTTT    000000    DDDDDDDD      IIIIII      SSSSSSSS    CCCCCCCC    000000    NN       NN
NN       NN      TT       00    00  DD      DD      II      SS          CC          00    00  NN       NN
NN       NN      TT       00    00  DD      DD      II      SS          CC          00    00  NN       NN
NNNN     NN      TT       00    00  DD      DD      II      SS          CC          00    00  NNNN     NN
NNNN     NN      TT       00    00  DD      DD      II      SS          CC          00    00  NNNN     NN
NN  NN   NN      TT       00  00 00  DD      DD      II        SSSSSS    CC          00    00  NN  NN   NN
NN  NN   NN      TT       00  00 00  DD      DD      II        SSSSSS    CC          00    00  NN  NN   NN
NN    NNNN       TT       0000  00  DD      DD      II              SS  CC          00    00  NN    NNNN
NN    NNNN       TT       0000  00  DD      DD      II              SS  CC          00    00  NN    NNNN
NN       NN      TT       00    00  DD      DD      II              SS  CC          00    00  NN       NN      ....
NN       NN      TT       00    00  DD      DD      II              SS  CC          00    00  NN       NN      ....
NN       NN      TT         000000    DDDDDDDD    IIIIII    SSSSSSSS      CCCCCCCC    000000    NN       NN      ....
NN       NN      TT         000000    DDDDDDDD    IIIIII    SSSSSSSS      CCCCCCCC    000000    NN       NN      ....

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
```

```
0000     1          $BEGIN  NTODISCON,000,NF$NETWORK,<NETWORK DISCONNECT STREAM>
0000     2
0000     3
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; Facility: RMS
0000    31 ;
0000    32 ; Abstract:
0000    33 ;
0000    34 ;       This module communicates with the File Access Listener (FAL) at the
0000    35 ;       remote node to disconnect the specified record stream.
0000    36 ;
0000    37 ; Environment: VAX/VMS, executive mode
0000    38 ;
0000    39 ; Author: James A. Krycka,      Creation Date:  09-DEC-1977
0000    40 ;
0000    41 ; Modified By:
0000    42 ;
0000    43 ;       V02-020 JAK0122         J A Krycka      22-AUG-1983
0000    44 ;               Unlock the remote FAL by sending a Continue Transfer interrupt
0000    45 ;               message on encountering an error due to a $PUT/$WRITE failure
0000    46 ;               that is reported after an Access Complete message has been sent
0000    47 ;               to terminate the file transfer.
0000    48 ;               Rewrite NT$LOOK_FOR_CMP routine to better handle obscure error
0000    49 ;               recovery conditions.
0000    50 ;               Also, move NT$RESUME_FAL routine here from NTOPUT module and
0000    51 ;               modify it to return status of the interrupt I/O operation.
0000    52 ;
0000    53 ;--
```

```
0000     55          .SBTTL  DECLARATIONS
0000     56
0000     57 ;
0000     58 ; Include Files:
0000     59 ;
0000     60
0000     61          $DAPPLGDEF                      ; Define DAP prologue symbols
0000     62          $DAPHDRDEF                      ; Define DAP message header
0000     63          $DAPCMPDEF                      ; Define DAP Access Complete message
0000     64          $DAPCNFDEF                      ; Define DAP Configuration message
0000     65          $DAPCONDEF                      ; Define DAP Continue Transfer message
0000     66          $DAPDATDEF                      ; Define DAP Data message
0000     67          $DAPSTSDEF                      ; Define DAP Status message
0000     68          $IFBDEF                         ; Define IFAB symbols
0000     69          $NWADEF                         ; Define Network Work Area symbols
0000     70
0000     71 ;
0000     72 ; Macros:
0000     73 ;
0000     74 ;        None
0000     75 ;
0000     76 ; Equated Symbols:
0000     77 ;
0000     78
0000     79          ASSUME  DAP$Q_DCODE_FLG EQ 0
0000     80          ASSUME  NWA$Q_FLG EQ 0
0000     81
0000     82 ;
0000     83 ; Own Storage:
0000     84 ;
0000     85 ;        None
0000     86 ;
```

```
                     0000     88          .SBTTL  NT$DISCONNECT - PERFORM NETWORK DISCONNECT FUNCTION
                     0000     89
                     0000     90 ;++
                     0000     91 ;  NT$DISCONNECT - engages in a DAP dialogue with the remote FAL to disconnect
                     0000     92 ;       the record stream.
                     0000     93 ;
                     0000     94 ; Calling Sequence:
                     0000     95 ;
                     0000     96 ;       BSBW    NT$DISCONNECT
                     0000     97 ;
                     0000     98 ; Input Parameters:
                     0000     99 ;
                     0000    100 ;       R8      RAB address
                     0000    101 ;       R9      IRAB address
                     0000    102 ;       R10     IFAB address
                     0000    103 ;       R11     Impure Area address
                     0000    104 ;
                     0000    105 ; Implicit Inputs:
                     0000    106 ;
                     0000    107 ;       DAP$Q_SYSCAP
                     0000    108 ;       DAP$V_GEQ_V42
                     0000    109 ;       DAP$V_GEQ_V56
                     0000    110 ;       DAP$V_VAXVMS
                     0000    111 ;       IFB$V_SQO
                     0000    112 ;
                     0000    113 ; Output Parameters:
                     0000    114 ;
                     0000    115 ;       R0      Status code (RMS)
                     0000    116 ;       R1-R7   Destroyed
                     0000    117 ;       AP      Destroyed
                     0000    118 ;
                     0000    119 ; Implicit Outputs:
                     0000    120 ;
                     0000    121 ;       NWA$V_FTM_RETRV cleared
                     0000    122 ;       NWA$V_FTM_STORE cleared
                     0000    123 ;
                     0000    124 ; Completion Codes:
                     0000    125 ;
                     0000    126 ;       Standard RMS completion codes
                     0000    127 ;
                     0000    128 ; Side Effects:
                     0000    129 ;
                     0000    130 ;       None
                     0000    131 ;
                     0000    132 ;--
                     0000    133
                     0000    134 NT$DISCONNECT::                             ; Entry point
                     0000    135         $TSTPT  NTDISCON                    ;
 57    3C AA    D0  0006    136         MOVL    IFB$L_NWA_PTR(R10),R7       ; Get address of NWA (and DAP)
                     000A    137
                     000A    138 ;++
                     000A    139 ; Build and send DAP Access Complete message to partner.
                     000A    140 ;
                     000A    141 ; Note: The Access Complete message exchange sequence is skipped if the remote
                     000A    142 ;       FAL is a nonVMS partner implemented to DAP before V5.6 (thus the record
                     000A    143 ;       stream is not disconnected at the remote node at this time). However,
                     000A    144 ;       these systems do not support record access mode, and for file transfer
```

```
                     000A   145 ;       mode a close operation will follow which will implicitly disconnect
                     000A   146 ;       the record stream.
                     000A   147 ;--
                     000A   148
                     000A   149 SEND_CMP:                                    ;
04 67    34    E0    000A   150          BBS     #DAP$V_VAXVMS,(R7),10$      ; Send message if partner is VAX/VMS
1F 67    24    E1    000E   151          BBC     #DAP$V_GEQ_V56,(R7),SUC     ;  or if partner uses DAP since V5.6
                     0012   152 10$:     $SETBIT #NWA$V_LAST_MSG,(R7)        ; Declare this last message to block
   50    07    D0    0016   153          MOVL    #DAP$K_CMP_MSG,R0           ; Get message type value
      FFE4'   30    0019   154          BSBW    NT$BUILD_HEAD               ; Construct message header
   85    04    90    001C   155          MOVB    #DAP$K_DISCONN,(R5)+        ; Store CMPFUNC field
      FFDE'   30    001F   156          BSBW    NT$BUILD_TAIL               ; Finish building message
      FFDB'   30    0022   157          BSBW    NT$TRANSMIT                 ; Send Access Complete message to FAL
     14 50    E9    0025   158          BLBC    R0,EXIT                     ; Branch on failure
                     0028   159
                     0028   160 ;++
                     0028   161 ; Receive DAP Access Complete message from partner.
                     0028   162 ;
                     0028   163 ; Note: The DAP V4.1 specification did not permit a response to Access Complete
                     0028   164 ;       (end-of-stream)!!
                     0028   165 ;--
                     0028   166
                     0028   167 RECV_CMP:                                    ;
05 67    21    E1    0028   168          BBC     #DAP$V_GEQ_V42,(R7),SUC     ; Response to Access Complete required
                     002C   169                                              ;  iff partner uses DAP since V4.2
      0F    10    002C   170          BSBB    NT$LOOK_FOR_CMP             ; Wait for response from FAL
   0B 50    E9    002E   171          BLBC    R0,EXIT                     ; Branch if not Access Complete message
                     0031   172 SUC:     RMSSUC                          ; Return success
                     0034   173          $CLRBIT #NWA$V_FTM_RETRV,(R7)   ; Clear file transfer mode status bits
                     0038   174          $CLRBIT #NWA$V_FTM_STORE,(R7)   ;
                  05 003C   175 EXIT:    RSB                             ; Exit with RMS code in R0
```

```
                     003D   177          .SBTTL  NT$LOOK_FOR_CMP
                     003D   178
                     003D   179 ;++
                     003D   180 ; This routine is called after an Access Complete message has been sent to FAL
                     003D   181 ; to terminate the record stream (which may be in an active file transfer mode
                     003D   182 ; state). This routine waits for an Access Complete message response from FAL
                     003D   183 ; to establish a synchronization point while carefully filtering out other valid
                     003D   184 ; messages that may be in the pipe.
                     003D   185 ;--
                     003D   186
                     003D   187 NT$LOOK_FOR_CMP::                           ; Entry point
 67   1D   E0        003D   188          BBS     #NWA$V_FTM_EOF,(R7),-      ; Branch if we're in file transfer
      08              0040   189                  ALREADY_SYNCHRONIZED       ;  retrieval mode with eof reached
 67   1A   E0        0041   190          BBS     #NWA$V_FTM_RETRV,(R7),-    ; Branch if we're in file transfer
      0D              0044   191                  PULLING_FILE               ;  retrieval mode without eof seen
 67   1B   E0        0045   192          BBS     #NWA$V_FTM_STORE,(R7),-    ; Branch if we're in file transfer
      4D              0048   193                  PUSHING_FILE               ;  storage mode
                     0049   194
                     0049   195 ;+
                     0049   196 ; We're either in record access mode or in file transfer retrieval mode with
                     0049   197 ; end-of-file encountered. Thus RMS and FAL are synchronized (nothing should
                     0049   198 ; be in the pipe), so we expect an Access Complete message to be returned in
                     0049   199 ; response to the Access Complete message already sent.
                     0049   200 ;-
                     0049   201
                     0049   202 ALREADY_SYNCHRONIZED:                       ; Terminating record access mode
                     0049   203          $SETBIT #DAP$K_CMP_MSG,DAP$L_MSG_MASK(R7)
                     004E   204                                             ; Expect response of Access Complete msg
    FFAF'  30        004E   205          BSBW    NT$RECEIVE                 ; Get reply from FAL
           05        0051   206          RSB                                ; Exit with RMS code in R0
                     0052   207
                     0052   208 ;+
                     0052   209 ; We're in file transfer mode retrieving a file and have just requested FAL to
                     0052   210 ; terminate the transfer prematurely (i.e., before we've processed end-of-file).
                     0052   211 ; The following actions will be taken while emptying the pipe:
                     0052   212 ;   (1) Access Complete message will cause exit with success.
                     0052   213 ;   (2) Data messages in the pipe will be discarded after computing their CRC
                     0052   214 ;       value (to keep in sync with FAL's CRC computation).
                     0052   215 ;  (3a) Status message pertaining to disconnecting the record stream (and/or
                     0052   216 ;       closing the file) will cause exit with status.
                     0052   217 ;  (3b) Status message pertaining to the interrupted data transfer (end-of-file
                     0052   218 ;       in particular) will be ignored.
                     0052   219 ;   (4) Any error from NT$RECEIVE that is not the result of a valid Status
                     0052   220 ;       message will cause exit with status (link down, protocol error, etc.).
                     0052   221 ;-
                     0052   222
                     0052   223 PULLING_FILE:                               ; Terminating file transfer retrieval
1C A7   00000180 8F  C8     0052   224          BISL2   #<<1@DAP$K_DAT_MSG>!-      ; Expect response of Access Complete,
                     005A   225                    <1@DAP$K_CMP_MSG>!-      ;  Data, or Status (end-of-file) message
                     005A   226                  0>,DAP$L_MSG_MASK(R7)      ;
    FFA3'  30        005A   227          BSBW    NT$RECEIVE                 ; Get reply from FAL
  34 18 A7   E9      005D   228          BLBC    DAP$L_DCODE_STS(R7),30$    ; Branch on failure to obtain a valid
                     0061   229                                             ;  DAP message
                     0061   230
                     0061   231          ASSUME  DAP$K_CMP_MSG EQ 7
                     0061   232          ASSUME  DAP$K_DAT_MSG EQ 8
                     0061   233          ASSUME  DAP$K_STS_MSG EQ 9
```

```
                                0061   234
                                0061   235          $CASEB  SELECTOR=DAP$B_TYPE(R7)-; Dispatch on message type:
                                0061   236                  BASE=#DAP$K_CMP_MSG-    ;
                                0061   237                  DISPL=<-                ;
                                0061   238                      30$-                ; Access Complete
                                0061   239                      10$-                ; Data
                                0061   240                      20$-                ; Status
                                0061   241                  >                       ; Should never take this path
                          15    E1    006C   242 10$:      BBC     #DAP$V_DAPCRC,-   ; Branch if partner does not support
                       28 A7          006E   243                  DAP$Q_SYSCAP(R7),-  ;  file level CRC checksum, then discard
                          E1          0070   244                  PULLING_FILE       ;  data and process next message
                    52    44 A7    7D  0071   245          MOVQ    DAP$Q_FILEDATA(R7),R2 ; Put descriptor of record in <R2,R3>
                       0000'CF    0B  0075   246          CRC     W^NT$CRC_TABLE,-   ; Compute CRC (destroying R0-R3)
                         20 A7        0079   247                  DAP$L_CRC_RSLT(R7),-  ;  using result of previous CRC
                         63    52     007B   248                  R2,(R3)            ;  calculation as initial CRC value
                       20 A7   50  D0  007D   249          MOVL    R0,DAP$L_CRC_RSLT(R7) ; Store CRC resultant value
                          CF   11     0081   250          BRB     PULLING_FILE       ; Process next message
                    0000'8F    50  B1  0083   251 20$:      CMPW    R0,#<RMS$_EOF&^XFFFF> ; Discard end-of-file Status message
                          C8   13     0088   252          BEQL    PULLING_FILE       ;  in pipe
51    40 A7    04    0C    EF  008A   253          EXTZV   #12,#4,DAP$W_STSCODE(R7),R1 ; Examine MACCODE status value
                   07    51    91  0090   254          CMPB    R1,#DAP$_FILE_CLOS ; Is this an error response to the
                                   0093   255                                     ;  Access Complete message we sent?
                         BD    12  0093   256          BNEQ    PULLING_FILE       ; No, ignore error and get next message
                               05  0095   257 30$:      RSB                        ; Exit with RMS code in R0
                                   0096   258
                                   0096   259 ;+
                                   0096   260 ; We're in file transfer mode storing a file and have just requested FAL to
                                   0096   261 ; terminate the transfer (i.e., Access Complete signals end-of-data to FAL).
                                   0096   262 ; The following actions will be taken while emptying the pipe:
                                   0096   263 ;   (1) Access Complete message will cause exit with success.
                                   0096   264 ;  (2a) Status message pertaining to disconnecting the record stream (and/or
                                   0096   265 ;       closing the file) will cause exit with status.
                                   0096   266 ;  (2b) Status message pertaining to the processing of a Data message (ahead
                                   0096   267 ;       of the Access Complete message in the pipe for FAL) will result in the
                                   0096   268 ;       sending of an interrupt Continue Transfer message to unlock FAL, and
                                   0096   269 ;       this (write behind) error will be reported to the caller.
                                   0096   270 ;   (3) Any error from NT$RECEIVE that is not the result of a valid Status
                                   0096   271 ;       message will cause exit with status (link down, protocol error, etc.).
                                   0096   272 ;-
                                   0096   273
                                   0096   274 PUSHING_FILE:                        ; Terminating file transfer storage
                                   0096   275          $SETBIT #DAP$K_CMP_MSG,DAP$L_MSG_MASK(R7)
                                   009B   276                                     ; Expect response of Access Complete msg
                       FF62'       30  009B   277          BSBW    NT$RECEIVE     ; Get reply from FAL
                    25 18 A7       E9  009E   278          BLBC    DAP$L_DCODE_STS(R7),10$ ; Branch on failure to obtain a valid
                                   00A2   279                                     ;  DAP message
                       30 A7       91  00A2   280          CMPB    DAP$B_TYPE(R7),-   ; We're done if this is an Access
                          07           00A5   281                  #DAP$K_CMP_MSG     ;  Complete message; otherwise it must
                          1F       13  00A6   282          BEQL    10$                ;  be a Status message
51    40 A7    04    0C    EF  00A8   283          EXTZV   #12,#4,DAP$W_STSCODE(R7),R1 ; Examine MACCODE status value
                   07    51    91  00AE   284          CMPB    R1,#DAP$_FILE_CLOS ; Is this an error response to the
                                   00B1   285                                     ;  Access Complete message we sent?
                          14       13  00B1   286          BEQL    10$            ; Yes, return status code and exit
                         05    51    91  00B3   287          CMPB    R1,#DAP$_FILE_XFER ; Branch if this was not the result of
                                   00B6   288                                     ;  a $PUT or $WRITE failure; otherwise
                          DE       12  00B6   289          BNEQ    PUSHING_FILE   ;  we're sure FAL is in a wait state
                                   00B8   290
```

```
                     00B8    291 ;+
                     00B8    292 ; FAL has reported an error on a put or write operation while processing a Data
                     00B8    293 ; message that was in the pipe ahead of the Access Complete message, and is
                     00B8    294 ; waiting for a Continue Transfer interrupt message before it can proceed to
                     00B8    295 ; empty its pipe and finally process our Access Complete message.
                     00B8    296 ;-
                     00B8    297
      01    BB       00B8    298          PUSHR   #^M<R0>                         ; Save primary error code
0008  30    BA       00BA    299          BSBW    NT$RESUME_FAL                   ; Then unlock remote FAL by sending a
                     00BD    300                                                  ;  Continue Transfer interrupt message
                     00BD    301          $SETBIT #DAP$K_CMP_MSG,DAP$L_MSG_MASK(R7)
                     00C2    302                                                  ; Expect response of Access Complete msg
FF3B'  30    BA       00C2    303          BSBW    NT$RECEIVE                      ; Get reply from FAL
      01    BA       00C5    304          POPR    #^M<R0>                         ; Restore primary error code
            05       00C7    305 10$:     RSB                                     ; Exit with RMS code in R0
```

```
                     00C8    307          .SBTTL  NT$RESUME_FAL
                     00C8    308
                     00C8    309 ;++
                     00C8    310 ; This routine handles $PUT and $WRITE error recovery for both record access
                     00C8    311 ; and file transfer modes of operation. The DAP specification requires that
                     00C8    312 ; a Continue Transfer message be sent to FAL as an interrupt message to tell
                     00C8    313 ; it what to do with the record (or block) that it failed to $PUT/$WRITE to
                     00C8    314 ; to the file. After sending a Status message to indicate the error condition,
                     00C8    315 ; FAL waits for receipt of an interrupt message, then processes the next
                     00C8    316 ; message in the pipe.
                     00C8    317 ;
                     00C8    318 ; Note: The Continue Transfer message must not be sent on error from an $UPDATE
                     00C8    319 ;       operation; just on an error from a $PUT or $WRITE operation.
                     00C8    320 ;
                     00C8    321 ; Note: If the error occurs in file transfer mode, the file transfer operation
                     00C8    322 ;       will be aborted. Consequently, the next RMS operation that the user
                     00C8    323 ;       requests must be $DISCONNECT or $CLOSE (because FAL expects to receive
                     00C8    324 ;       an Access Complete message). This restriction does not apply to record
                     00C8    325 ;       transfer mode.
                     00C8    326 ;
                     00C8    327 ; Input parameters:
                     00C8    328 ;
                     00C8    329 ;       R7      NWA (=DAP) address
                     00C8    330 ;
                     00C8    331 ; Output Parameters:
                     00C8    332 ;
                     00C8    333 ;       R0      Status code (RMS) to return to user
                     00C8    334 ;       R1-R5   Destroyed
                     00C8    335 ;--
                     00C8    336
                     00C8    337 NT$RESUME_FAL::                         ; Entry point
                     00C8    338
                     00C8    339 ;+
                     00C8    340 ; Build and send DAP Continue Transfer message to partner.
                     00C8    341 ;-
                     00C8    342
                     00C8    343 SEND_CON:                               ;
     50    05    D0  00C8    344         MOVL    #DAP$K_CON_MSG,R0       ; Get message type value
         FF32'   30  00CB    345         BSBW    NT$BUILD_HEAD           ; Construct message header
     85    03    90  00CE    346         MOVB    #DAP$K_ABORT,(R5)+      ; Store CONFUNC field assuming that
                     00D1    347                                         ;  file transfer mode is in effect
 04  67    1B    E0  00D1    348         BBS     #NWA$V_FTM_STORE,(R7),10$;Branch if we guessed right
 FF  A5    02    90  00D5    349         MOVB    #DAP$K_SKIP_REC,-1(R5)  ; Store CONFUNC field appropriate for
                     00D9    350                                         ;  record access mode
         FF24'   30  00D9    351 10$:    BSBW    NT$BUILD_TAIL           ; Finish building message
         FF21'   30  00DC    352         BSBW    NT$INTERRUPT            ; Send Continue Transfer message to FAL
                     00DF    353                                         ;  as an interrupt message
               05    00DF    354         RSB                             ; Exit with RMS code in R0
                     00E0    355
                     00E0    356         .END                            ; End of module
```

```
$$.PSECT_EP                     = 00000000
$$COUNT                         = 00000003
$$RMSTEST                       = 0000001A
$$RMS_PBUGCHK                   = 00000010
$$RMS_TBUGCHK                   = 00000008
$$RMS_UMODE                     = 00000004
ALREADY_SYNCHRONIZED              00000049 R      01
DAP$B_BITCNT                      00000035
DAP$B_CMPFUNC                     00000040
DAP$B_CONFUNC                     00000040
DAP$B_DCODE_FID                   00000019
DAP$B_DCODE_MAC                   0000001B
DAP$B_DCODE_MSG                   0000001A
DAP$B_DECVER                      00000047
DAP$B_ECONUM                      00000045
DAP$B_FILESYS                     00000043
DAP$B_FLAGS                       00000031
DAP$B_LEN256                      00000034
DAP$B_LENGTH                      00000033
DAP$B_OSTYPE                      00000042
DAP$B_STREAMID                    00000032
DAP$B_TYPE                        00000030
DAP$B_USRNUM                      00000046
DAP$B_USRVER                      00000048
DAP$B_VERNUM                      00000044
DAP$B_X_FIELD                     00000024
DAP$C_BLN                         000000C0
DAP$K_ABORT                     = 00000003
DAP$K_BLN                         000000C0
DAP$K_CMP_MSG                   = 00000007
DAP$K_CON_MSG                   = 00000005
DAP$K_DAT_MSG                   = 00000008
DAP$K_DISCONN                   = 00000004
DAP$K_SKIP_REC                  = 00000002
DAP$K_STS_MSG                   = 00000009
DAP$L_CMWA                        00000030
DAP$L_CRC_RSLT                    00000020
DAP$L_DCODE_STS                   00000018
DAP$L_FOP2                        00000044
DAP$L_MSG_MASK                    0000001C
DAP$L_RECNUM1                     00000040
DAP$L_RECNUM2                     00000048
DAP$L_SSPWA                       00000080
DAP$L_STV                         0000004C
DAP$L_TEMP                        00000090
DAP$M_BITCNT                    = 00000008
DAP$M_SEGMENT                   = 00000040
DAP$M_TMP1$                     = 00000010
DAP$M_TMP2$                     = 00000080
DAP$Q_DCODE_FLG                   00000000
DAP$Q_FILEDATA                    00000044
DAP$Q_MSG_BUF1                    00000008
DAP$Q_MSG_BUF2                    00000010
DAP$Q_STX                         00000050
DAP$Q_SYSCAP                      00000028
DAP$Q_SYSPEC                      00000038
DAP$V_DAPCRC                    = 00000015
DAP$V_GEQ_V42                   = 00000021
DAP$V_GEQ_V56                   = 00000024
DAP$V_VAXVMS                    = 00000034
DAP$W_BUFSIZ                      00000040
DAP$W_CHECK                       00000042
DAP$W_PARTNER                     00000006
DAP$W_RFA                         00000042
DAP$W_STSCODE                     00000040
DAP$W_VERSION                     00000004
DAP$_FILE_CLOS                  = 00000007
DAP$_FILE_XFER                  = 00000005
EXIT                              0000003C R      01
IFB$L_NWA_PTR                   = 0000003C
NT$BUILD_READ                     ********    X   01
NT$BUILD_TAIL                     ********    X   01
NT$CRC_TABLE                      ********    X   01
NT$DISCONNECT                     00000000 RG     01
NT$INTERRUPT                      ********    X   01
NT$LOOK_FOR_CMP                   0000003D RG     01
NT$RECEIVE                        ********    X   01
NT$RESUME_FAL                     000000C8 RG     01
NT$TRANSMIT                       ********    X   01
NWA$B_ALLXABCNT                   0000011C
NWA$B_DAP_RAC                     000000C9
NWA$B_FILESYS                     000000C5
NWA$B_KEYXABCNT                   0000011D
NWA$B_NETSTRSIZ                   0000016F
NWA$B_NODBUFSIZ                   00000168
NWA$B_ORG                         000000C6
NWA$B_OSTYPE                      000000C4
NWA$B_RFM                         000000C7
NWA$B_RMS_RAC                     000000C8
NWA$C_BLN                         00000800
NWA$K_BLN                         0000080C
NWA$L_ALLXABADR                   00000100
NWA$L_DATXABADR                   00000104
NWA$L_DEV                         000000C0
NWA$L_FHCXABADR                   00000108
NWA$L_KEYXABADR                   0000010C
NWA$L_MSG_MASK                    000000D4
NWA$L_PROXABADR                   00000110
NWA$L_RDTXABADR                   00000114
NWA$L_SAVE_FLGS                   00000128
NWA$L_SUMXABADR                   00000118
NWA$L_THREAD                      000000FC
NWA$L_XLTATTR                     00000238
NWA$L_XLTBUFFLG                   0000022C
NWA$L_XLTCNT                      00000228
NWA$L_XLTMAXINDX                  00000234
NWA$L_XLTSIZ                      00000230
NWA$Q_ACS                         00000244
NWA$Q_BIGBUF                      00000170
NWA$Q_BLD                         000000F0
NWA$Q_FLG                         00000000
NWA$Q_INODE                       0000025C
NWA$Q_IOSB                        000000D8
NWA$Q_LNODE                       00000160
```

```
NWA$Q_LOGNAME                   0000023C
NWA$Q_NCB                       00000264
NWA$Q_RCV                       000000E0
NWA$Q_SAVE_DESC                 00000120
NWA$Q_XLTBUF1                   0000024C
NWA$Q_XLTBUF2                   00000254
NWA$Q_XMT                       000000E8
NWA$T_ACSBUF                    0000026C
NWA$T_AUXBUF                    000005E0
NWA$T_DAP                       00000000
NWA$T_INODEBUF                  000004AC
NWA$T_ITM_ATTR                  00000200
NWA$T_ITM_END                   00000224
NWA$T_ITM_LST                   00000200
NWA$T_ITM_MAXINDX               00000218
NWA$T_ITM_STRING                0000020C
NWA$T_NCBBUF                    0000052C
NWA$T_NODEBUF                   00000169
NWA$T_RCVBUF                    000001A0
NWA$T_SCAN                      00000100
NWA$T_TEMP                      00000120
NWA$T_XLTBUF1                   000002AC
NWA$T_XLTBUF2                   000003AC
NWA$T_XMTBUF                    000003C0
NWA$V_FTM_EOF                 = 0000001D
NWA$V_FTM_RETRV               = 0000001A
NWA$V_FTM_STORE               = 0000001B
NWA$V_LAST_MSG                = 00000000
NWA$W_BUILD                     000000D2
NWA$W_DAPBUFSIZ                 000000CA
NWA$W_DIR_OFF                   000000CC
NWA$W_DISPLAY                   000000D0
NWA$W_FIL_OFF                   000000CE
NWA$W_JNLXABJOP                 0000011E
PIO$A_TRACE                     ********    X   01
PULLING_FILE                    00000052 R      01
PUSHING_FILE                    00000096 R      01
RECV_CMP                        00000028 R      01
RMS$_EOF                        ********    X   01
SEND_CMP                        0000000A R      01
SEND_CON                        000000C8 R      01
SUC                             00000031 R      01
TPT$L_NTDISCON                  ********    X   01
```

+----------------+
! Psect synopsis !
+----------------+

PSECT name	Allocation		PSECT No.		Attributes									
. ABS .	00000000	(0.)	00	(0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC BYTE
NF$NETWORK	000000E0	(224.)	01	(1.)	PIC	USR	CON	REL	GBL	NOSHR	EXE	RD	NOWRT	NOVEC BYTE
ABS	00000800	(2048.)	02	(2.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.09	00:00:00.70
Command processing	116	00:00:00.64	00:00:05.53
Pass 1	234	00:00:07.12	00:00:18.52
Symbol table sort	0	00:00:00.77	00:00:01.33
Pass 2	86	00:00:01.47	00:00:04.34
Symbol table output	19	00:00:00.16	00:00:01.00
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	494	00:00:10.30	00:00:31.47

The working set limit was 1350 pages.
33570 bytes (66 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 539 non-local and 13 local symbols.
356 source lines were read in Pass 1, producing 14 object records in Pass 2.
25 pages of virtual memory were used to define 24 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[RMS.OBJ]RMS.MLB;1	16
_255DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	20

749 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:NTODISCON/OBJ=OBJ$:NTODISCON MSRC$:NTODISCON/UPDATE=(ENH$:NTODISCON)+LIB$:RMS/LIB
```

NT0DAPRMS
LIS

NT0GET
LIS

NT0NWASET
LIS

NT0EXTEND
LIS

NT0DECODE
LIS

NT0ENCODE
LIS

NT0ERASE
LIS

NT0DISCON
LIS

NT0DISPLY
LIS

NT0MISC
LIS