C 13

```
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRRRRRRRRRRR     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  SSSSSSSSSSSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
```

```
_$
Sy
---
NT$
NT$
NT$
NT$
NT$
NT$

NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$
NT$

NT$

NT$
NT$
NT$
NT$
NT$
NT$
NT

NT
NT
NT
NT
NT
NT
PI
```

```
**FILE**ID**NTOACCFIL                                   N  6
```

```
NN      NN  TTTTTTTTTT    000000      AAAAAA      CCCCCCCC    CCCCCCCC  FFFFFFFFFF    IIIIII    LL
NN      NN  TTTTTTTTTT    000000      AAAAAA      CCCCCCCC    CCCCCCCC  FFFFFFFFFF    IIIIII    LL
NN      NN      TT    00      00  AA      AA  CC          CC          FF              II      LL
NN      NN      TT    00      00  AA      AA  CC          CC          FF              II      LL
NNNN    NN      TT    00    0000  AA      AA  CC          CC          FF              II      LL
NNNN    NN      TT    00    0000  AA      AA  CC          CC          FF              II      LL
NN  NN  NN      TT    00  00  00  AA      AA  CC          CC          FFFFFFFF        II      LL
NN  NN  NN      TT    00  00  00  AA      AA  CC          CC          FFFFFFFF        II      LL
NN    NNNN      TT    0000    00  AAAAAAAAAA  CC          CC          FF              II      LL
NN    NNNN      TT    0000    00  AAAAAAAAAA  CC          CC          FF              II      LL
NN      NN      TT    00      00  AA      AA  CC          CC          FF              II      LL               ....
NN      NN      TT    00      00  AA      AA  CC          CC          FF              II      LL               ....
NN      NN      TT      000000    AA      AA    CCCCCCCC    CCCCCCCC  FF            IIIIII    LLLLLLLLLL       ....
NN      NN      TT      000000    AA      AA    CCCCCCCC    CCCCCCCC  FF            IIIIII    LLLLLLLLLL       ....

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
```

NT
VO

```
    (2)       95        DECLARATIONS
    (3)       128       NT$EXCH_CNF - EXCHANGE DAP CONFIGURATION MESSAGES
    (4)       436       NT$GET_FILESPEC - BUILDS A FILESPEC
```

```
0000        $BEGIN  NTOACCFIL,000,NF$NETWORK,<COMMON FILE ACCESS ROUTINES>
0000
0000
0000  ;
0000  ;*************************************************************************
0000  ;*                                                                       *
0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.               *
0000  ;*  ALL RIGHTS RESERVED.                                                 *
0000  ;*                                                                       *
0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
0000  ;*  TRANSFERRED.                                                         *
0000  ;*                                                                       *
0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0000  ;*  CORPORATION.                                                         *
0000  ;*                                                                       *
0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FuR THE USE  OR  RELIABILITY OF ITS *
0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.              *
0000  ;*                                                                       *
0000  ;*                                                                       *
0000  ;*************************************************************************
0000  ;
0000
0000  ;++
0000  ; Facility: RMS
0000  ;
0000  ; Abstract:
0000  ;
0000  ;       This module contains commonly used file access support routines.
0000  ;
0000  ; Environment: VAX/VMS, executive mode
0000  ;
0000  ; Author: James A. Krycka,      Creation Date:  09-DEC-1977
0000  ;
0000  ; Modified By:
0000  ;
0000  ;       V03-018 JEJ0039         J E Johnson             19-Jun-1984
0000  ;               Fix NT$GET_FILESPEC to recognize that a resultant file
0000  ;               spec has already been given back.  This is of importance
0000  ;               because of the possible open loop in NT$OPEN regarding to
0000  ;               BRO option.
0000  ;
0000  ;       V03-017 JAK0145         J A Krycka      12-APR-1984
0000  ;               Track changes in DAP message building algorithm.
0000  ;
0000  ;       V03-016 JAK0138         J A Krycka      28-MAR-1984
0000  ;               Use process or system network block count value (PIO$GB_DFNBC
0000  ;               or SYS$GB_DFNBC) instead of process buffered I/O byte limit
0000  ;               quota (BYTLM value from the UAF) as the basis for calculating a
0000  ;               requested DAP buffer size to send to the remote FAL in the DAP
0000  ;               Configuration message. SET RMS_DEFAULT and SHOW RMS_DEFAULT will
0000  ;               allow the network block count value to be set from 1 to 127.
```

```
0000     ;                       The size of NBC affects both file transfer performance and the
0000     ;                       largest record that can be exchanged in record I/O mode. The
0000     ;                       remote FAL, of course, can force a smaller DAP buffer size (thar
0000     ;                       RMS requests) to be used.
0000     ;
0000     ;               V03-015 JAK0133         J A Krycka      20-MAR-1984
0000     ;                       Lay framework for use of logical link QIOs to transfer more
0000     ;                       than 4K bytes of data.
0000     ;
0000     ;               V03-014 RAS0223         Ron Schaefer    16-Dec-1983
0000     ;                       Change $SCBDEF and SCB$xxx to $FSCBDEF and FSCB$xxx.
0000     ;
0000     ;               V03-013 JAK0127         J A Krycka      12-SEP-1983
0000     ;                       Correct NT$GET_FILESPEC to properly check for elipsis in
0000     ;                       directory component of a file specification.
0000     ;
0000     ;               V03-012 JAK0105         J A Krycka      11-MAY-1983
0000     ;                       Enable access to large records (over 512 bytes in length) based
0000     ;                       on the negotiated DAP buffer size.
0000     ;
0000     ;               V03-011 KRM0107         K Malik         10-May-1983
0000     ;                       Update to support DAP V7.0 specification.
0000     ;
0000     ;               V03-010 JAK0104         J A Krycka      22-APR-1983
0000     ;                       Enhance NT$EXCH_CNF to dynamically determine a suggested DAP
0000     ;                       buffer size to send to FAL (instead of using a predetermined
0000     ;                       value) based on BYTLM quota obtained from a call to $GETJPI.
0000     ;
0000     ;               V03-009 KRM0079         K Malik         31-Jan-1983
0000     ;                       Turn on indirect command file support.
0000     ;
0000     ;               V03-008 KRM0053         K Malik         10-Aug-1982
0000     ;                       Move nodename from FWA$T_NODEBUF to NWA$T_NODEBUF (used by
0000     ;                       NT$CRC_LOGERR)
0000     ;
0000     ;--
```

```
0000          .SBTTL  DECLARATIONS
0000
0000    ;
0000    ; Include Files:
0000    ;
0000
0000          $DAPPLGDEF                        ; Define DAP prologue symbols
0000          $DAPHDRDEF                        ; Define DAP message header
0000          $DAPCNFDEF                        ; Define DAP Configuration message
0000          $DAPACCDEF                        ; Define DAP Access message
0000          $FABDEF                           ; Define File Access Block symbols
0000          $FWADEF                           ; Define File Work Area symbols
0000          $IFBDEF                           ; Define IFAB symbols
0000          $NWADEF                           ; Define Network Work Area symbols
0000          $FSCBDEF                          ; Define Scan Control Block symbols
0000
0000    ;
0000    ; Macros:
0000    ;
0000          None
0000    ;
0000    ; Equated Symbols:
0000    ;
0000
0000          ASSUME  DAP$Q_DCODE_FLG EQ 0
0000          ASSUME  NWA$Q_FLG EQ 0
0000
0000    ;
0000    ; Own Storage:
0000    ;
0000          None
0000    ;
```

```
0000          .SBTTL  NT$EXCH_CNF - EXCHANGE DAP CONFIGURATION MESSAGES
0000
0000    ;++
0000    ; NT$EXCH_CNF - engages in a DAP dialogue to exchange DAP Configuration
0000    ;       messages with the remote FAL which includes negotiation of a DAP
0000    ;       buffer size for subsequent message exchange.
0000    ;
0000    ; Calling Sequence:
0000    ;
0000    ;       BSBW    NT$EXCH_CNF
0000    ;
0000    ; Input Parameters:
0000    ;
0000    ;       R0      Type of file access (DAP ACCFUNC code)
0000    ;       R7      NWA address (=DAP)
0000    ;       R8      FAB address
0000    ;       R9      IFAB address
0000    ;       R10     FWA address
0000    ;       R11     Impure Area address
0000    ;
0000    ; Implicit Inputs:
0000    ;
0000    ;       DAPDEF constants
0000    ;       DAP$W_BUFSIZ
0000    ;       FWA$T_NODEBUF
0000    ;       IFB$L_DEVBUFSIZ
0000    ;       PIO$GB_DFNBC
0000    ;       SYS$GB_DFNBC
0000    ;
0000    ; Output Parameters:
0000    ;
0000    ;       R0      Status code (RMS)
0000    ;       R1-R6   Destroyed
0000    ;       AP      Destroyed
0000    ;
0000    ; Implicit Outputs:
0000    ;
0000    ;       DAP control block
0000    ;       NWA$W_DAPBUFSIZ
0000    ;       NWA$T_NODEBUF
0000    ;       NWA$B_NODBUFSIZ
0000    ;       NWA$B_FILESYS
0000    ;       NWA$B_OSTYPE
0000    ;       NWA$Q_RCV
0000    ;       NWA$Q_XMT
0000    ;       NWA$Q_BIGBUF
0000    ;       IFB$L_DEVBUFSIZ
0000    ;
0000    ; Completion Codes:
0000    ;
0000    ;       Standard RMS completion codes
0000    ;
0000    ; Side Effects:
0000    ;
0000    ;       None
0000    ;
0000    ;--
```

```
                                 0000
                                 0000   NT$EXCH_CNF::                              ; Entry point
                                 0000
                                 0000   ;
                                 0000   ; Obtain specified Network Block Count (NBC) value which is a sysgen parameter
                                 0000   ; and also settable via the DCL command SET RMS_DEFAULT/NETWORK_BLOCK_COUNT = n.
                                 0000   ;
                                 0000
       55   00000000'9F    98    0000          CVTBL   @#PIO$GB_DFNBC,R5          ; Get process network block count value
                     0C    14    0007          BGTR    10$                        ; Branch if specified (non-zero & pos)
       55   00000000'9F    98    0009          CVTBL   @#SYS$GB_DFNBC,R5          ; Get system network block count value
                     03    14    0010          BGTR    10$                        ; Branch if specified (non-zero & pos)
              55     01    D0    0012          MOVL    #1,R5                      ; We should never get here as range of
                                 0015                                             ;  DFNBC sysgen parameter is 1 to 127
           7F 8F     55    91    0015   10$:   CMPB    R5,#127                    ; Make internal NBC value between 1 and
                     02    12    0019          BNEQ    20$                        ;  126 so that requested DAP buffer size
                     55    D7    001B          DECL    R5                         ;  (with overhead) will be < 65536 bytes
                                 001D
                                 001D   ;
                                 001D   ; Dispatch to file operation specific code to process network block count.
                                 001D   ;
                                 001D
                                 001D          ASSUME  DAP$K_OPEN EQ 1
                                 001D          ASSUME  DAP$K_CREATE EQ 2
                                 001D          ASSUME  DAP$K_RENAME EQ 3
                                 001D          ASSUME  DAP$K_ERASE EQ 4
                                 001D          ASSUME  DAP$K_DIR_LIST EQ 6
                                 001D
                                 001D   20$:   $CASEB  SELECTOR=R0-               ; Dispatch on file access type
                                 001D                  BASE=#DAP$K_OPEN-          ;
                                 001D                  DISPL=<-                   ;
                                 001D                      OPEN_CREATE-           ; Open file
                                 001D                      OPEN_CREATE-           ; Create file
                                 001D                      ERASE_RENAME-          ; Rename file
                                 001D                      ERASE_RENAME-          ; Erase file
                                 001D                      ERASE_RENAME-          ; Reserved
                                 001D                      SEARCH-                ; Search file (DAP directory list)
                                 001D                  >                          ; Any other type of access
                                 002D
                                 002D   ;+
                                 002D   ; For erase and rename operations set NBC to one for minimum DAP buffer size.
                                 002D   ;-
                                 002D
                                 002D   ERASE_RENAME:                             ; For $ERASE and $RENAME operations
              55     01    D0    002D          MOVL    #1,R5                      ; One page of memory is sufficient
                     23    11    0030          BRB     EXCH_COMMON                ; Join common code
                                 0032
                                 0032   ;+
                                 0032   ; For a search operation, scale down NBC (DAP buffer size) by a factor of four
                                 0032   ; to reduce the time waiting for FAL to send the next bufferred set of messages,
                                 0032   ; especially when FAL must open each file to return file attribute information.
                                 0032   ; This will help to "smooth out" the display for the DCL DIRECTORY command.
                                 0032   ;
                                 0032   ; However, for access to a process permanent file, set NBC to one for minimum
                                 0032   ; DAP buffer size to conserve use of the process I/O segment in P1 space.
                                 0032   ;-
                                 0032
```

```
                         0032    SEARCH:                                         ; For $SEARCH operation
             03 6B   E8  0032            BLBS    (R11),10$                       ; Branch if not accessing a process
             55 01   D0  0035            MOVL    #1,R5                           ;  permanent file, else use one block
             55 03   C0  0038    10$:    ADDL2   #3,R5                           ; Reduce NBC to approximately one-fourth
  55  55  FE 8F      78  003B            ASHL    #-2,R5,R5                       ;  its value
                13   11  0040            BRB     EXCH_COMMON                     ; Join common code
                         0042
                         0042    ;+
                         0042    ; For open and create operations use the specified NBC value as the basis for
                         0042    ; generating requested DAP buffer size unless we have a process permanent file.
                         0042    ;
                         0042    ; For access to a process permanent file, scale down NBC (DAP buffer size) by a
                         0042    ; factor of eight to conserve use of the process I/O segment in P1 space that
                         0042    ; is available to RMS. This reduction serves to increase the total number of
                         0042    ; process permanent files that can be simultaneously open for network access.
                         0042    ; Since the DCL OPEN command opens a process permanent file and the DCL READ and
                         0042    ; and WRITE commands are limited to 2048 byte records, the maximum NBC value
                         0042    ; will be 4 for process permanent files.
                         0042    ;-
                         0042
                         0042    OPEN_CREATE:                                    ; For $OPEN and $CREATE operations
             10 6B   E8  0042            BLBS    (R11),EXCH_COMMON               ; Branch if not accessing a process
                         0045                                                    ;  permanent file
             55 07   C0  0045            ADDL2   #7,R5                           ; Reduce NBC to approximately one-eighth
  55  55  FD 8F      78  0048            ASHL    #-3,R5,R5                       ;  its value but not more than 4, so the
             04 50   D1  004D            CMPL    R0,#4                           ;  resultant value is in the range of
                03   15  0050            BLEQ    EXCH_COMMON                     ;  1 to 4 blocks
             55 04   D0  0052            MOVL    #4,R5                           ;
                         0055
                         0055    ;+
                         0055    ; Compute DAP buffer size to request in the DAP Configuration message based on:
                         0055    ;    (1) the (modified) network block count value,
                         0055    ;    (2) the addition of up to 8 bytes of overhead per DAP DATA message,
                         0055    ;    (3) the desire to be able to block a DAP CONTROL message with the first set
                         0055    ;        of blocked DATA messages in file transfer mode,
                         0055    ;    (4) the desire to have the DAP buffer fit into the nominal line buffer size
                         0055    ;        of 576 bytes (which includes lower layer protocol overhead) when the NBC
                         0055    ;        is one or the remote FAL can support only a one block data buffer.
                         0055    ;-
                         0055
                         0055    EXCH_COMMON:                                    ; Compute DAP buffer size to request
             08 55   D1  0055            CMPL    R5,#8                           ; Choose a formula based on NBC size
                0A   15  0058            BLEQ    10$                             ;  to optimize requested buffer size
      56   0208 8F   3C  005A            MOVZWL  #<512+8>,R6                     ; Compute desired buffer size using the
             56 55   C4  005F            MULL2   R5,R6                           ;  formula: (NBC * (512+8)) where
                0B   11  0062            BRB     EXCH_INIT                       ;  NBC has a value from 9 to 126
      56   0204 8F   3C  0064    10$:    MOVZWL  #<512+4>,R6                     ; Compute desired buffer size using the
             56 55   C4  0069            MULL2   R5,R6                           ;  formula: (NBC * (512+4) + 28) where
             56 1C   C0  006C            ADDL2   #28,R6                          ;  NBC has a value from 1 to 8
                         006F
                         006F    ;+
                         006F    ; Initialize the DAP control block and the transmit and receive buffers in the
                         006F    ; NWA. These buffers will be used to exchange DAP Configuration messages, then
                         006F    ; they may be replaced by larger DAP buffers if the negotiated DAP buffer size
                         006F    ; is larger than NWA$C_BUFFERSIZ. Note that the transmit buffer is used for
                         006F    ; both building a new DAP message (BLD descriptor) and for concatenating DAP
                         006F    ; messages before sending them to FAL (XMT descriptor).
```

```
                              006F  ;-
                              006F
                              006F  EXCH_INIT:                                      ; Initialize control block and buffers
                              006F          $ZERO_FILL-                             ; Zero DAP control block
                              006F                  DST=(R7)-                       ;
                              006F                  SIZE=#DAP$C_BLN                 ;
                              0077
                              0077          ASSUME  NWA$C_BUFFERSIZ GE <512+4+28>
                              0077          ASSUME  NWA$Q_XMT EQ NWA$Q_RCV+8
                              0077
      50   00E0 C7     7E     0077          MOVAQ   NWA$Q_RCV(R7),R0                ; Get start address of descriptors
                80     D4     007C          CLRL    (R0)+                           ; Initialize receive descriptor
      80   01A0 C7     9E     007E          MOVAB   NWA$T_RCVBUF(R7),(R0)+          ;
                80     D4     0083          CLRL    (R0)+                           ; Initialize transmit descriptor
      80   03C0 C7     9E     0085          MOVAB   NWA$T_XMTBUF(R7),(R0)+          ;
           0220 8F     B0     008A          MOVW    #NWA$C_BUFFERSIZ,-              ; Make the preallocated buffer size
           00CA C7            008E                  NWA$W_DAPBUFSIZ(R7)             ;  the current DAP buffer size
                              0091
                              0091  ;+
                              0091  ; Build and send DAP Configuration message to partner.
                              0091  ;-
                              0091
                              0091  SEND_CNF:                                       ; (required message)
                              0091          $SETBIT #NWA$V_LAST_MSG,(R7)            ; Declare this last message to block
           50    01    D0     0095          MOVL    #DAP$K_CNF_MSG,R0               ; Get message type value
              FF65'    30     0098          BSBW    NT$BUILD_HEAD                   ; Construct message header
           85    56    B0     009B          MOVW    R6,(R5)+                        ; Store BUFSIZ field (desired value)
           85    07    90     009E          MOVB    #DAP$K_VAXVMS,(R5)+             ; Store OSTYPE field
           85    03    90     00A1          MOVB    #DAP$K_RMS32,(R5)+              ; Store FILESYS field
           85    07    90     00A4          MOVB    #DAP$K_VERNUM_V,(R5)+           ; Store VERNUM field
           85    00    90     00A7          MOVB    #DAP$K_ECONUM_V,(R5)+           ; Store ECONUM field
           85    00    90     00AA          MOVB    #DAP$K_USRNUM_V,(R5)+           ; Store USRNUM field
           85    04    90     00AD          MOVB    #DAP$K_DECVER_V,(R5)+           ; Store DECVER field
           85    00    90     00B0          MOVB    #DAP$K_USRVER_V,(R5)+           ; Store USRVER field
 51   EFF67DF7 8F      D0     00B3          MOVL    #DAP$K_SYSCAP1_V,R1             ; Get VAX supported capabilities
 52   00001962 8F      D0     00BA          MOVL    #DAP$K_SYSCAP2_V,R2             ;  quadword mask
              FF3C'    30     00C1          BSBW    NT$CVT_BN8_EXT                  ; Store SYSCAP as an extensible field
              FF39'    30     00C4          BSBW    NT$BUILD_TAIL                   ; Finish building message
              FF36'    30     00C7          BSBW    NT$TRANSMIT                     ; Send Configuration message to FAL
              03 50    E8     00CA          BLBS    R0,RECV_CNF                     ; Branch on success
              009A     31     00CD  ERROR1: BRW     ERROR                           ; Branch on failure
                              00D0
                              00D0  ;+
                              00D0  ; Receive DAP Configuration message response from partner.
                              00D0  ;-
                              00D0
                              00D0  RECV_CNF:                                       ; (required message)
                              00D0          $SETBIT #DAP$K_CNF_MSG,DAP$L_MSG_MASK(R7)
                              00D5                                                  ; Expect response of configuration msg
              FF28'    30     00D5          BSBW    NT$RECEIVE                      ; Get reply from FAL
              F2 50    E9     00D8          BLBC    R0,ERROR1                       ; Branch on failure
              42 A7    90     00DB          MOVB    DAP$B_OSTYPE(R7),-              ; Save OSTYPE field in NWA
           00C4 C7            00DE                  NWA$B_OSTYPE(R7)                ;
              43 A7    90     00E1          MOVB    DAP$B_FILESYS(R7),-             ; Save FILESYS field in NWA
           00C5 C7            00E4                  NWA$B_FILESYS(R7)               ;
                              00E7
                              00E7  ;+
                    00E7   ; Determine the 'agreed upon' DAP buffer size to use and save this value.
                    00E7   ; It is the smaller of partner's buffer size and our requested buffer size.
                    00E7   ;-
                    00E7
  00CA C7   56   B0 00E7          MOVW    R6,NWA$W_DAPBUFSIZ(R7)   ; Assume we'll use requested buffer size
        40 A7    B5 00EC          TSTW    DAP$W_BUFSIZ(R7)         ; Use our buffer size if partner
           0C    13 00EF          BEQL    10$                      ;  has unlimited buffer space
   56   40 A7    B1 00F1          CMPW    DAP$W_BUFSIZ(R7),R6      ; Use our buffer size if partner
           06    1E 00F5          BGEQU   10$                      ;  has buffer size GEQ ours
        40 A7    B0 00F7          MOVW    DAP$W_BUFSIZ(R7),-       ; Use partner's buffer size which is
      00CA C7       00FA                  NWA$W_DAPBUFSIZ(R7)      ;  smaller than ours
                    00FD
                    00FD   ;+
                    00FD   ; Allocate big DAP buffers if standard size buffers already allocated as part of
                    00FD   ; the NWA are not sufficient. Note that the transmit buffer will be twice the
                    00FD   ; size of the receive buffer (or twice NWA$W_DAPBUFSIZ) as it is used for both
                    00FD   ; building new DAP messages and for concatenating DAP messges before sending
                    00FD   ; them to FAL. The overflow buffer space may be used when a new message is being
                    00FD   ; constructed and there are messages already blocked in the transmit buffer.
                    00FD   ;-
                    00FD
 56   00CA C7   3C 00FD   10$:   MOVZWL  NWA$W_DAPBUFSIZ(R7),R6   ; Get negotiated DAP buffer size
 0220 8F   56   B1 0102          CMPW    R6,#NWA$C_BUFFERSIZ      ; Use standard buffers if they are large
           22   1B 0107          BLEQU   20$                      ;  enough
      56   07   C0 0109          ADDL2   #7,R6                    ; Round up buffer size to quadword
      56   07   CA 010C          BICL2   #7,R6                    ;  boundary
 52   56   03   C5 010F          MULL3   #3,R6,R2                 ; Compute size of desired receive buffer
                    0113                                           ;  plus a double-length transmit buffer
        FEEA'   30 0113          BSBW    RM$GETPAG                ; Allocate space (NOT ZEROED)
        51 50   E9 0116          BLBC    R0,ERROR                 ; Branch on failure
 0170 C7   52   7D 0119          MOVQ    R2,NWA$Q_BIGBUF(R7)      ; Update big buffer descriptor
 00E4 C7   53   D0 011E          MOVL    R3,NWA$Q_RCV+4(R7)       ; Update receive descriptor
      53   56   C0 0123          ADDL2   R6,R3                    ; Move pointer to next buffer
 00EC C7   53   D0 0126          MOVL    R3,NWA$Q_XMT+4(R7)       ; Update transmit descriptor
                    012B
                    012B   ;+
                    012B   ; Determine the maximum record size that can be supported for network access
                    012B   ; in record I/O operations. This is accomplished by examining the negotiated
                    012B   ; DAP buffer size and then updating the device buffer size value in the IFAB
                    012B   ; (if appropriate from its initial setting in NT$MOD_DEV_CHAR).
                    012B   ;
                    012B   ; The value in IFB$L_DEVBUFSIZ establishes the network record size limit as
                    012B   ; this value is used by RMS at $CONNECT time to allocate the BDB buffer. The
                    012B   ; size of this buffer determines the largest record that can be moved to/from
                    012B   ; user's buffer during $GET, $PUT, and $UPDATE operations on a remote file.
                    012B   ;
                    012B   ; The algorithm establishes a maximum record size that is equal to 1 to 64 pages
                    012B   ; of memory (i.e., 512, 1024, ..., 32768 bytes).
                    012B   ;
                    012B   ; Note that IFB$L_DEVBUFSIZ does not limit the size of a user block I/O request
                    012B   ; which can be from 1 to 127 blocks.
                    012B   ;-
                    012B
   56   00CA C7   3C 012B   20$:   MOVZWL  NWA$W_DAPBUFSIZ(R7),R6   ; Get negotiated DAP buffer size
        56   08   C2 0130          SUBL2   #8,R6                    ; Subtract DAP DATA message overhead
  56   56   F7 8F  78 0133          ASHL    #-9,R6,R6                ; Compute # whole pages
             0F   13 0138          BEQL    FINISH                   ; Keep initial value if DAPBUFSIZ < 520
        40 8F    56   91  013A           CMPB    R6,#64                       ; Limit value to 64 pages as the largest
                 04   1B  013E           BLEQU   30$                          ;  record defined by RMS is slightly
        56    40 8F   9A  0140           MOVZBL  #64,R6                       ;  less than 32K bytes
  48 A9    56    09   78  0144     30$:  ASHL    #9,R6,IFB$L_DEVBUFSIZ(R9);Compute largest record size supported
                          0149
                          0149
                          0149     ;+
                          0149     ; While we have both a FWA and a NWA, move the nodename (sans delimiters or
                          0149     ; access strings) and the nodename size to NWA$T_NODEBUF & NWA$B_NODBUFSIZ
                          0149     ; for use by NT$CRC_LOGERR.
                          0149     ;-
                          0149
                          0149     FINISH:                                    ; Miscellaneous
        53    07E9 CA     9E  0149           MOVAB   FWA$T_NODEBUF(R10),R3        ; Get address of nodename (spec list)
        63    07   22     3A  014E           LOCC    #^A/"/,#FWA$C_MAXNODNAM+1,(R3) ; Search for quote
                   04     12  0152           BNEQ    10$                          ; Branch if access control string
        63    07   3A     3A  0154           LOCC    #^A/:/,#FWA$C_MAXNODNAM+1,(R3) ; Find the colon (must be there)
        52    51   53     C3  0158     10$:  SUBL3   R3,R1,R2                     ; Compute the nodename length
   0169 C7    63   52     28  015C           MOVC3   R2,(R3),NWA$T_NODEBUF(R7);Move nodename to NWA
   0168 C7    52          90  0162           MOVB    R2,NWA$B_NODBUFSIZ(R7)       ; Move length to NWA
                              0167           RMSSUC                               ; Return success
                          05  016A     ERROR: RSB                                 ; Exit with RMS code in R0
```

```
016B          .SBTTL  NT$GET_FILESPEC - BUILDS A FILESPEC
016B
016B   ;++
016B   ; NT$GET_FILESPEC - builds a filespec (less primary node name) from its
016B   ;       constituent parts and stores it as a counted ASCII string.
016B   ;
016B   ; Calling Sequence:
016B   ;
016B   ;       BSBW    NT$GET_FILESPEC
016B   ;
016B   ; Input Parameters:
016B   ;
016B   ;       R5      Address of buffer to receive counted ASCII string
016B   ;       R7      NWA address
016B   ;       R8      FAB address
016B   ;       R9      IFAB address
016B   ;       R10     FWA address
016B   ;       R11     Impure Area address
016B   ;
016B   ; Implicit Inputs:
016B   ;
016B   ;       FWA$B_DIRTERM
016B   ;       FWA$Q_DEVICE
016B   ;       FWA$Q_DIR1
016B   ;       FWA$Q_DIR2
016B   ;       FWA$Q_DIR2+8 thru FWA$Q_DIR2+48
016B   ;       FWA$Q_NAME
016B   ;       FWA$Q_QUOTED
016B   ;       FWA$Q_VERSION
016B   ;       FWA$V_DEVICE
016B   ;       FWA$V_EXP_VER
016B   ;       FWA$V_GRPMBR
016B   ;       FWA$V_DIR
016B   ;       FWA$V_DIR_LVLS
016B   ;       FWA$V_QUOTED
016B   ;       NWA$B_OSTYPE
016B   ;
016B   ; Output Parameters:
016B   ;
016B   ;       R0-R3   Destroyed
016B   ;       R5      Updated buffer pointer (address of end of string + 1)
016B   ;       AP      Destroyed
016B   ;
016B   ; Implicit Outputs:
016B   ;
016B   ;       None
016B   ;
016B   ; Completion Codes:
016B   ;
016B   ;       None
016B   ;
016B   ; Side Effects:
016B   ;
016B   ;       None
016B   ;
016B   ;--
016B
```

```
                         0168   NT$GET_FILESPEC::                                  ; Entry point
        0150 8F    BB    016B           PUSHR   #^M<R4,R6,R8>                      ; Save registers
          53 55    D0    016F           MOVL    R5,R3                              ; Copy next byte pointer
             83    94    0172           CLRB    (R3)+                              ; Skip over count byte
          58 53    D0    0174           MOVL    R3,R8                              ; Save pointer to start of DST string
                         0177
                         0177   ;
                         0177   ; Process secondary node spec strings.
                         0177   ;
                         0177
           2F AA   95    0177           TSTB    FWA$B_SUBNODCNT(R10)               ; Branch if there is only one
              24   13    017A           BEQL    10$                                ;  node spec in node spec list
     50  01B4 CA   3C    017C           MOVZWL  FWA$Q_NODE1(R10),R0                ; Get size of primary node spec
 51  50  00D8 CA   A3    0181           SUBW3   R0,FWA$Q_NODE(R10),R1              ; Compute descriptor of concatenated
 52  50  00DC CA   C1    0187           ADDL3   R0,FWA$Q_NODE+4(R10),R2            ;  secondary node spec strings
     63  62   51   28    018D           MOVC3   R1,(R2),(R3)                       ; Copy secondary node specs
                         0191
                         0191   ;
                         0191   ; Process quoted string.
                         0191   ;
                         0191   ; Note: If there is only a primary node spec, then the quoted string is copied
                         0191   ;       with the quote delimiters removed. Conversely, if secondary node specs
                         0191   ;       are present, then the quoted string is copied with the quote delimiters
                         0191   ;       intact.
                         0191   ;
                         0191
        04 6A   35   E0  0191           BBS     #FWA$V_REMRESULT,(R10),5$; Branch if result already delivered.
        22 6A   1A   E1  0195           BBC     #FWA$V_QUOTED,(R10),30$ ; Branch if no quoted string follows
     50    0170 CA   7D  0199   5$:     MOVQ    FWA$Q_QUOTED(R10),R0               ; Get descriptor of quoted string
                         019E                                                      ;  (including quote delimiters)
              14    11   019E           BRB     20$                                ; Join common code
        F5 6A   35   E0  01A0   10$:    BBS     #FWA$V_REMRESULT,(R10),5$; Branch if result already delivered.
        13 6A   1A   E1  01A4           BBC     #FWA$V_QUOTED,(R10),30$ ; Branch if no quoted string follows
 50  0170 CA    02   C3  01A8           SUBL3   #2,FWA$Q_QUOTED(R10),R0            ; Get size of string less quotes
 51  0174 CA    01   C1  01AE           ADDL3   #1,FWA$Q_QUOTED+4(R10),R1;Get address of string
     63  61     50   28  01B4   20$:    MOVC3   R0,(R1),(R3)                       ; Copy quoted string
            009B    31   01B8           BRW     120$                               ; Join common code
                         01BB
                         01BB   ;
                         01BB   ; Process device name.
                         01BB   ;
                         01BB
        0B 6A   0F   E1  01BB   30$:    BBC     #FWA$V_DEVICE,(R10),40$ ; Branch if no device name present
         00E0 CA    28   01BF           MOVC3   FWA$Q_DEVICE(R10),-                ; Copy device name
     63  00E4 DA         01C3                   @FWA$Q_DEVICE+4(R10),(R3)
          83  3A    90   01C7           MOVB    #^A\:\,(R3)+                       ; Append delimiter
                         01CA
                         01CA   ;
                         01CA   ; Process directory list.
                         01CA   ; It is either in the [group,member] or [directory_name_list] format.
                         01CA   ;
                         01CA
        4A 6A   0E   E1  01CA   40$:    BBC     #FWA$V_DIR,(R10),90$               ; Branch if no directory present
        0A AA   02   83  01CE           SUBB3   #2,FWA$B_DIRTERM(R10),-            ; Store left bracket ('[' or '<')
                83       01D2                   (R3)+                              ;  (ASCII code is right bracket + 2)
        2A 6A   1B   E0  01D3           BBS     #FWA$V_GRPMBR,(R10),70$ ; Branch if [group,member] format
     56    0130 CA   7E  01D7           MOVAQ   FWA$Q_DIR1(R10),R6                 ; Get address of directory descriptor
```

```
          1D   EF  01DC   550          EXTZV    #FWA$V_DIR_LVLS,-           ; Get # of directory sub-levels
  5C  6A  03       01DE   551                   #FWA$S_DIR_LVLS,(R10),AP;    (0 means UFD level only)
      50  86   D0  01E1   552 50$:     MOVL     (R6)+,R0                    ; Get size of string
  63  96  50   28  01E4   553          MOVC3    R0,@(R6)+,(R3)              ; Copy next (sub)directory name
      83  2E   90  01E8   554          MOVB     #^A\.\,(R3)+                ; Copy directory separator
05 F8 A6  10   E1  01EB   555          BBC      #FSCB$V_ELIPS,-8(R6),60$          ; Branch if elipsis does not follow
                   01F0   556                                              ;  (sub)directory name
  83  2E2E 8F  B0  01F0   557          MOVW     #^A\..\,(R3)+               ; Append two dots to make an elipsis
       E9 5C   F4  01F5   558 60$:     SOBGEQ   AP,50$                      ; Branch if more directory names
17 F8 A6  10   E0  01F8   559          BBS      #FSCB$V_ELIPS,-8(R6),80$          ; Branch if we just copied an elipsi
          53   D7  01FD   560          DECL     R3                          ; Otherwise, remove unwanted (single)
          13   11  01FF   561          BRB      80$                         ;  trailing dot
    0130  CA   28  0201   562 70$:     MOVC3    FWA$Q_DIR1(R10),-           ; Copy group directory field
  63 0134 DA       0205   563                   @FWA$Q_DIR1+4(R10),(R3) ;
      83  2C   90  0209   564          MOVB     #^A\.\,(R3)+                ; Copy directory separator
    0138  CA   28  020C   565          MOVC3    FWA$Q_DIR2(R10),-           ; Copy member directory field
  63 013C DA       0210   566                   @FWA$Q_DIR2+4(R10),(R3) ;
   83  0A AA   90  0214   567 80$:     MOVB     FWA$B_DIRTERM(R10),(R3)+; Store right bracket (']' or '>')
                   0218   568
                   0218   569 ;
                   0218   570 ; Process file name, file type, and file version.
                   0218   571 ; To facilitate communication with non-VMS systems, several system specific
                   0218   572 ; version number checks will be made.
                   0218   573 ;
                   0218   574 ; Note: The file name string described by FWA$Q_NAME is guaranteed to contain
                   0218   575 ;       both the "." and ";" delimiters, even if the user did not specify a
                   0218   576 ;       file type or file version number. Furthermore, a "." version number
                   0218   577 ;       delimiter entered by the user will have been converted to a ";"
                   0218   578 ;       delimiter by RMOXPFN!
                   0218   579 ;
                   0218   580
    0170  CA   28  0218   581 90$:     MOVC3    FWA$Q_NAME(R10),-           ; Copy file name string (assembled
  63 0174 DA       021C   582                   @FWA$Q_NAME+4(R10),(R3) ;  into one string by RMOXPFN)
  32 67   34   E0  0220   583          BBS      #DAP$V_VAXVMS,(R7),120$ ; Branch if partner is VAX/VMS
                   0224   584
                   0224   585 ;
                   0224   586 ; If the remote node is not VMS, delete the trailing semi-colon (null version
                   0224   587 ; number) if the user did not explicitly enter a version # in the primary
                   0224   588 ; filespec string.
                   0224   589 ;
                   0224   590
   0E 6A  10   E0  0224   591          BBS      #FWA$V_EXP_VER,(R10),100$;Branch if version # was explicit
   3B  FF A3   91  0228   592          CMPB     -1(R3),#^A\;\              ; Is last character a semi-colon?
          08   12  022C   593          BNEQ     100$                        ; Branch if not
          53   D7  022E   594          DECL     R3                          ; Otherwise delete it here and from
    0170  CA   B7  0230   595          DECW     FWA$Q_NAME(R10)             ;  filename descriptor in FWA
          20   11  0234   596          BRB      120$                        ; All done
                   0236   597
                   0236   598 ;
                   0236   599 ; If the remote node is RT-11, remove the version number substring (either
                   0236   600 ; ";" or ";ver") because RT-11 does not recognize the version number element.
                   0236   601 ;
                   0236   602
  50 0180 CA   3C  0236   603 100$:    MOVZWL   FWA$Q_VERSION(R10),R0       ; Get number of digits in version #
          50   D6  023B   604          INCL     R0                          ; Add semi-colon delimiter to count
   0A 67  38   E1  023D   605          BBC      #DAP$V_RT11,(R7),110$       ; Branch if remote node is not RT-11
      53  50   C2  0241   606          SUBL2    R0,R3                       ; Delete version number substring here
```

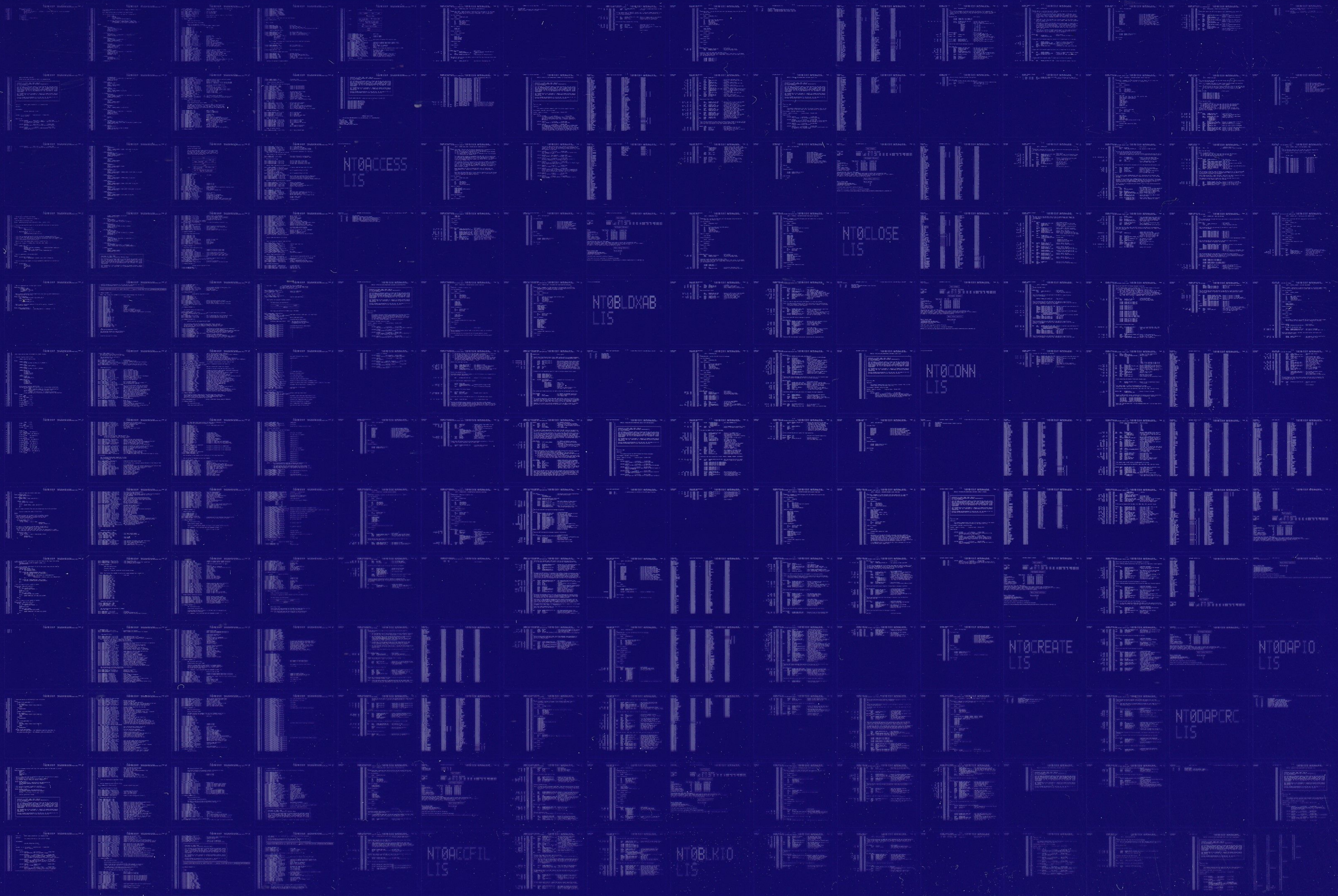
```
0170 CA    50    A2  0244   607             SUBW2   R0,FWA$Q_NAME(R10)         ;  and from filename descriptor in FWA
           0B    11  0249   608             BRB     120$                       ; All done
                     024B   609
                     024B   610 ;
                     024B   611 ; If the remote node is TOPS-20, convert the ";" version number delimiter to a
                     024B   612 ; "." delimiter because TOPS-20 requires uses the semi-colon character as a
                     024B   613 ; file attribute delimiter.
                     024B   614 ;
                     024B   615
  07 67    37    E1  024B   616 110$:       BBC     #DAP$V_TOPS20,(R7),120$    ; Branch if remote node is not TOPS-20
51    53   50    C3  024F   617             SUBL3   R0,R3,R1                   ; Calculate address of delimiter
      61   2E    90  0253   618             MOVB    #^A\.\,(R1)                ; Convert period to semi-colon
                     0256   619
                     0256   620 ;
                     0256   621 ; Finish building counted ASCII string.
                     0256   622 ;
                     0256   623
50    53   58    C3  0256   624 120$:       SUBL3   R8,R3,R0                   ; Calculate size of string
  FF A8    50    90  025A   625             MOVB    R0,-1(R8)                  ; Store the count
      55   53    D0  025E   626             MOVL    R3,R5                      ; Put next byte pointer in proper reg
    0150 8F      BA  0261   627             POPR    #^M<R4,R6,R8>              ; Restore registers
                 05  0265   628             RSB                                ; Exit
                     0266   629
                     0266   630             .END                               ; End of module
```

```
$$.PSECT_EP                    = 00000000       DAP$M_TMP2$                   = 0000FC00
$$COUNT                        = 00000006       DAP$Q_DCODE_FLG                 00000000
$$RMSTEST                      = 0000001A       DAP$Q_FILESPEC                  00000044
$$RMS_PBUGCHK                  = 00000010       DAP$Q_MSG_BUF1                  00000008
$$RMS_TBUGCHK                  = 00000008       DAP$Q_MSG_BUF2                  00000010
$$RMS_UMODE                    = 00000004       DAP$Q_PASSWORD                  00000050
DAP$B_ACCFUNC                    00000040       DAP$Q_SYSCAP                    00000028
DAP$B_ACCOPT                     00000041       DAP$Q_SYSPEC                    00000038
DAP$B_BITCNT                     00000035       DAP$V_RT11                    = 00000038
DAP$B_DCODE_FID                  00000019       DAP$V_TOPS20                  = 00000037
DAP$B_DCODE_MAC                  0000001B       DAP$V_VAXVMS                  = 00000034
DAP$B_DCODE_MSG                  0000001A       DAP$W_BUFSIZ                    00000040
DAP$B_DECVER                     00000047       DAP$W_DISPLAY1                  0000004C
DAP$B_ECONUM                     00000045       DAP$W_PARTNER                   00000006
DAP$B_FAC                        00000042       DAP$W_VERSION                   00000004
DAP$B_FILESYS                    00000043       ERASE_RENAME                    0000002D R      01
DAP$B_FLAGS                      00000031       ERROR                           0000016A R      01
DAP$B_LEN256                     00000034       ERROR1                          000000CD R      01
DAP$B_LENGTH                     00000033       EXCH_COMMON                     00000055 R      01
DAP$B_OSTYPE                     00000042       EXCH_INIT                       0000006F R      01
DAP$B_SHR                        00000043       FINISH                          00000149 R      01
DAP$B_STREAMID                   00000032       FSCB$V_ELIPS                  = 00000010
DAP$B_TYPE                       00000030       FWA$B_DIRTERM                 = 0000000A
DAP$B_USRNUM                     00000046       FWA$B_SUBNODCNT               = 0000002F
DAP$B_USRVER                     00000048       FWA$C_MAXNODNAM               = 00000006
DAP$B_VERNUM                     00000044       FWA$Q_DEVICE                  = 000000E0
DAP$B_X_FIELD                    00000024       FWA$Q_DIR1                    = 00000130
DAP$C_BLN                        000000C0       FWA$Q_DIR2                    = 00000138
DAP$K_BLN                        000000C0       FWA$Q_NAME                    = 00000170
DAP$K_CNF_MSG                  = 00000001       FWA$Q_NODE                    = 000000D8
DAP$K_CREATE                   = 00000002       FWA$Q_NODE1                   = 000001B4
DAP$K_DECVER_V                 = 00000004       FWA$Q_QUOTED                  = 00000170
DAP$K_DIR_LIST                 = 00000006       FWA$Q_VERSION                 = 00000180
DAP$K_ECONUM_V                 = 00000000       FWA$S_DIR_LVLS                = 00000003
DAP$K_ERASE                    = 00000004       FWA$T_NODEBUF                 = 000007E9
DAP$K_OPEN                     = 00000001       FWA$V_DEVICE                  = 0000000F
DAP$K_RENAME                   = 00000003       FWA$V_DIR                     = 0000000E
DAP$K_RMS32                    = 00000003       FWA$V_DIR_LVLS                = 0000001D
DAP$K_SYSCAP1_V                = EFF67DF7       FWA$V_EXP_VER                 = 00000010
DAP$K_SYSCAP2_V                = 00001962       FWA$V_GRPMBR                  = 0000001B
DAP$K_USRNUM_V                 = 00000000       FWA$V_QUOTED                  = 0000001A
DAP$K_USRVER_V                 = 00000000       FWA$V_REMRESULT               = 00000035
DAP$K_VAXVMS                   = 00000007       IFB$L_DEVBUFSIZ               = 00000048
DAP$K_VERNUM_V                 = 00000007       NT$BUILD_HEAD                   ********   X    01
DAP$L_CHWA                       00000030       NT$BUILD_TAIL                   ********   X    01
DAP$L_CRC_RSLT                   00000020       NT$CVT_BN8_EXT                  ********   X    01
DAP$L_DCODE_STS                  00000018       NT$EXCH_CNF                     00000000 RG     01
DAP$L_MSG_MASK                   0000001C       NT$GET_FILESPEC                 0000016B RG     01
DAP$L_SSPWA                      00000080       NT$RECEIVE                      ********   X    01
DAP$L_TEMP                       00000090       NT$TRANSMIT                     ********   X    01
DAP$M_BITCNT                   = 00000008       NWA$B_ALLXABCNT                 0000011C
DAP$M_DSP_3NAM                 = 00000200       NWA$B_DAP_RAC                   000000C9
DAP$M_GET                      = 00000002       NWA$B_FILESYS                   000000C5
DAP$M_GO_NOGO                  = 00000010       NWA$B_KEYXABCNT                 0000011D
DAP$M_MSE                      = 00000010       NWA$B_NETSTRSIZ                 0000016F
DAP$M_SEGMENT                  = 00000040       NWA$B_NODBUFSIZ                 00000168
DAP$M_TMP1$                    = 000000C0       NWA$B_ORG                       000000C6
```

```
NWA$B_OSTYPE                      000000C4              NWA$W_DISPLAY                  000000D0
NWA$B_RFM                         000000C7              NWA$W_FIL_OFF                  000000CE
NWA$B_RMS_RAC                     000000C8              NWA$W_JNLXABJOP                0000011E
NWA$C_BLN                         00000800              OPEN_CREATE                    00000042 R     01
NWA$C_BUFFERSIZ                 = 00000220              PIO$GB_DFNBC                   ********   X   01
NWA$K_BLN                         00000800              RECV_CNF                       000000D0 R     01
NWA$L_ALLXABADR                   00000100              RMS$GETPAG                     ********   X   01
NWA$L_DATXABADR                   00000104              SEARCH                         00000032 R     01
NWA$L_DEV                         000000C0              SEND_CNF                       00000091 R     01
NWA$L_FHCXABADR                   00000108              SYS$GB_DFNBC                   ********   X   01
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_LAST_MSG                  = 00000000
NWA$W_BUILD                       000000D2
NWA$W_DAPBUFSIZ                   000000CA
NWA$W_DIR_OFF                     000000CC
```

```
                                +----------------+
                                ! 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                      00000266  (   614.)  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                  30     00:00:00.06    00:00:00.63
Command processing             141     00:00:00.62    00:00:04.20
Pass 1                         342     00:00:12.65    00:00:38.01
Symbol table sort                0     00:00:01.72    00:00:02.40
Pass 2                         124     00:00:02.65    00:00:06.58
Symbol table output             22     00:00:00.16    00:00:00.33
Psect synopsis output            2     00:00:00.03    00:00:00.02
Cross-reference output           0     00:00:00.00    00:00:00.00
Assembler run totals           663     00:00:17.90    00:00:52.19

The working set limit was 1500 pages.
67627 bytes (133 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1186 non-local and 24 local symbols.
630 source lines were read in Pass 1, producing 14 object records in Pass 2.
24 pages of virtual memory were used to define 23 macros.

                             +--------------------------+
                             ! Macro library statistics !
                             +--------------------------+

Macro library name                         Macros defined
------------------                         --------------
_$255$DUA28:[RMS.OBJ]RMS.MLB;1                   15
_$255$DUA28:[SYSLIB]STARLET.MLB;2                 4
TOTALS (all libraries)                           19

1400 GETS were required to define 19 macros.

There were no errors, warnings or information messages.

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

0315 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY



NT0ACCESS
LIS

NT0CLOSE
LIS

NT0BLDXAB
LIS

NT0CONN
LIS

NT0CREATE
LIS

NT0DAPIO
LIS

NT0DAPCRC
LIS

NT0ACCFIL
LIS

NT0BLKIO
LIS