


```

NN      NN      TTTTTTTTTT      000000      PPPPPPPP      UU      UU      TTTTTTTTTT
NN      NN      TTTTTTTTTT      000000      PPPPPPPP      UU      UU      TTTTTTTTTT
NN      NN      TT              00          00      PP          PP      UU      UU      TT
NN      NN      TT              00          00      PP          PP      UU      UU      TT
NNNN    NN      TT              00          0000    PP          PP      UU      UU      TT
NNNN    NN      TT              00          0000    PP          PP      UU      UU      TT
NN      NN      NN      TT          00      00      00      PPPPPPPP      UU      UU      TT
NN      NN      NN      TT          00      00      00      PPPPPPPP      UU      UU      TT
NN      NNNN     TT              0000         00      PP          UU      UU      TT
NN      NNNN     TT              0000         00      PP          UU      UU      TT
NN      NN      TT              00          00      PP          UU      UU      TT
NN      NN      TT              00          00      PP          UU      UU      TT
NN      NN      TT              000000         00      PP          UUUUUUUUUU      TT
NN      NN      TT              000000         00      PP          UUUUUUUUUU      TT

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II         SS
LL      II         SS
LL      II         SS
LL      II         SS
LL      II         SSSSSS
LL      II         SSSSSS
LL      II         SS
LL      II         SS
LL      II         SS
LL      II         SS
LLLLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLLLL IIIIII SSSSSSSS

```

(2)	64
(3)	126
(4)	411
(5)	495

DECLARATIONS
NT\$PUT - PERFORM NETWORK PUT FUNCTION
NT\$UPDATE - PERFORM NETWORK UPDATE FUNCTION
NT\$DELETE - PERFORM NETWORK DELETE FUNCTION


```

0000 64      .SBTTL  DECLARATIONS
0000 65
0000 66      :
0000 67      : Include Files:
0000 68      :
0000 69
0000 70      $BDBDEF      : Define BDB symbols
0000 71      $DAPPLGDEF   : Define DAP prologue symbols
0000 72      $DAPHDRDEF   : Define DAP message header
0000 73      $DAPCNFDEF   : Define DAP Configuration message
0000 74      $DAPCTLDEF   : Define DAP Control message
0000 75      $DAPDATDEF   : Define DAP Data message
0000 76      $FABDEF      : Define File Access Block symbols
0000 77      $IFBDEF      : Define IFAB symbols
0000 78      $IRBDEF      : Define IRAB symbols
0000 79      $NWADEF      : Define Network Work Area symbols
0000 80      $RABDEF      : Define Record Access Block symbols
0000 81      $RMSDEF      : Define RMS completion codes
0000 82
0000 83      :
0000 84      : Macros:
0000 85      :
0000 86      :     None
0000 87      :
0000 88      : Equated Symbols:
0000 89      :
0000 90
0000000A 0000 91 LF=10      : Line feed
0000000B 0000 92 VT=11      : Vertical tab
0000000C 0000 93 FF=12      : form feed
00000010 0000 94 DLE=16     : Data link error (Control-P)
00000011 0000 95 DC1=17     : Device control 1 (Control-Q)
00000012 0000 96 DC2=18     : Device control 2 (Control-R)
00000013 0000 97 DC3=19     : Device control 3 (Control-S)
00000014 0000 98 DC4=20     : Device control 4 (Control-T)
0000001A 0000 99 CTRLZ=26   : Control-Z
0000001B 0000 100 ESC=27    : Escape
0000 101
0000A0D 0000 102 CRLF=^X0A0D : ASCII codes for CR and LF
0000000A 0000 103 LF=^X0A    : ASCII code for LF
0000000C 0000 104 CR=^X0D    : ASCII code for CR
0000 105
0000 106      ASSUME  DAP$Q_DCODE_FLG EQ 0
0000 107      ASSUME  NWA$Q_FLG EQ 0
0000 108
0000 109      :
0000 110      : Own Storage:
0000 111      :
0000 112      :
0000 113      :
0000 114      :
0000 115      : The following bitmask represents the DAP terminator set for stream (STM)
0000 116      : format files where each bit position corresponds to an ASCII character code
0000 117      : for a single-character terminator. Note that the default terminator, CRLF,
0000 118      : is not expressed in this mask.
0000 119      :
0000 120

```

```
0C1F1C00 0000 121 NT$STM_MASK:: ; DAP stream terminator bitmask  
          0000 122          .LONG <<1@LF> + <1@FF> + <1@VT> + <1@CTRLZ> + <1@ESC> + -  
          0004 123          <1@DLE> + <1@DC1> + <1@DC2> + <1@DC3> + <1@DC4>>  
          0004 124
```

```

0004 126      .SBTTL  NT$PUT - PERFORM NETWORK PUT FUNCTION
0004 127
0004 128      :++
0004 129      : NT$PUT - engages in a DAP dialogue with the remote FAL to put (store) the
0004 130      : specified record of a sequential, relative, or indexed file. This
0004 131      : routine supports both DAP file transfer and record transfer modes.
0004 132
0004 133      : Calling Sequence:
0004 134
0004 135      :     BSBW  NT$PUT
0004 136
0004 137      : Input Parameters:
0004 138
0004 139      :     R4    BDB address
0004 140      :     R8    RAB address
0004 141      :     R9    IRAB address
0004 142      :     R10   IFAB address
0004 143      :     R11   Impure Area address
0004 144
0004 145      : Implicit Inputs:
0004 146
0004 147      :     BDB buffer contents
0004 148      :     BDB$B_ADDR
0004 149      :     BDB$W_NUMB
0004 150      :     DAP$V_DAPCRC
0004 151      :     DAP$V_GEQ_V56
0004 152      :     DAP$B_CRC_RSLT
0004 153      :     IFB$B_NWA_PTR
0004 154      :     IFB$V_SQO
0004 155      :     IRB$B_MODE
0004 156      :     NWA$V_FTM_INIT
0004 157      :     NWA$V_FTM_RETRV
0004 158      :     NWA$W_DAPBUFSIZ
0004 159      :     NWA$B_RFM
0004 160      :     NWA$B_ORG
0004 161      :     NWA$Q_BLD
0004 162      :     RAB$B_KSZ
0004 163      :     RAB$B_RAC
0004 164      :     RAB$B_ROP
0004 165      :     RAB$B_KBF
0004 166
0004 167      : Output Parameters:
0004 168
0004 169      :     R0    Status code (RMS)
0004 170      :     R1-R3 Destroyed
0004 171      :     AP    Destroyed
0004 172
0004 173      : Implicit Outputs:
0004 174
0004 175      :     BDB$B_FLGS
0004 176      :     DAP$B_CRC_RSLT
0004 177      :     IRB$B_IOS and IRB$B_IOS4 zeroed
0004 178      :     NWA$V_FTM_INIT cleared
0004 179      :     NWA$V_FTM_STORE
0004 180      :     RAB$B_BKI
0004 181      :     RAB$W_RFA
0004 182

```



```

0004 183 : Completion Codes:
0004 184 :
0004 185 :         Standard RMS completion codes
0004 186 :
0004 187 : Side Effects:
0004 188 :
0004 189 :         None
0004 190 :
0004 191 : --
0004 192 :
0004 193 NT$PUT:: : Entry point
0004 194 $STSTPT NTPUT :
000A 195 PUSHR #M<R4,R5,R6,R7> : Save registers
56 54 DO 000E 196 MOVL R4,R6 : Copy BDB address
57 3C AA DO 0011 197 MOVL IFBSL_NWA_PTR(R10),R7 : Get address of NWA (and DAP)
    FFE8' 30 0015 198 BSBW NT$CHR_RAC : Validate record access mode
    10 50 E9 0018 199 BLBC R0,10$ : Branch on failure
OF 67 EO 001B 200 BBS #NWSV_FTM_RETRV,(R7),20$ : $PUT after $GET not allowed in FTM
001F 201
001F 202 ASSUME RAB$C_SEQ EQ 0
001F 203 ASSUME RAB$C_KEY EQ 1
001F 204 ASSUME RAB$C_RFA EQ 2
001F 205
001F 206 $CASEB SELECTOR=RAB$B_RAC(R8)- : Dispatch on access mode:
001F 207 DISPL=<-
001F 208 PUT_SEQ- : Sequential record access
001F 209 PUT_KEY- : Access by key value
001F 210 > : Access by record file address
014E 31 0028 211 BRW ERRRAC : Branch aid
0157 31 002B 212 10$: BRW EXIT : Branch aid
014F 31 002E 213 20$: BRW ERRFTM : Branch aid
0031 214
0031 215 :+
0031 216 : Sequential access has been requested.
0031 217 :
0031 218 : Note: In record transfer mode a Control message must be sent for each record
0031 219 : retrieved, whereas, in file transfer mode only an initial Control
0031 220 : message is required.
0031 221 :-
0031 222
0031 223 PUT_SEQ:
05 6A 2D EO 0031 224 BBS #IFBSV_SQO,(R10),10$ : Branch if in file transfer mode
    51 00 9A 0035 225 MOVZBL #DAP$K_SEQ_ACC,R1 : Set RAC for DAP message
    0B 11 0038 226 BRB PUT_SEQ_COMMON : Join common code
    67 19 E5 003A 227 10$: BBCC #NWSV_FTM_INIT,(R7),- : Branch if no Control message required
    25 : PUT_SEND_DAT : and turn off single-shot flag
    003D 228
    003E 229 $SETBIT #NWSV_FTM_STORE,(R7) : Set file transfer mode storage flag
    51 03 9A 0042 230 MOVZBL #DAP$K_SEQ_FILE,R1 : Set RAC for DAP message
    0045 231
0045 232 PUT_SEQ_COMMON: : Common code for put operation
    50 04 DO 0045 233 MOVL #DAP$K_CTL_MSG,R0 : Get message type value
    FFB5' 30 0048 234 RSBW NT$BUI[CD HEAD : Construct message header
    85 04 90 004B 235 MOVB #DAP$K_PUT_WRITE,(R5)+ : Store CTLFUNC field
    85 09 90 004E 236 MOVB #<<DAP$M_RAC>!- : Store CTLMENU field
    0051 237 <DAP$M_ROP>!-
    0051 238 0>,(R5)+
    85 51 90 0051 239 MOVB R1,(R5)+ : Store RAC field

```

```

0054 240
0054 241 :+
0054 242 : Common code to build rest of the Control message and send it to partner.
0054 243 :-
0054 244
0054 245 PUT_SEND_CTL_COMMON: ; Used for both put and update
FFA9' 30 0054 246 BSBW NT$MAP ROP ; Store ROP as an extensible field
FFA6' 30 0057 247 BSBW NT$BUI[CD] TAIL ; Finish building message
FFA3' 30 005A 248 BSBW NT$TRANSMIT ; Send Control message to FAL
03 50 EB 005D 249 BLBS RO,PUT_SEND_DAT ; Branch on success
0122 31 0060 250 BRW EXIT ; Branch aid
0063 251
0063 252 :+
0063 253 : Build and send DAP Data message to partner containing user record.
0063 254 :-
0063 255
0063 256 PUT_SEND_DAT:
04 6A 2D E0 0063 257 BBS #IFBSV_SQO,(R10),10$ ; Branch if in file transfer mode
50 08 DO 0067 258 $SETBIT #NWSV_LAST MSG,(R7) ; Declare this last message to block
FF8F' 30 006E 260 10$: MOVL #DAP$K_DAT MSG,R0 ; Get message type value
54 00F4 C7 DO 0071 261 BSBW NT$BUI[CD] HEAD ; Construct message header
85 94 0076 262 MOVL NWSQ_BLD+4(R7),R4 ; Get address of build message buffer
53 55 DO 0078 263 CLRB (R5)+ ; Zero RECNUM field
52 14 A6 3C 007B 264 MOVL R5,R3 ; Save next byte pointer
51 55 54 C3 007F 265 MOVZWL BDB$W_NUMB(R6),R2 ; Get record size
55 51 52 C1 0083 266 SUBL3 R4,R5,R1 ; Compute # DAP overhead bytes in msg
00CA C7 55 B1 0087 267 ADDL3 R2,R1,R5 ; Compute projected size of DAP message
38 1A 008C 268 CMPW R5,NWSW_DAPBUFSIZ(R7) ; Make sure message will fit in buffer
0120 C7 52 7D 008E 269 BGTRU 40$ ; Branch if record is too big
20 BB 0093 270 MOVQ R2,NWSQ_SAVE_DESC(R7) ; Save descriptor of user record
63 18 B6 52 28 0095 271 PUSHR #*M<R5> ; Save register
20 BA 009A 272 MOVCL R2,@BDB$L_ADDR(R6),(R3) ; Move record into DAP message
009C 273 POPR #*M<R5> ; Restore register
009C 274 ASSUME FAB$C_UDF EQ 0
009C 275 ASSUME FAB$C_FIX EQ 1
009C 276 ASSUME FAB$C_VAR EQ 2
009C 277 ASSUME FAB$C_VFC EQ 3
009C 278 ASSUME FAB$C_STM EQ 4
009C 279 ASSUME FAB$C_STMLF EQ 5
009C 280 ASSUME FAB$C_STMCR EQ 6
009C 281
009C 282 $CASEB SELECTOR=NWSB_RFM(R7)- ; Dispatch on record format:
009C 283 BASE=#FAB$C_STM-
009C 284 DISPL=<-
009C 285 20$- ; STM
009C 286 60$- ; STMLF
009C 287 70$- ; STMCR
009C 288 > ; UDF, FIX, VAR, VFC
50 11 00A8 289 BRB 100$ ; Bypass stream specific processing
00AA 290
00AA 291 :
00AA 292 : The remote file is in stream format--add a CRLF to the user record if it
00AA 293 : does not end in a stream form effector character.
00AA 294 :
00AA 295
14 A6 B5 00AA 296 20$: TSTW BDB$W_NUMB(R6) ; Branch if this is a null record

```

```

50   FF 0F 13 00AD 297      BEQL 30$      :
      FF A3 9A 00AF 298      MOVZBL -1(R3),R0 : Get record terminator character
      1F 50 91 00B3 299      CMPB R0,#31 : Branch if out-of-range for a stream
      06 1A 00B6 300      BGTRU 30$ : terminator character
19 FF43 CF 50 E0 00B8 301      BBS R0,W*NT$STM_MASK,60$ : Branch on stream terminator
      55 02 C0 00BE 302 30$: ADDL2 #2,R5 : Increase record length for CRLF
00CA C7 55 B1 00C1 303      CMPW R5,NWASW_DAPBUFSIZ(R7) : Make sure message will fit in buffer
      03 1B 00C6 304 40$: BLEQU 50$ : Branch if record will fit
      00A7 31 00C8 305      BRW ERRRSZ : Branch (record is too big)
83 0A0D 8F B0 00CB 306 50$: MOVW #CRLF,(R3)+ : Add CRLF to terminate record
0120 C7 02 C0 00D0 307      ADDL2 #2,NWASQ_SAVE_DESC(R7) : Update record length in descriptor
      23 11 00D5 308      BRB 100$ : Branch to common code
      00D7 309 :
      0GD7 310 :
      00D7 311 : The remote file is in STMLF or STMCR format--add a CR or LF to the user
      00D7 312 : record if it does not end in a proper terminator.
      00D7 313 :
      00D7 314 :
      50 0A 9A 00D7 315 60$: MOVZBL #LF,R0 : Setup terminator to add
      03 11 00DA 316      BRB 80$ :
      50 0D 9A 00DC 317 70$: MOVZBL #CR,R0 : Setup terminator to add
      14 A6 B5 00DF 318 80$: TSTW BDB$W_NUMB(R6) : Is it a null record?
      06 13 00E2 319      BEQL 90$ : Branch if yes
      FF A3 50 91 00E4 320      CMPB R0,-1(R3) : Is the proper terminator present?
      10 13 00E8 321      BEQL 100$ : Branch if yes
      55 D6 00EA 322 90$: INCL R5 : Increase record length
00CA C7 55 B1 00EC 323      CMPW R5,NWASW_DAPBUFSIZ(R7) : Will message fit in buffer?
      7F 1A 00F1 324      BGTRU ERRRSZ : Branch if record is too big
      83 50 90 00F3 325      MOVW R0,(R3)+ : Add the terminator
0120 C7 D6 00F6 326      INCL NWASQ_SAVE_DESC(R7) : Update record length in descriptor
      00FA 327 :
      00FA 328 :
      00FA 329 : End of record conversion.
      00FA 330 :
      00FA 331 :
      55 53 D0 00FA 332 100$: MOVL R3,R5 : Save next byte pointer
      FF00' 30 00FD 333      BSBW NT$BUILD_TAIL : Finish building message
      15 E1 0100 334      BBC #DAP$V_DAPCRC,- : Branch if partner does not support
      11 28 A7 0102 335      DAP$Q_SYSCAP(R7),110$ : file level CRC checksum
52 0120 C7 7D 0105 336      MOVQ NWASQ_SAVE_DESC(R7),R2 : Put descriptor of record in <R2,R3>
      0000' CF 0B 010A 337      CRC W*NT$CRC_TABLE,- : Compute CRC (destroying R0-R3)
      20 A7 010E 338      DAP$L_CRC_RSLT(R7),- : using result of previous CRC
      63 52 0110 339      R2,(R3) : calculation as initial CRC value
      20 A7 50 D0 C112 340      MOVL R0,DAP$L_CRC_RSLT(R7) : Store CRC resultant value
      FEE7' 30 0116 341 110$: BSBW NT$TRANSMIT : Send record
      43 50 E9 0119 342      BLBC R0,CHKSTS : Branch on failure
      OA A6 02 8A 011C 343      BICB2 #<1@BDB$V_DRT>,BDB$B_FLGS(R6) : Say not dirty anymore
      0120 344 :
      0120 345 :+
      0120 346 : Receive DAP Status message from partner if we are in record transfer mode
      0120 347 : and return record file address of the first block accessed.
      0120 348 :-
      0120 349 :
      61 6A 2D E0 0120 350 PUT_RECV_STS:
      5D 67 24 E1 0124 351      BBS #IFBSV_SQO,(R10),EXIT : Branch if in file transfer mode
      0128 352      BBC #DAP$V_GEQ_V56,(R7),EXIT : Branch if partner uses DAP before V5.6
      353 : ***** $SETBIT #DAP$K_STS_MSG,DAP$L_MSG_MASK(R7) Implied for receive

```

```

FED5' 30 0128 354 BSBW NT$RECEIVE ; Obtain status of put/update request
31 50 E9 0128 355 BLBC RO,CHKSTS ; Branch on failure
FECE' 30 012E 356 BSBW NT$RET_RFA ; Return RFA value to user RAB
FECC' 30 0131 357 BSBW NT$RET_RRN ; Return relative record number in BKT
; field (if appropriate)
4F 11 0134 358 BRB EXIT ; Branch aid
0136 360
0136 361 ;+
0136 362 ; Access by relative record number or by key value has been requested.
0136 363 ; This operation must be performed in DAP record transfer mode.
0136 364 ;-
0136 365
0136 366 PUT_KEY:
46 6A 2D E0 0136 367 BBS #IFBSV_SQO,(R10),ERRFTM ; Disallow file transfer mode
50 04 D0 013A 368 MOVL #DAPSK_CTL_MSG,R0 ; Get message type value
FECE' 30 013D 369 BSBW NT$BUIED HEAD ; Construct message header
85 04 90 0140 370 MOVB #DAPSK_PUT_WRITE,(R5)+ ; Store CTLFUNC field
20 00C6 C7 91 0143 371 CMPB N$ASB_ORG(R7),#N$ASK_IDX ; Branch if IDX organization
85 08 90 014A 372 BEQL 10$
014D 373 MOVB #<<DAPSM_RAC>!-- ; Store CTLMENU field
014D 374 <DAPSM_KEY>!-- ; (must send relative record number
014D 375 <DAPSM_ROP>!-- ; in key field)
014D 376 0>,(R5)+
85 01 90 014D 377 MOVB #DAPSK_KEY_ACC,(R5)+ ; Store RAC field
FEAD' 30 0150 378 BSBW NT$OBTAIN KEY ; Store key value
FEFE 31 0153 379 BRW PUT_SEND_CTL_COMMON ; Join common code
85 09 90 0156 380 10$: MOVB #<<DAPSM_RAC>!-- ; Store CTLMENU field
0159 381 <DAPSM_ROP>!-- ; (key string is implicit in data)
0159 382 0>,(R5)+
85 01 90 0159 383 MOVB #DAPSK_KEY_ACC,(R5)+ ; Store RAC field
FEF5 31 015C 384 BRW PUT_SEND_CTL_COMMON ; Join common code
015F 385
015F 386 ;+
015F 387 ; Common exit code for $PUT and $UPDATE.
015F 388 ;-
015F 389
22 67 08 E0 015F 390 CHKSTS: BBS #N$ASV_UPDATE,(R7),EXIT ; Branch if this is an update operation
30 A7 91 0163 391 CMPB DAPSB_TYPE(R7),- ; Branch if failure was not the result
09 0166 392 #DAPSR_STS_MSG ; of Status message returned by FAL
1C 12 0167 393 BNEQ EXIT
01 BB 0169 394 PUSHR #*M<R0> ; Save primary error code
FE92' 30 016B 395 BSBW NT$RESUME_FAL ; Tell FAL what to do on put error via
016E 396 ; interrupt Continue Transfer message
01 BA 016E 397 POPR #*M<R0> ; Restore primary error code
13 11 0170 398 BRB EXIT
0C 11 0172 399 ERRRSZ: RMSERR RSZ ; Invalid record size
05 11 0177 400 BRB EXIT
0179 401 ERRRAC: RMSERR RAC ; Invalid record access value
017E 402 BRB EXIT
0180 403 ERRFTM: RMSERR FTM ; File transfer mode error
0C A9 7C 0185 404 EXIT: $CLRBIT #N$ASV_UPDATE,(R7) ; Initialize flag for next time thru
0189 405 CLRQ IRBSL_IOS(R9) ; Zero I/O status block
018C 406 ; (pertinent status info is already
018C 407 ; in R0 and RAB$L_STV)
00F0 8F BA 018C 408 POPR #*M<R4,R5,R6,R7> ; Restore registers
05 0190 409 RSB ; Exit with RMS code in R0

```

```

0191 411      .SBTTL  NTSUPDATE - PERFORM NETWORK UPDATE FUNCTION
0191 412
0191 413 :++
0191 414 : NTSUPDATE - engages in a DAP dialogue with the remote FAL to update (modify)
0191 415 : the specified record of a sequential, relative, or indexed file.
0191 416 :
0191 417 : Calling Sequence:
0191 418 :
0191 419 :     BSBW  NTSUPDATE
0191 420 :
0191 421 : Input Parameters:
0191 422 :
0191 423 :     R4      BDB address
0191 424 :     R8      RAB address
0191 425 :     R9      IRAB address
0191 426 :     R10     IFAB address
0191 427 :     R11     Impure Area address
0191 428 :
0191 429 : Implicit Inputs:
0191 430 :
0191 431 :     BDB buffer contents
0191 432 :     BDB$$_ADDR
0191 433 :     BDB$$_NUMB
0191 434 :     DAP$$_DAPCRC
0191 435 :     DAP$$_GEQ_V56
0191 436 :     DAP$$_CRC_RSLT
0191 437 :     IFB$$_NWA_PTR
0191 438 :     IFB$$_SQO
0191 439 :     NWA$$_RFM
0191 440 :     NWA$$_DAPBUFSIZ
0191 441 :     NWA$$_BLD
0191 442 :     RAB$$_RAC
0191 443 :     RAB$$_ROP
0191 444 :
0191 445 : Output Parameters:
0191 446 :
0191 447 :     R0      Status code (RMS)
0191 448 :     R1-R7   Destroyed
0191 449 :     AP      Destroyed
0191 450 :
0191 451 : Implicit Outputs:
0191 452 :
0191 453 :     BDB$$_FLGS
0191 454 :     DAP$$_CRC_RSLT
0191 455 :     IRB$$_IOS and IRB$$_IOS4 zeroed
0191 456 :     RAB$$_RFA
0191 457 :
0191 458 : Completion Codes:
0191 459 :
0191 460 :     Standard RMS completion codes
0191 461 :
0191 462 : Side Effects:
0191 463 :
0191 464 :     None
0191 465 :
0191 466 :--
0191 467

```

```

0191 468 NTSUPDATE:: ; Entry point
0191 469 $STSTPT NTUPDATE ;
0197 470 PUSHR #^M<R4,R5,R6,R7> ; Save registers
E1 00F0 8F BB 019B 471 BBS #IFBSV_SQ0,(R10),ERRFTM ; Network update function not allowed
    6A 2D E0 019F 472 ; if file transfer mode selected
57 56 54 DO 019F 473 MOVL R4,R6 ; Copy BDB address
    3C AA DO 01A2 474 MOVL IFBSL_NWA_PTR(R10),R7 ; Get address of NWA (and DAP)
01A6 475 $SETBIT #NWA$V_UPDATE,(R7) ; Denote this is an update operation
01AA 476
01AA 477 :+
01AA 478 : Build and send DAP Control message to partner.
01AA 479 :-
01AA 480
01AA 481 UPD_SEND_CTL: ;
01AA 482 $SETBIT #NWA$V_LAST_MSG,(R7) ; Declare this last message to block
50 04 DO 01AE 483 MOVL #DAP$K_CTL_MSG,R0 ; Get message type value
    FE4C 30 01B1 484 BSBW NTSBUIED_HEAD ; Construct message header
85 03 90 01B4 485 MOV B #DAP$K_UPDATE,(R5)+ ; Store CTLFUNC field
85 09 90 01B7 486 MOV B #<<DAP$M_RAC>!- ; Store CTLMENU field
    01BA 487 <DAP$M_ROP>!-
    01BA 488 0>,(R5)+
01 85 00 90 01BA 489 MOV B #DAP$K_SEQ_ACC,(R5)+ ; Store RAC field--assume SEQ access
    1E A5 91 01BD 490 CMPB RAB$B_RAC(R5),#RAB$C_KEY ; Keyed access specified?
    04 12 01C1 491 BNEQ 10$ ; Branch if not
FF A5 01 90 01C3 492 MOV B #DAP$K_KEY_ACC,-1(R5) ; Change RAC field
    FEBA 31 01C7 493 10$: BRW PUT_SEND_CTL_COMMON ; Join common code
    
```

```

01CA 495          .SBTTL NT$DELETE - PERFORM NETWORK DELETE FUNCTION
01CA 496
01CA 497 :++
01CA 498 : NT$DELETE - engages in a DAP dialogue with the remote FAL to delete the
01CA 499 : specified record of a relative or indexed file.
01CA 500
01CA 501 : Calling Sequence:
01CA 502
01CA 503 :         BSBW  NT$DELETE
01CA 504
01CA 505 : Input Parameters:
01CA 506
01CA 507 :         R8      RAB address
01CA 508 :         R9      IRAB address
01CA 509 :         R10     IFAB address
01CA 510 :         R11     Impure Area address
01CA 511
01CA 512 : Implicit Inputs:
01CA 513
01CA 514 :         DAP$V_GEQ_V56
01CA 515 :         IFB$$_NWA_PTR
01CA 516 :         IFB$$_SQO
01CA 517 :         NWA$$_ORG
01CA 518
01CA 519 : Output Parameters:
01CA 520
01CA 521 :         R0      Status code (RMS)
01CA 522 :         R1-R3   Destroyed
01CA 523 :         AP      Destroyed
01CA 524
01CA 525 : Implicit Outputs:
01CA 526
01CA 527 :         None
01CA 528
01CA 529 : Completion Codes:
01CA 530
01CA 531 :         Standard RMS completion codes
01CA 532
01CA 533 : Side Effects:
01CA 534
01CA 535 :         None
01CA 536
01CA 537 :--
01CA 538
01CA 539 NT$DELETE:: ; Entry point
01CA 540 $STPT NTDELETE ;
01CA 541 PUSHR #^M<R4,R5,R6,R7> ; Save registers
38 00F0 8F BB 01D0 541 ;
6A 2D E0 01D4 542 BBS #IFB$$_SQO,(R10),ERRFTM2 ; Network delete function not allowed
57 56 >4 D0 01D8 543 ; if file transfer mode selected
3C AA D0 01DB 544 MOVL R4,R6 ; Copy BDB address
00 00C6 C7 91 01DF 545 MOVL IFB$$_NWA_PTR(R10),R7 ; Get address of NWA (and DAP)
31 13 01E4 546 CMPB NWA$$_ORG(R7),#NWA$$_SEQ ; Screen out SEQ file organization
01E6 547 BEQL ERRIOP ; Branch on error
01E6 548
01E6 549 :+
01E6 550 : Build and send DAP Control message to partner.
01E6 551 :-

```


NTOPUT
Symbol table

NETWORK PUT/UPDATE/DELETE RECORD N 4

16-SEP-1984 00:05:11 VAX/VMS Macro V04-00
5-SEP-1984 16:21:02 [RMS.SRC]NTOPUT.MAR;1

\$\$PSECT_EP	= 00000000
\$\$COUNT	= 00000003
\$\$RMSTEST	= 0000001A
\$\$RMS_PDEBUGCHK	= 00000010
\$\$RMS_TDEBUGCHK	= 00000008
\$\$RMS_UMODE	= 00000004
BDBSB_FLGS	= 0C00000A
BDBSL_ADDR	= 00000018
BDBSV_DRT	= 00000001
BDBSW_NUMB	= 00000014
CHKSTS	0000015F R 01
CR	= 0000000D
CRLF	= 00000A0D
CTRLZ	= 0000001A
DAPSB_BITCNT	00000035
DAPSB_BLKCNT	00000056
DAPSB_CTLFUNC	00000040
DAPSB_DCODE_FID	00000019
DAPSB_DCODE_MAC	0000001B
DAPSB_DCODE_MSG	0000001A
DAPSB_DECVER	00000047
DAPSB_ECONUM	00000045
DAPSB_FILESYS	00000043
DAPSB_FLAGS	00000031
DAPSB_KRF	00000047
DAPSB_LEN256	00000034
DAPSB_LENGTH	00000033
DAPSB_OSTYPE	00000042
DAPSB_RAC	00000046
DAPSB_STREAMID	00000032
DAPSB_TYPE	00000030
DAPSB_USRNUM	00000046
DAPSB_USRVER	00000048
DAPSB_VERNUM	00000044
DAPSB_X_FIELD	00000024
DAPSC_BCN	000000C0
DAPSK_BLN	000000C0
DAPSK_CTL_MSG	= 00000004
DAPSK_DAT_MSG	= 00000008
DAPSK_DELETE	= 00000005
DAPSK_KEY_ACC	= 00000001
DAPSK_PUT_WRITE	= 00000004
DAPSK_SEQ_ACC	= 00000000
DAPSK_SEQ_FILE	= 00000003
DAPSK_STG_MSG	= 00000009
DAPSK_UPDATE	= 00000003
DAPSL_CMWA	00000030
DAPSL_CRC_RSLT	00000020
DAPSL_DCODE_STS	00000018
DAPSL_MSG_MASK	0000001C
DAPSL_RECRUM1	00000040
DAPSL_ROP	00000050
DAPSL_SSPWA	00000080
DAPSL_TEMP	00000090
DAPSM_BITCNT	= 00000008
DAPSM_BLKCNT	= 00000040
DAPSM_KEY	= 00000002

DAPSM_RAC	= 00000001
DAPSM_ROP	= 00000008
DAPSM_SEGMENT	= 00000040
DAPSM_TMP1\$	= 00000008
DAPSM_TMP2\$	= FFF80000
DAPSQ_DCODE_FLG	00000000
DAPSQ_FILEDATA	00000044
DAPSQ_KEY	00000048
DAPSQ_MSG_BUF1	00000008
DAPSQ_MSG_BUF2	00000010
DAPSQ_SYSCAP	00000028
DAPSQ_SYSPEC	00000038
DAPSV_DAPCRC	= 00000015
DAPSV_GEQ_V56	= 00000024
DAPSW_BUF1SZ	00000040
DAPSW_CTLMENU	00000044
DAPSW_DISPLAY2	00000054
DAPSW_PARTNER	00000006
DAPSW_VERSION	00000004
DC1	= 00000011
DC2	= 00000012
DC3	= 00000013
DC4	= 00000014
DEL_RECV_STS	00000202 R 01
DEL_SEND_CTL	000001E6 R 01
DLE	= 00000010
ERRFTM	00000180 R 01
ERRFTM2	00000210 R R 01
ERRIOP	00000217 R R 01
ERRRAC	00000179 R R 01
ERRRSZ	00000172 R 01
ESC	= 0000001B
EXIT	00000185 R 01
EXIT2	00000209 R 01
FABSC_FIX	= 00000001
FABSC_STM	= 00000004
FABSC_STMCR	= 00000006
FABSC_STMLF	= 00000005
FABSC_UDF	= 00000000
FABSC_VAR	= 0000C002
FABSC_VFC	= 00000003
FF	= 0000000C
IFBSL_NWA_PTR	= 0000003C
IFBSV_SQO	= 0000002D
IRBSL_IOS	= 0000000C
LF	= 0000000A
NT\$BUILD_HEAD	***** X 01
NT\$BUILD_TAIL	***** X 01
NT\$CHK_RAC	***** X 01
NT\$CRC_TABLE	***** X 01
NT\$DELETE	000001CA RG 01
NT\$EXRMS	***** X 01
NT\$MAP_ROP	***** X 01
NT\$OBTAIN_KEY	***** X 01
NT\$PUT	00000004 RG 01
NT\$RECEIVE	***** X 01
NT\$RESUME_FAL	***** X 01

NT(Syt
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
NWA
PIC
REI
REI
REI
REI
RMI
RMI
SEI
SEI
TP

PS
--
NF
SA

NTSRET_RFA	*****	X	01	NWAST_ITM_STRING	0000020C		
NTSRET_RRN	*****	X	01	NWAST_NCBBUF	0000052C		
NTSSTM_MASK	00000000	RG	01	NWAST_NODEBUF	00000169		
NTSTRANSMIT	*****	X	01	NWAST_RCVBUF	000001A0		
NTSUPDATE	00000191	RG	01	NWAST_SCAN	00000100		
NWASB_ALLXABCNT	0000011C			NWAST_TEMP	00000120		
NWASB_DAP_RAC	000000C9			NWAST_XLTBUF1	000002AC		
NWASB_FILESYS	000000C5			NWAST_XLTBUF2	000003AC		
NWASB_KEYXABCNT	0000011D			NWAST_XMTBUF	000003C0		
NWASB_NETSTRSIZ	0000016F			NWASV_FTM_INIT	= 00000019		
NWASB_NODBUFSIZ	00000168			NWASV_FTM_RETRV	= 0000001A		
NWASB_ORG	000000C6			NWASV_FTM_STORE	= 0000001B		
NWASB_OSTYPE	000000C4			NWASV_LAST_MSG	= 00000000		
NWASB_RFM	000000C7			NWASV_UPDATE	= 00000008		
NWASB_RMS_RAC	000000C8			NWASW_BUILD	000000D2		
NWASC_BLN	00000800			NWASW_DAPBUFSIZ	000000CA		
NWASK_BLN	00000800			NWASW_DIR_OFF	000000CC		
NWASK_IDX	= 00000020			NWASW_DISPLAY	000000D0		
NWASK_SEQ	= 00000000			NWASW_FIL_OFF	000000CE		
NWASL_ALLXABADR	00000100			NWASW_JNLXABJOP	0000011E		
NWASL_DATXABADR	00000104			PIOSA_TRACE	*****	X	01
NWASL_DEV	000000C0			PUT_KEY	00000136	R	01
NWASL_FHCXABADR	00000108			PUT_RECV_STS	00000120	R	01
NWASL_KEYXABADR	0000010C			PUT_SEND_CTL_COMMON	00000054	R	01
NWASL_MSG_MASK	000000D4			PUT_SEND_DAT	00000063	R	01
NWASL_PROXABADR	00000110			PUT_SEQ	00000031	R	01
NWASL_RDTXABADR	00000114			PUT_SEQ_COMMON	00000045	R	01
NWASL_SAVE_FLGS	00000128			RABSB_RAC	= 0000001E		
NWASL_SUMXABADR	00000118			RABSC_KEY	= 00000001		
NWASL_THREAD	000000FC			RABSC_RFA	= 00000002		
NWASL_XLTATTR	00000238			RABSC_SEQ	= 00000000		
NWASL_XLTBUFFLG	0000022C			RMS\$FTM	= 000187C4		
NWASL_XLTCNT	00000228			RMS\$IOP	= 00018574		
NWASL_XLTMAXIDX	00000234			RMS\$RAC	= 00018644		
NWASL_XLTSIZ	00000230			RMS\$RSZ	= 000186A4		
NWASQ_ACS	00000244			TPT\$C_NTDELETE	*****	X	01
NWASQ_BIGBUF	00000170			TPT\$L_NTPUT	*****	X	01
NWASQ_BLD	000000F0			TPT\$L_NTUPDATE	*****	X	01
NWASQ_FLG	00000000			UPD_SEND_CTL	000001AA	R	01
NWASQ_INODE	0000025C			VT	= 0000000B		
NWASQ_IOSB	00C000D8						
NWASQ_LNODE	00000160						
NWASQ_LOGNAME	0000023C						
NWASQ_NCB	00000264						
NWASQ_RCV	000000E0						
NWASQ_SAVE_DESC	00000120						
NWASQ_XLTBUF1	0000024C						
NWASQ_XLTBUF2	00000254						
NWASQ_XMT	000000E8						
NWAST_ACSBUF	0000026C						
NWAST_AUXBUF	000005E0						
NWAST_DAP	00000000						
NWAST_INODEBUF	000004AC						
NWAST_ITM_ATTR	00000200						
NWAST_ITM_END	00000224						
NWAST_ITM_LST	0000020C						
NWAST_ITM_MAXIDX	00000218						

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
NFSNETWORK	0000021E (542.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
\$ABSS	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	37	00:00:00.10	00:00:00.84
Command processing	140	00:00:00.54	00:00:03.40
Pass 1	360	00:00:14.10	00:00:38.23
Symbol table sort	0	00:00:01.88	00:00:03.69
Pass 2	112	00:00:02.74	00:00:08.55
Symbol table output	25	00:00:00.17	00:00:00.54
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	678	00:00:19.55	00:00:55.44

The working set limit was 1650 pages.
75704 bytes (148 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1371 non-local and 24 local symbols.
584 source lines were read in Pass 1, producing 15 object records in Pass 2.
29 pages of virtual memory were used to define 28 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	20
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	24

1582 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:NTOPUT/OBJ=OBJ\$:NTOPUT MSRC\$:NTOPUT/UPDATE=(ENH\$:NTOPUT)+LIB\$:RMS/LIB

The image shows a grid of 100 small, faint screenshots of VAX/VMS system screens, arranged in 10 rows and 10 columns. Each screen displays technical data and system status. Several screens are highlighted with larger text labels:

- NT0SCNXAB LIS
- RM0CACHE LIS
- NT0RENAME LIS
- NT0OPEN LIS
- NT0SEARCH LIS
- NT0PUT LIS
- RM0ACCESS LIS