



```

NN      NN      TTTTTTTTTT  000000  SSSSSSSS  EEEEEEEEE  AAAAAA  RRRRRRRR  CCCCCCCC  HH      HH
NN      NN      TTTTTTTTTT  000000  SSSSSSSS  EEEEEEEEE  AAAAAA  RRRRRRRR  CCCCCCCC  HH      HH
NN      NN      TT          00      00  SS          EE          AA      AA  RR      RR  CC          HH      HH
NN      NN      TT          00      00  SS          EE          AA      AA  RR      RR  CC          HH      HH
NNNN    NN      TT          00      0000  SS          EE          AA      AA  RR      RR  CC          HH      HH
NNNN    NN      TT          00      0000  SS          EE          AA      AA  RR      RR  CC          HH      HH
NN  NN  NN      TT          00  00  00  SSSSSS  EEEEEEEE  AA      AA  RRRRRRRR  CC          HHHHHHHHHH
NN  NN  NN      TT          00  00  00  SSSSSS  EEEEEEEE  AA      AA  RRRRRRRR  CC          HHHHHHHHHH
NN      NNNN    TT          0000    00  SS          EE          AAAAAAAAAA  RR  RR  CC          HH      HH
NN      NNNN    TT          0000    00  SS          EE          AAAAAAAAAA  RR  RR  CC          HH      HH
NN      NN      TT          00      00  SS          EE          AA      AA  RR      RR  CC          HH      HH
NN      NN      TT          00      00  SS          EE          AA      AA  RR      RR  CC          HH      HH
NN      NN      TT          000000  SSSSSSSS  EEEEEEEEE  AA      AA  RR      RR  CCCCCCCC  HH      HH
NN      NN      TT          000000  SSSSSSSS  EEEEEEEEE  AA      AA  RR      RR  CCCCCCCC  HH      HH

```

```

LL      I111111  SSSSSSSS
LL      I111111  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  I111111  SSSSSSSS
LLLLLLLLLLLL  I111111  SSSSSSSS

```



```

0000 1          $BEGIN NTOSEARCH,000,NF$NETWORK,<NETWORK SEARCH FILES>
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 perform the search function.
0000 36 :
0000 37 : Environment: VAX/VMS, executive mode
0000 38 :
0000 39 : Author: James A. Krycka,      Creation Date: 10-SEP-1979
0000 40 :
0000 41 : Modified By:
0000 42 :
0000 43 :     V03-005 JAK0138      J A Krycka      28-MAR-1984
0000 44 :     Call modified NT$EXCH_CNF routine with a parameter.
0000 45 :
0000 46 :     V03-004 JEJ0015      J E Johnson    26-Mar-1984
0000 47 :     Modify NT$SEARCH to look for NAMS$V_SRCHXABS in NOP field as
0000 48 :     the signal to return XAB information on a $SEARCH operation.
0000 49 :
0000 50 :     V03-003 JAK0114      J A Krycka      23-JUN-1983
0000 51 :     Fix bug in V03-002.
0000 52 :
0000 53 :     V03-002 KRM0111      K Malik        31-May-1983
0000 54 :     Update to support DAP V7.0 message exchange sequence.
0000 55 :
0000 56 :--

```

NT  
Sy  
DA  
DA  
DA  
DA  
DA  
DA  
DA  
DA  
DA  
DA  
DA  
DA  
DA  
DI  
ER  
ER  
ER  
ER  
FA  
FA  
FA  
FA  
FA  
FA  
FA  
FA  
FA  
FI  
FW  
FW  
IF  
LO  
ME  
MC  
NA  
NA  
NT  
NT



```
0000 96 .SBTTL NT$SEARCH - PERFORM NETWORK SEARCH FUNCTION
0000 97
0000 98 :++
0000 99 : NT$SEARCH - engages in a DAP dialogue with the remote FAL to obtain a list
0000 100 : of directory entries that match a given (wildcard) file specification.
0000 101 :
0000 102 : The first call to NT$SEARCH causes RMS to request that FAL perform the
0000 103 : DAP directory-list function and returns the resultant name string of
0000 104 : the first file that matches the expanded name string template.
0000 105 : Thereafter, each call to NT$SEARCH returns the resultant name string
0000 106 : of the next match until no more files are found or an error occurs.
0000 107 :
0000 108 : Further, if the NAMS_L_NOP field has the NAMS_V_SRCHXABS bit set then the
0000 109 : FAB and certain XABS chained to it will be updated. This is done to
0000 110 : support the DIRECTORY utility which would otherwise have to open each
0000 111 : file to obtain its attributes, causing a separate invocation of FAL for
0000 112 : each file!
0000 113 :
0000 114 : Calling Sequence:
0000 115 :
0000 116 :     BSBW    NT$SEARCH
0000 117 :
0000 118 : Input Parameters:
0000 119 :
0000 120 :     R0      First-time-thru flag in bit0 (0=no, 1=yes)
0000 121 :     R8      FAB address
0000 122 :     R9      IFAB address
0000 123 :     R10     FWA address
0000 124 :     R11     Impure Area address
0000 125 :
0000 126 : Implicit Inputs:
0000 127 :
0000 128 :     DAP$Q_SYSCAP
0000 129 :     DAP$V_STM_ONLY
0000 130 :     FWA fields
0000 131 :     IFB$L_NWA_PTR
0000 132 :     NAMS$V_SRCHXABS
0000 133 :     NWA$V_FTM_CONT
0000 134 :
0000 135 : Output Parameters:
0000 136 :
0000 137 :     R0      Status code (RMS)
0000 138 :     R1-R7   Destroyed
0000 139 :     AP      Destroyed
0000 140 :
0000 141 : Implicit Outputs:
0000 142 :
0000 143 :     FWASQ_QUOTED
0000 144 :     FWASV_REMRESULT
0000 145 :     NWA$V_FTM_CONT
0000 146 :
0000 147 : Completion Codes:
0000 148 :
0000 149 :     Standard RMS completion codes
0000 150 :
0000 151 : Side Effects:
0000 152 :
```

```

0000 153 : None
0000 154 :
0000 155 :--
0000 156 :
0000 157 NT$SEARCH:: : Entry point
0000 158 $STPT NTSEARCH :
57 3C A9 DO 0006 159 MOVL IFB$N NWA PTR(R9),R7 : Get address of NWA (and DAP)
000A 160 $CLRBIT #FWASV REMRESULT,(R10) : Say no resultant name string received
1C 50 EB 000E 161 BLBS RO,SEND_CNF : Perform set-up with FAL if this is
0011 162 : the first call to $SEARCH
67 1E E5 0011 163 BBCC #NWSV FTM_CONT,(R7),- : Branch if previous search call was a
72 0014 164 RECV_NAM : complete success and clear flag
0015 165 :
0015 166 :+
0015 167 : Build and send DAP Continue Transfer message to partner to skip to next
0015 168 : directory entry following receipt of a DAP (error) Status message. This code
0015 169 : segment is entered iff the previous call to $SEARCH resulted in an error
0015 170 : (except RMSS_NMF) or a partial success (i.e., the file name was returned
0015 171 : without the requested file attributes because FAL was unable to open the
0015 172 : file.)
0015 173 :-
0015 174 :
0015 175 SEND_CON: : (required to continue after error)
0015 176 $SETBIT #NWSV_LAST MSG,(R7) : Declare this last message to block
50 05 DO 0019 177 MOVL #DAPSK_CON MSG,RO : Get message type value
FFE1' 30 001C 178 BSBW NT$BUICD HEAD : Construct message header
85 02 90 0C1F 179 MOV B #DAPSK SRIP_REC,(R5)+ : Store CONFUNC field
FFDB' 30 0022 180 BSBW NT$BUICD TAIL : Finish building message
FFDB' 30 0025 181 BSBW NT$TRANSMIT : Send Continue Transfer message to FAL
13 50 E9 0028 182 BLBC RO,FAIL1 : Branch on failure
5A 11 002B 183 BRB RECV_NAM : Continue now that DAP stream from FAL
002D 184 : has been restarted
002D 185 :
002D 186 :+
002D 187 : Exchange DAP Configuration messages with FAL and determine DAP buffer size.
002D 188 :-
002D 189 :
002D 190 SEND_CNF: : (required message on first pass)
50 06 DO 002D 191 MOVL #DAPSK DIR_LIST,RO : Denote type of file access
FFCD' 30 0030 192 BSBW NT$EXCH CNF : Exchange Configuration messages
08 50 E9 0033 193 BLBC RO,FAIL1 : Branch on failure
19 E0 0036 194 BBS #DAPSV DIRECTORY,- : Branch if remote FAL supports
28 A7 0038 195 DAP$Q SYSCAP(R7),- : directory-list function
04 003A 196 BUILD_MASK :
FFC2' 30 003B 197 BSBW NT$RMT_ACCFUNC : Declare RMSS_SUPPORT error
05 003E 198 FAIL1: RSB : Exit with RMS code in RO
003F 199 :
003F 200 :+
003F 201 : Build display field request mask which will be used in the Access message
003F 202 : to request that optional DAP messages be returned by FAL. For $SEARCH, these
003F 203 : are the ATT, PRO, SUM, and TIM messages. (The Attributes message will supply
003F 204 : information to update both the FAB and the FHCXAB.)
003F 205 :
003F 206 : Note: The FAB and XABs are not documented as outputs from $SEARCH. This
003F 207 : information will be returned iff the NAMS$ SRCHXABS flag is set in the
003F 208 : NOP field. It is expected that only the DIRECTORY utility will request
003F 209 : this additional output to avoid opening each file to obtain its file

```

```

003F 210 : attributes, which would be very very costly in a network context. Since
003F 211 : network search is implemented using the DAP directory_list operation,
003F 212 : XAB information can be requested on search, thus requiring only one
003F 213 : remote FAL to be active for network directory operations.
003F 214 :
003F 215 : Note: Utilities like COPY that do a $PARSE, $SEARCH-$OPEN, $SEARCH-$OPEN, etc.
003F 216 : on wildcard operations, will invoke one FAL for all search requests and
003F 217 : an additional FAL for each file subsequently opened.
003F 218 :-
003F 219 :-
003F 220 BUILD_MASK: ; Build N$W$W_DISPLAY
52 D4 003F 221 CLRL R2 ; Initialize request mask
56 D4 0041 222 CLRL R6 ; Indicate this is not a close operation
50 28 A8 D0 0043 223 MOVL FAB$$_NAM(R8),R0 ; Get NAM block address
11 13 0047 224 BEQL 10$ ; If NAM block is not present, or the
06 E1 0049 225 BBC #N$M$V_SRCHXABS,- ; network specific search flag is not
OC 08 A0 004B 226 N$M$B_NOP(R0),10$ ; set, then ignore XABs in chain
FFAF' 30 004E 227 BSBW NT$SCAN_XABCHN ; Scan user XAB chain and check FAL's
EA 50 E9 0051 228 ; capabilities; request mask put in R2
52 06 AA 0054 229 BLBC R0,FAIL1 ; Branch on failure to complete scan
0057 230 BICW2 #<<DAP$M_DSP_KEY>!- ; Ignore Key Definition and Allocation
0057 231 <DAP$M_DSP_ALL>!- ; XABs in chain
0057 232 0>,R2 ;
52 01 A8 0057 233 BISW2 #DAP$M_DSP_ATT,R2 ; Request Attributes message to get
005A 234 ; information for FHCXAB and FAB
00D0 C7 52 E0 005A 235 10$: MOVW R2,N$W$W_DISPLAY(R7) ; Save request mask
005F 236 ;
005F 237 ;+
005F 238 ; Build and send DAP Access message to partner.
005F 239 ; Note that the filespec used is the expanded name string generated from a
005F 240 ; previous call to $PARSE.
005F 241 :-
005F 242 :-
005F 243 SEND_ACC: ; (required message on first pass)
50 03 D0 0063 244 $SETBIT #N$W$V_LAST_MSG,(R7) ; Declare this last message to block
FF97' 30 0066 245 MOVL #DAP$K_ACC_MSG,R0 ; Get message type value
85 06 90 0069 246 BSBW NT$BUIED_HEAD ; Construct message header
85 01 90 006C 247 MOVB #DAP$K_DIR_LIST,(R5)+ ; Store ACCFUNC field
FF8E' 30 006F 248 MOVB #DAP$M_NONFATAL,(R5)+ ; Store ACCOPT field
0072 249 BSBW NT$GET_FILESPEC ; Store FILESPEC field as a counted
51 00D0 C7 3C 0072 250 ; ASCII string
05 13 0077 251 MOVZWL N$W$W_DISPLAY(R7),R1 ; Get request mask
85 B4 0079 252 BEQL 10$ ; Branch if nothing set
FF82' 30 007B 253 CLRW (R5)+ ; Store dummy FAC and SHR fields
FF7F' 30 007E 254 BSBW NT$CVT_BN4_EXT ; Store DISPLAY as an extensible field
FF7C' 30 0081 255 10$: BSBW NT$BUIED_TAIL ; Finish building message
B7 50 E9 0084 256 BSBW NT$TRANSMIT ; Send Access message to FAL
257 BLBC R0,FAIL1 ; Branch on failure

```



```

0087 259 :+
0087 260 : Receive next DAP Name message (or Access Complete message) from partner.
0087 261 : One, two, or three Name messages may be received with the sequence terminated
0087 262 : by a (file) Name message.
0087 263 :
0087 264 : In addition, receipt of an Access Complete message signals normal completion
0087 265 : of the search (DAP directory-list) function and is mapped into RMS$_NMF.
0087 266 :-
0087 267
0087 268 RECV_NAM: ; (required message sequence)
1C A7 00008080 8F C8 0087 269 BISL2 #<<1@DAP$K_NAM_MSG>!- ; Expect response of either Name
008F 270 <1@DAP$K_CMP_MSG>!- ; or Access Complete message
008F 271 0>,DAP$K_MSG_MASK(R7) ;
008F 272 BSBW NT$RECEIVE ; Get reply from FAL
09 50 E9 0092 273 BLBC R0,10$ ; Branch on failure
30 A7 91 0095 274 CMPB DAP$B_TYPE(R7),- ; Branch if Name message
OF 0098 275 #DAP$K_NAM_MSG ;
06 13 0099 276 BEQL MERGE_NAM ; It returns to RECV_NAM iff
009B 277 ; resultant name is still incomplete
00E4 31 009B 278 BRW SEARCH_ENDED ; It must be Access Complete or
009E 279 ; incorrect (success) Status message
0158 31 009E 280 10$: BRW FAIL4 ; Branch aid
00A1 281
00A1 282 :+
00A1 283 : A Name message has been received and decoded in the DAP control block.
00A1 284 : The new name string is merged into the existing (partial) resultant name
00A1 285 : string and control is returned to RECV_NAM unless the Name message contains a
00A1 286 : filename string which completes the resultant name string.
00A1 287 :-
00A1 288
00A1 289 MERGE_NAM: ;
50 44 A7 7D 00A1 290 MOVQ DAP$Q_NAMESPEC(R7),R0 ; Copy name descriptor to <R0,R1>
00A5 291
00A5 292 ASSUME DAP$V_FILSPEC EQ 0
00A5 293 ASSUME DAP$V_FILSPEC+1 EQ DAP$V_FILNAME
00A5 294 ASSUME DAP$V_FILNAME+1 EQ DAP$V_DIRNAME
00A5 295 ASSUME DAP$V_DIRNAME+1 EQ DAP$V_VOLNAME
00A5 296 ASSUME DAP$V_VOLNAME+1 EQ DAP$V_DFTSPEC
00A5 297
00A5 298 FFS #0,#DAP$V_DFTSPEC+1,- ; Convert NAMETYPE field bit offset
52 05 00 EA 00A8 299 DAP$B_NAMETYPE(R7),R2 ; to a value
00AB 300 $CASEB SELECTOR=R2- ; Type of name string:
00AB 301 DISPL=<- ;
00AB 302 10$- ; FILSPEC -- error
00AB 303 FIL_NAM- ; FILNAME
00AB 304 DIR_NAM- ; DIRNAME
00AB 305 VOL_NAM- ; VOLNAME
00AB 306 10$- ; DFTSPEC -- error
00AB 307 > ;
FF44' 31 00B9 308 10$: BRW NT$BUG_NAMETYPE ; Declare RMS$ BUG_DAP error and
00BC 309 ; exit with RMS code in R0
00BC 310
00BC 311 :+
00BC 312 : Process (volume) Name message.
00BC 313 :-
00BC 314
00BC 315 VOL_NAM: ; Save device (volume) Name string

```

```

0170 CA 50 D0 00BC 316      MOVL   R0,FWASQ_QUOTED(R10) ; Update descriptor length
00CC C7 50 B0 00C1 317      MOVW   R0,NWASW_DIR_OFF(R7) ; Update offset for directory string
      61 50 28 00C6 318      MOVW   R0,(R1),- ; Copy volume name string to
      0174 DA 00C9 319      @FWASQ_QUOTED+4(R10) ; resultant name string buffer
      B9 11 00CC 320      BRB    RECV_NAM ; Get next name string component
      00CE 321
      00CE 322 ;+
      00CE 323 ; Process (directory) Name message.
      00CE 324 ; -
      00CE 325
      00CE 326 DIR_NAM: ; Save directory name string
      54 00CC C7 3C 00CE 327      MOVZWL  NWASW_DIR_OFF(R7),R4 ; Get offset to use
      55 54 50 C1 00D3 328      ADDL3   R0,R4,R5 ; Compute new offset
0170 CA 55 D0 00D7 329      MOVL   R5,FWASQ_QUOTED(R10) ; Update descriptor length
00CE C7 55 B0 00DC 330      MOVW   R5,NWASW_FIL_OFF(R7) ; Update offset for file name string
      61 50 28 00E1 331      MOVW   R0,(R1),- ; Append directory name to
      0174 DA44 00E4 332      @FWASQ_QUOTED+4(R10)[R4]; resultant name string buffer
      9D 11 00E8 333      BRB    RECV_NAM ; Get next name string component
      00EA 334
      00EA 335 ;+
      00EA 336 ; Process (file) Name message.
      00EA 337 ; -
      00EA 338
      00EA 339 FIL_NAM: ; Assemble resultant name string
0170 CA 54 00CE C7 3C 00EA 340      MOVZWL  NWASW_FIL_OFF(R7),R4 ; Get offset to use
      54 50 C1 00EF 341      ADDL3   R0,R4,FWASQ_QUOTED(R10) ; Update descriptor length
      61 50 28 00F5 342      MOVW   R0,(R1),- ; Append file-type-version name to
      0174 DA44 00F8 343      @FWASQ_QUOTED+4(R10)[R4]; resultant name string buffer
      00FC 344      SSETBIT #FWASV_REMRESULT,(R10) ; Flag receipt of resultant string

```

```

0100 346 :+
0100 347 : Receive DAP Attributes message from partner and update the user FAB and
0100 348 : FHCXAB.
0100 349 :
0100 350 : In addition, receipt of an Access Complete message signals normal completion
0100 351 : of the search (DAP directory-list) function and is mapped into RMS$_NMF.
0100 352 :
0100 353 : Note: The user XAB chain is scanned again to probe all user XABs to protect
0100 354 : RMS from a user who deletes the address space where an XAB was
0100 355 : previously found.
0100 356 :-
0100 357 :
0100 358 RECV_ATT: : (optional--must be requested)
0100 359 BBC #DAPSV_DSP_ATT,- : Branch if Attributes message
0000 C7 E1 0102 360 N$WASV_DISPLAY(R7),- : was not requested
31 0105 361 RECV_EXT_ATT :
1C A7 00000084 8F C8 0106 362 BISL2 #<<1@DAP$K_ATT_MSG>!-- : Expect response of either Attributes
010E 363 <1@DAP$K_CMP_MSG>!-- : or Access Complete message
010E 364 0>_DAP$L_MSG_MASK(R7) :
FEEF' 30 010E 365 BSBW NT$RECEIVE : Get reply from FAL
03 50 E8 0111 366 BLBS RO,SS : Branch on success
0088 31 0114 367 BRW FAIL3 : Branch on failure
30 A7 91 0117 368 5$: CMPB DAP$B_TYPE(R7),- : Branch if not Attributes message
02 011A 369 #DAP$K_ATT_MSG :
65 12 011B 370 BNEQ SEARCH_ENDED : It must be Access Complete or
04 64 A7 06 E1 011D 371 : incorrect (success) Status message
0122 372 BBC #DAPSV_FLK,DAP$L_FOP1(R7),10$ :
56 D4 0126 374 10$: CLRL R6 : Denote FOP FLK bit set
FED5' 30 0128 375 BSBW NT$SCAN_XABCHN : Indicate this is not a close operation
59 50 E9 012B 376 BLBC RO,FAIL2 : Scan user XAB list again
0149 30 012E 377 BSBW MOD_RFM_RAT : Branch on failure to scan XABs
00D0 30 0131 378 BSBW UPDATE_FAB : Check for undefined record format
FEC9' 30 0134 379 BSBW NT$UPDATE_FHC : Update user FAB
0137 380 : Update user FHCXAB
0137 381 :+
0137 382 : Receive DAP Extended Attributes messages from partner and update the user
0137 383 : DAT, PRO, and SUM XABs. These messages may be received in any order.
0137 384 :
0137 385 : In addition, receipt of an Access Complete message signals normal completion
0137 386 : of the search (DAP directory-list) function and is mapped into RMS$_NMF.
0137 387 :-
0137 388 :
0137 389 RECV_EXT_ATT: : (optional--must be requested)
51 00D0 C7 3C 0137 390 #OVZWL N$WASV_DISPLAY(R7),R1 : Get DAP message request mask
52 D4 013C 391 CLRL R2 : Clear valid messages to receive mask
013E 392 $MAPBIT DAPSV_DSP_PRO,DAP$K_PRO_MSG; : Map request for Protection message
0146 393 $MAPBIT DAPSV_DSP_SUM,DAP$K_SUM_MSG; : Map request for Summary message
014E 394 $MAPBIT DAPSV_DSP_TIM,DAP$K_TIM_MSG; : Map request for Date and Time msg
00D4 C7 52 D0 0156 395 MOVL R2,N$WASL_MSG_MASK(R7) : Save valid message mask for use again
00D4 C7 D0 015B 396 LOOP: MOVL N$WASL_MSG_MASK(R7),- : Expect response of any of these DAP
1C A7 015F 397 DAP$L_MSG_MASK(R7) : messages
25 13 0161 398 BEQL SEARCH_SUCCESS : Branch if no more to receive
0163 399 $SETBIT #DAP$K_CMP_MSG,DAP$L_MSG_MASK(R7) :
0168 400 : Access complete is also valid
FE95' 30 0168 401 BSBW NT$RECEIVE : Get reply from FAL
31 50 E9 016B 402 BLBC RO,FAIL3 : Branch on failure

```

```

30 A7 91 016E 403      CMPB  DAP$B_TYPE(R7),-      ; Branch if Access Complete message
07      0171 404      #DAP$R_CMP_MSG          ;
OE      13 0172 405      BEQL  SEARCH_ENDED        ;
      0174 406
      0174 407      ASSUME DAP$K_SUM_MSG EQ 12
      0174 408      ASSUME DAP$K_TIM_MSG EQ 13
      0174 409      ASSUME DAP$K_PRO_MSG EQ 14
      0174 410
E4 AF 9F 0174 411 10$:  PUSHAB B^LOOP          ; Push return address on stack
      0177 412      $CASEB SELECTOR=DAP$B_TYPE(R7)- ; Dispatch to process message:
      0177 413      BASE=#DAP$K_SUM_MSG-
      0177 414      DISPL=<-
      0177 415      NT$DECODE_SUM-      ; Summary message
      0177 416      NT$DECODE_TIM-      ; Date and Time message
      0177 417      NT$DECODE_PRO-      ; Protection message
      0177 418      >
      0182 419
      0182 420 :+
      0182 421 : This call to $SEARCH has found no more files; the search sequence has
      0182 422 : terminated.
      0182 423 :-
      0182 424
      0182 425 SEARCH_ENDED:          ; Success path for no more files
      0182 426      RMSERR NMF        ; Convert Access Complete message
      0187 427      ; into no-more-files status code
05      0187 428 FAIL2: RSB          ; Exit with RMS code in R0
      0188 429
      0188 430 :+
      0188 431 : This call to $SEARCH has found a match; a resultant name string has been
      0188 432 : returned and user control blocks have been updated per CTX field.
      0188 433 :-
      0188 434
      0188 435 SEARCH_SUCCESS:        ; Success path for next file found
      0188 436
      0188 437 :+
      0188 438 : Receive DAP Acknowledge message from partner.
      0188 439 :-
      0188 440
      0188 441 RECV_ACK:          ; Acknowledge message required if
OB 67 26 E1 0188 442      BBC #DAP$V_GEQ_V70,(R7),10$ ; partner uses DAP since V7.0
      018C 443      $SETBIT #DAP$K_ACK_MSG,DAP$L_MSG_MASK(R7)
      0191 444      ; Expect response of Acknowledge message
      FE6C' 30 0191 445      BSBW NT$RECEIVE          ; Get reply from FAL
07 50 E9 0194 446      BLBC RO,20$          ; Branch on failure
67 12 E4 0197 447 10$:  BBSC #N$WASV_FOP_FLK,(R7),- ; Map FOP FLK bit returned by FAL into
4D      019A 448      ERRACC          ; partial success (SS$_FILELOCKED)
      019B 449      ; Return success
      05 019E 450 20$:  RMBSSUC      ;
      RSB          ; Exit with RMS code in R0

```

```

019F 452 :+
019F 453 : An error has occurred while trying to obtain the attributes of the file.
019F 454 : Determine whether to return the error or whether to return success with an
019F 455 : appropriate system service code in STV. The latter denotes partial success
019F 456 : and that the search can be resumed. Note that the DIRECTORY utility examines
019F 457 : the error code in the STV to decide whether to terminate the directory
019F 458 : operation or to continue issuing $SEARCH calls (i.e., whether or not the
019F 459 : error condition is fatal).
019F 460 :-
019F 461 :-
828A 8F 50 B1 019F 462 FAIL3: CMPW R0,#<RMS$_FLK&^XFFFF> ; Map file locked error
      3B 13 01A4 463 BEQL ERRFLK ;
829A 8F 50 B1 01A6 464 CMPW R0,#<RMS$_PRV&^XFFFF> ; Map privilege violation
      42 13 01AB 465 BEQL ERRPRV ;
C002 8F 50 B1 01AD 466 CMPW R0,#<RMS$_ACC&^XFFFF> ; Map file access error
      34 13 01B2 467 BEQL ERRACC ;
C0CC 8F 50 B1 01B4 468 CMPW R0,#<RMS$_ATR&^XFFFF> ; Map read attributes error
      1F 13 01B9 469 BEQL ERRATR ;
874C 8F 50 B1 01BB 470 CMPW R0,#<RMS$_NET&^XFFFF> ; Check for general network error
      37 12 01C0 471 BNEQ FAIL4 ; Fatal error if no match
40E7 8F 0C A8 B1 01C2 472 CMPW FAB$L_STV(R8),#^X40E7 ; Map read attributes error
      10 13 01C8 473 BEQL ERRATR ;
40E9 8F 0C A8 B1 01CA 474 CMPW FAB$L_STV(R8),#^X40E9 ; Map privilege violation
      1D 13 01D0 475 BEQL ERRPRV ;
40EA 8F 0C A8 B1 01D2 476 CMPW FAB$L_STV(R8),#^X40EA ; Map no attributes for file error
      1F 12 01D8 477 BNEQ FAIL4 ; Fatal error if no match
51 0888 8F 3C 01DA 478 ERRATR: MOVZWL #SS$_FCPREADERR,R1 ; Generate SS code for RMS$_ATR
      11 11 01DF 479 BRB PARTIAL_SUCCESS ;
51 0800 8F 3C 01E1 480 ERRFLK: MOVZWL #SS$_ACCONFLICT,R1 ; Generate SS code for RMS$_FLK
      0A 11 01E6 481 BRB PARTIAL_SUCCESS ;
51 08A8 8F 3C 01E8 482 ERRACC: MOVZWL #SS$_FICELOCKED,R1 ; Generate SS code for RMS$_ACC
      03 11 01ED 483 BRB PARTIAL_SUCCESS ;
      51 24 3C 01EF 484 ERRPRV: MOVZWL #SS$_NOPRIV,R1 ; Generate SS code for RMS$_PRV
01F2 485
01F2 486 :+
01F2 487 : Return partial success to the user on call to $SEARCH (i.e., STS = RMS$_SUC
01F2 488 : and STV = SS$_errorcode).
01F2 489 :-
01F2 490
01F2 491 ASSUME FAB$L_STS+4 EQ FAB$L_STV
01F2 492
01F2 493 PARTIAL_SUCCESS:
01F2 494 RMSSUC ; Return success
08 A8 50 7D 01F5 495 MOVQ R0,FAB$L_STS(R8) ; Update FAB$L_STS and FAB$L_STV
01F9 496
01F9 497 :+
01F9 498 : If FAL returned an error via a DAP Status message, then set a flag to denote
01F9 499 : that the next call to $SEARCH must send a DAP Continue Transfer message to
01F9 500 : restart the DAP directory-list operation at the remote node!
01F9 501 :-
01F9 502
30 A7 91 01F9 503 FAIL4: CMPB DAP$B_TYPE(R7),- ; Branch if reply from FAL was not a
      09 01FC 504 #DAP$R_STS_MSG ; Status message
      04 12 01FD 505 BNEQ 10$ ;
01FF 506 $SETBIT #NWA$V_FTM_CONT,(R7) ; Denote Continue Transfer msg required
      05 0203 507 10$: RSB ; Exit with RMS code in R0

```

```

0204 509 .SBTTL UPDATE_FAB
0204 510
0204 511 :++
0204 512 : Update the user FAB from the the Attributes message.
0204 513 :--
0204 514
0204 515 UPDATE_FAB: ; Entry point
0204 516
0204 517 :
0204 518 : Process the DAP ORG, MRN, BLS, RFM, and RAT fields.
0204 519 :
0204 520
1D A8 45 A7 90 0204 521 MOVB DAP$B_ORG(R7),FAB$B_ORG(R8)
00 45 A7 91 0209 522 CMPB DAP$B_ORG(R7),#DAP$K_SEQ; Branch if SEQ organization
10 45 A7 91 020D 523 BEQL 10$ ;
38 A8 58 A7 D0 0213 524 CMPB DAP$B_ORG(R7),#DAP$K_REL; Branch if not REL organization
3C A8 48 A7 B0 0215 525 BNEQ 20$ ;
1F A8 46 A7 90 021A 526 MOVL DAP$L_MRN(R7),FAB$L_MRN(R8)
FDD7' 30 021C 527 BRB 20$ ;
1E A8 47 A7 90 0221 528 10$: MOVW DAP$W_BLS(R7),FAB$W_BLS(R8)
0226 529 20$: MOVB DAP$B_RFM(R7),FAB$B_RFM(R8)
0229 530 BSBW NT$MOD_RAT ; Modify RAT bits returned from FAL
0229 531 ; as required
022E 532 MOVB DAP$B_RAT(R7),FAB$B_RAT(R8)
022E 533
022E 534 :
022E 535 : Process the DAP MRS, ALQ, BKS, FSZ, and DEQ fields.
022E 536 :
022E 537
36 A8 4A A7 B0 022E 538 MOVW DAP$W_MRS(R7),FAB$W_MRS(R8)
10 A8 4C A7 D0 0233 539 MOVL DAP$L_ALQ1(R7),FAB$L_ALQ(R8)
3E A8 50 A7 90 0238 540 MOVB DAP$B_BKS(R7),FAB$B_BKS(R8)
3F A8 51 A7 90 023D 541 MOVB DAP$B_FSZ(R7),FAB$B_FSZ(R8)
14 A8 54 A7 B0 0242 542 MOVW DAP$W_DEQ1(R7),FAB$W_DEQ(R8)
0247 543
0247 544 :
0247 545 : Process the DAP FOP field.
0247 546 :
0247 547 :
51 64 A7 D0 0247 548 MOVL DAP$L_FOP1(R7),R1 ; Get DAP FOP bits
52 D4 024B 549 CLRL R2 ; Clear resultant FOP bits
024D 550 $MAPBIT DAP$V_CTG,FAB$V_CTG ; Map CTG bit
0255 551 $MAPBIT DAP$V_CBT,FAB$V_CBT ; Map CBT bit
025D 552 $MAPBIT DAP$V_RCK,FAB$V_RCK ; Map RCK bit
0265 553 $MAPBIT DAP$V_WCK,FAB$V_WCK ; Map WCK bit
04 A8 00B00200 8F CA 026D 554 BICL2 #<<<FAB$M_CTG>>!- ; Clear FOP bits in user FAB
0275 555 <FAB$M_CBT>!- ; that may be updated
0275 556 <FAB$M_RCK>!-
0275 557 <FAB$M_WCK>!-
0275 558 0>,FAB$L_FOP(R8)
04 A8 52 C8 0275 559 BISL2 R2,FAB$L_FOP(R8) ; Update FOP field
05 0279 560 RSB ; Exit

```

```
027A 562 :++
027A 563 : Check for undefined record format returned from a 'stream-only' machine. If so,
027A 564 : convert UDF to STM in the RFM field and return CR in the RAT field.
027A 565 :--
027A 566
027A 567 MOD_RFM_RAT:
OE 67 32 E1 027A 568 -BBC ; Entry point
00 46 A7 91 027E 569 -CMPB #DAPSV_STM_ONLY,(R7),10$; Branch if not 'stream-only' machine
46 A7 08 12 0282 570 -BNEQ DAPSB_RFM(R7),#DAPSK_UDF; Since no Attributes message was sent
47 A7 02 90 0284 571 -MOVB 10$ ; to remote to specify record format
90 0288 572 -MOVB #DAPSK_STM,DAPSB_RFM(R7); (as done on open), convert UDF to STM
05 028C 573 10$: -RSB #DAPSM_CR,DAPSB_RAT(R7) ; Also declare implied carriage control
028D 574 ; Exit
028D 575 -END ; End of module
```

\$\$PSECT\_EP = 00000000  
\$\$COUNT = 00000003  
\$\$RMSTEST = 0000001A  
\$\$RMS\_PBUGCHK = 00000010  
\$\$RMS\_TBUGCHK = 00000008  
\$\$RMS\_UMODE = 00000004  
BUILD\_MASK = 0000003F R 01  
DAPSB\_ACCFUNC 00000040  
DAPSB\_ACCOPT 00000041  
DAPSB\_BITCNT 00000035  
DAPSB\_BKS 00000050  
DAPSB\_BSZ 00000052  
DAPSB\_CONFUNC 00000040  
DAPSB\_DATATYPE 00000044  
DAPSB\_DCODE\_FID 00000019  
DAPSB\_DCODE\_MAC 00000018  
DAPSB\_DCODE\_MSG 0000001A  
DAPSB\_DECVER 00000047  
DAPSB\_ECONUM 00000045  
DAPSB\_FAC 00000042  
DAPSB\_FILESYS 00000043  
DAPSB\_FLAGS 00000031  
DAPSB\_FSZ 00000051  
DAPSB\_LEN256 00000034  
DAPSB\_LENGTH 00000033  
DAPSB\_NAMETYPE 00000040  
DAPSB\_ORG 00000045  
DAPSB\_OSTYPE 00000042  
DAPSB\_RAT 00000047  
DAPSB\_RFM 00000046  
DAPSB\_SHR 00000043  
DAPSB\_STREAMID 00000032  
DAPSB\_TYPE 00000030  
DAPSB\_USRNUM 00000046  
DAPSB\_USRVER 00000048  
DAPSB\_VERNUM 00000044  
DAPSB\_X\_FIELD 00000024  
DAPSC\_BLN 000000C0  
DAPSK\_ACC\_MSG = 00000003  
DAPSK\_ACK\_MSG = 00000006  
DAPSK\_ATT\_MSG = 00000002  
DAPSK\_BLN 000000C0  
DAPSK\_CMP\_MSG = 00000007  
DAPSK\_CON\_MSG = 00000005  
DAPSK\_DIR\_LIST = 00000006  
DAPSK\_FIX = 00000001  
DAPSK\_NAM\_MSG = 0000000F  
DAPSK\_PRO\_MSG = 0000000E  
DAPSK\_REL = 00000010  
DAPSK\_SEQ = 00000000  
DAPSK\_SKIP\_REC = 00000002  
DAPSK\_STM = 00000004  
DAPSK\_STS\_MSG = 00000009  
DAPSK\_SUM\_MSG = 0000000C  
DAPSK\_TIM\_MSG = 0000000D  
DAPSK\_UDF = 00000000  
DAPSL\_ALQ1 0000004C

DAPSL\_ATTMENU 00000040  
DAPSL\_CMWA 00000030  
DAPSL\_CRC\_RSLT 00000020  
DAPSL\_DCODE\_STS 00000018  
DAPSL\_DEV 00000068  
DAPSL\_EBK 00000078  
DAPSL\_FOP1 00000064  
DAPSL\_HBK 00000074  
DAPSL\_MRN 00000058  
DAPSL\_MSG\_MASK 0000001C  
DAPSL\_SBN 0000007C  
DAPSL\_SSPWA 00000080  
DAPSL\_TEMP 00000090  
DAPSM\_BITCNT = 00000008  
DAPSM\_CMPFMT = 00000008  
DAPSM\_CR = 00000002  
DAPSM\_DFTSPEC = 00000010  
DAPSM\_DMO = 00002000  
DAPSM\_DSP\_3NAM = 00000200  
DAPSM\_DSP\_ALL = 00000004  
DAPSM\_DSP\_ATT = 00000001  
DAPSM\_DSP\_KEY = 00000002  
DAPSM\_EMBEDDED = 00000010  
DAPSM\_GET = 00000002  
DAPSM\_GO\_NOGO = 00000010  
DAPSM\_IMAGE = 00000002  
DAPSM\_LSA = 00000040  
DAPSM\_MACY11 = 00000080  
DAPSM\_MSE = 00000010  
DAPSM\_NONFATAL = 00000001  
DAPSM\_SEGMENT = 00000040  
DAPSM\_TMP1\$ = 00000020  
DAPSM\_TMP2\$ = 000000C0  
DAPSM\_TMP3\$ = 00020000  
DAPSM\_TMP4\$ = 01000000  
DAPSM\_TMP5\$ = F0000000  
DAPSM\_ZERO = 00000080  
DAPSQ\_DCODE\_FLG 00000000  
DAPSQ\_FILESPEC 00000044  
DAPSQ\_MSG\_BUF1 00000008  
DAPSQ\_MSG\_BUF2 00000010  
DAPSQ\_NAMESPEC 00000044  
DAPSQ\_PASSWORD 00000050  
DAPSQ\_RUNSYS 0000005C  
DAPSQ\_SYSCAP 00000028  
DAPSQ\_SYSPEC 00000038  
DAPSV\_CBT = 00000017  
DAPSV\_CTG = 00000007  
DAPSV\_DFTSPEC = 00000004  
DAPSV\_DIRECTORY = 00000019  
DAPSV\_DIRNAME = 00000002  
DAPSV\_DSP\_ATT = 00000000  
DAPSV\_DSP\_PRO = 00000005  
DAPSV\_DSP\_SUM = 00000003  
DAPSV\_DSP\_TIM = 00000004  
DAPSV\_FILENAME = 00000001  
DAPSV\_FILSPEC = 00000000



NTOSEARCH  
Symbol table

NETWORK SEARCH FILES

J 8

16-SEP-1984 00:03:05 VAX/VMS Macro V04-00  
5-SEP-1984 16:21:09 [RMS.SRC]NTOSEARCH.MAR;1

DAPSV_FLK	=	00000006		
DAPSV_GEQ_V70	=	00000026		
DAPSV_RCK	=	0000000F		
DAPSV_STM_ONLY	=	00000032		
DAPSV_VOLNAME	=	00000003		
DAPSV_WCK	=	0000000E		
DAPSW_BLS		00000048		
DAPSW_BUFSIZ		00000040		
DAPSW_DEQ1		00000054		
DAPSW_DISPLAY1		0000004C		
DAPSW_FFB		00000072		
DAPSW_LRL		00000070		
DAPSW_MRS		0000004A		
DAPSW_PARTNER		00000006		
DAPSW_VERSION		00000004		
DIR_NAM		000000CE	R	01
ERRACC		000001E8	R R	01
ERRATR		000001DA	R R	01
ERRFLK		000001E1	R	01
ERRPRV		000001EF	R	01
FABSB_BKS	=	0000003E		
FABSB_FSZ	=	0000003F		
FABSB_ORG	=	0000001D		
FABSB_RAT	=	0000001E		
FABSB_RFM	=	0000001F		
FABSL_ALQ	=	00000010		
FABSL_FOP	=	00000004		
FABSL_MRN	=	00000038		
FABSL_NAM	=	00000028		
FABSL_STS	=	00000008		
FABSL_STV	=	0000000C		
FABSM_CBT	=	00200000		
FABSM_CTG	=	00100000		
FABSM_RCK	=	00800000		
FABSM_WCK	=	00000200		
FABSV_CBT	=	00000015		
FABSV_CTG	=	00000014		
FABSV_RCK	=	00000017		
FABSV_WCK	=	00000009		
FABSW_BLS	=	0000003C		
FABSW_DEQ	=	00000014		
FABSW_MRS	=	00000036		
FAIL1		0000003E	R	01
FAIL2		00000187	R	01
FAIL3		0000019F	R	01
FAIL4		000001F9	R	01
FIL_NAM		000000EA	R	01
FWASQ_QUOTED	=	00000170		
FWASV_REMRESULT	=	00000035		
IFBSL_NWA_PTR	=	0000003C		
LOOP		0000015B	R	01
MERGE_NAM		000000A1	R	01
MOD_RFM_RAT		0000027A	R	01
NAMSB_NOP	=	00000008		
NAMSV_SRCHXABS	=	00000006		
NTSBUG_NAMETYPE		*****	X	01
NTSBUILD_HEAD		*****	X	01

NTSBUILD_TAIL	*****	X	01
NTSCVT_BR4_EXT	*****	X	01
NTSDECODE_PRO	*****	X	01
NTSDECODE_SUM	*****	X	01
NTSDECODE_TIM	*****	X	01
NTSEXCH_CNF	*****	X	01
NTSGET_FILESPEC	*****	X	01
NTSMOD_RAT	*****	X	01
NTSRECEIVE	*****	X	01
NTSRMT_ACCFUNC	*****	X	01
NTSSCAN_XABCHN	*****	X	01
NTSSEARCH	00000000	RG	01
NTSTRANSMIT	*****	X	01
NTSUPDATE_FHC	*****	X	01
NWASB_ALLXABCNT	0000011C		
NWASB_DAP_RAC	000000C9		
NWASB_FILESYS	000000C5		
NWASB_KEYXABCNT	0000011D		
NWASB_NETSTRSIZ	0000016F		
NWASB_NODBUFSIZ	00000168		
NWASB_ORG	000000C6		
NWASB_OSTYPE	000000C4		
NWASB_RFM	000000C7		
NWASB_RMS_RAC	000000C8		
NWASC_BLN	00000800		
NWASK_BLN	00000800		
NWASL_ALLXABADR	00000100		
NWASL_DATXABADR	00000104		
NWASL_DEV	000000C0		
NWASL_FHCXABADR	00000108		
NWASL_KEYXABADR	0000010C		
NWASL_MSG_MASK	000000D4		
NWASL_PROXABADR	00000110		
NWASL_RDTXABADR	00000114		
NWASL_SAVE_FLGS	00000128		
NWASL_SUMXABADR	00000118		
NWASL_THREAD	000000FC		
NWASL_XLTATTR	00000238		
NWASL_XLTBUFFLG	0000022C		
NWASL_XLTCNT	00000228		
NWASL_XLTMAXINDX	00000234		
NWASL_XLTSIZ	00000230		
NWASQ_ACS	00000244		
NWASQ_BIGBUF	00000170		
NWASQ_BLD	000000FC		
NWASQ_FLG	00000000		
NWASQ_INODE	0000025C		
NWASQ_IOSB	000000D8		
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		

NTOSEARCH  
Symbol table

NETWORK SEARCH FILES

K 8

16-SEP-1984 00:08:05  
5-SEP-1984 16.21:09

VAX/VMS Macro V04-00  
[RMS.SRC]NTOSEARCH.MAR;1

Page 15  
(8)

RM  
VO

NWAST_AUXBUF	000005E0		
NWAST_DAP	00000000		
NWAST_INODEBUF	000004AC		
NWAST_ITM_ATTR	00000200		
NWAST_ITM_END	00000224		
NWAST_ITM_LST	00000200		
NWAST_ITM_MAXINDX	00000218		
NWAST_ITM_STRING	0000020C		
NWAST_NCBBUF	0000052C		
NWAST_NODEBUF	00000169		
NWAST_RCVBUF	000001A0		
NWAST_SCAN	00000100		
NWAST_TEMP	00000120		
NWAST_XLTBUF1	000002AC		
NWAST_XLTBUF2	000003AC		
NWAST_XMTBUF	000003C0		
NWASV_FOP_FLK	= 00000012		
NWASV_FTM_CONT	= 0000001E		
NWASV_LAST_MSG	= 00000000		
NWASW_BUILD	000000D2		
NWASW_DAPBUFSIZ	000000CA		
NWASW_DIR_OFF	000000CC		
NWASW_DISPLAY	000000D0		
NWASW_FIL_OFF	000000CE		
NWASW_JNLXABJOP	0000011E		
PARTIAL_SUCCESS	000001F2	R	01
PIOSA_TRACE	*****	X	01
RCV_ACK	00000188	R	01
RCV_ATT	00000100	R R	01
RCV_EXT_ATT	00000137	R R	01
RCV_NAM	00000087	R	01
RMSS_ACC	= 0001C002		
RMSS_ATR	= 0001C0CC		
RMSS_FLK	= 0001828A		
RMSS_NET	= 0001874C		
RMSS_NMF	= 000182CA		
RMSS_PRV	= 0001829A		
SEARCH_ENDED	00000182	R	01
SEARCH_SUCCESS	00000188	R	01
SEND_ACC	0000005F	R	01
SEND_CNF	0000002D	R	01
SEND_CON	00000015	R	01
SSS_ACCONFLICT	= 00000800		
SSS_FCPREADERR	= 00000888		
SSS_FILELOCKED	= 000008A8		
SSS_NOPRIV	= 00000024		
TPTSL_NTOSEARCH	*****	X	01
UPDATE_FAB	00000204	R	01
VOL_NAM	000000BC	R	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
NFSNETWORK	00000280 ( 653.)	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	35	00:00:00.08	00:00:01.13
Command processing	132	00:00:00.67	00:00:06.44
Pass 1	492	00:00:20.58	00:01:02.81
Symbol table sort	0	00:00:03.04	00:00:06.33
Pass 2	120	00:00:03.54	00:00:08.63
Symbol table output	33	00:00:00.28	00:00:00.88
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	816	00:00:28.21	00:01:26.24

The working set limit was 1650 pages.  
111320 bytes (218 pages) of virtual memory were used to buffer the intermediate code.  
There were 110 pages of symbol table space allocated to hold 2101 non-local and 30 local symbols.  
575 source lines were read in Pass 1, producing 15 object records in Pass 2.  
33 pages of virtual memory were used to define 32 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	23
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	28

2428 GETS were required to define 28 macros.

There were no errors, warnings or information messages.

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



The image displays a grid of 100 small terminal window screenshots, arranged in 10 rows and 10 columns. Each window shows a different system utility or command output, with some windows containing the following labels:

- NTOPEN LIS
- NT0SCNXAB LIS
- NT0RENAME LIS
- RM0CACHE LIS
- RM0BUFMR LIS
- NT0SEARCH LIS
- NT0PUT LIS
- RM0ACCESS LIS

The screenshots are too small to read the specific text within them, but they appear to be various system utilities or command outputs.