



```

NN      NN      TTTTTTTTTT      000000      NN      NN      WW      WW      AAAAAA      SSSSSSSS      EEEEEEEFE      TTTTTTTTTT
NN      NN      TTTTTTTTTT      000000      NN      NN      WW      WW      AAAAAA      SSSSSSSS      FEEEEEEEEE      TTTTTTTTTT
NN      NN      TT          00          00      NN      NN      WW      WW      AA          AA      SS          EE          TT
NN      NN      TT          00          00      NN      NN      WW      WW      AA          AA      SS          EE          TT
NNNN     NN      TT          00          0000     NNNN     NN      WW      WW      AA          AA      SS          EE          TT
NNNN     NN      TT          00          0000     NNNN     NN      WW      WW      AA          AA      SS          EE          TT
NN      NN      NN      TT          00      00      00      NN      NN      NN      WW      WW      AA          AA      SSSSSS      EEEEEEEE      TT
NN      NN      NN      TT          00      00      00      NN      NN      NN      WW      WW      AA          AA      SSSSSS      EEEEEEEE      TT
NN      NNNN     TT          0000      00      NN      NNNN     WW      WW      WWWW     AAAAAAAAAA      SS          EE          TT
NN      NNNN     TT          0000      00      NN      NNNN     WW      WW      WWWW     AAAAAAAAAA      SS          EE          TT
NN      NN      TT          00          00      NN      NN      WWWW     WWWW     AA          AA      SS          EE          TT
NN      NN      TT          00          00      NN      NN      WWWW     WWWW     AA          AA      SS          EE          TT
NN      NN      TT          000000      NN      NN      WW      WW      AA          AA      SSSSSSSS      EEEEEEEEEE      TT
NN      NN      TT          000000      NN      NN      WW      WW      AA          AA      SSSSSSSS      EEEEEEEEEE      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

```

(3)	74
(4)	100
(5)	195

DECLARATIONS  
NTSNWA\_INIT, Allocate and Initialize NWA  
NTSNWA\_FREE, Deallocate NWA

```
0000 1          $BEGIN NTONWASET,000,NFS$NETWORK,<NWA SETUP AND RELEASE>
0000 2
0000 3
0000 4 :*****
0000 5 :*
0000 6 :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 :*  ALL RIGHTS RESERVED.
0000 9 :*
0000 10 :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 :*  TRANSFERRED.
0000 16 :*
0000 17 :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 :*  CORPORATION.
0000 20 :*
0000 21 :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 :*
0000 24 :*
0000 25 :*****
0000 26 :
```

```
0000 28 :++
0000 29 : Facility: RMS
0000 30 :
0000 31 : Abstract:
0000 32 :
0000 33 :     This module is responsible for the allocation, initialization, and
0000 34 :     deallocation of the Network Work Area (NWA) control block.
0000 35 :
0000 36 : Environment: VAX/VMS, executive mode
0000 37 :
0000 38 : Author: James A. Krycka,      Creation Date: 13-JAN-1978
0000 39 :
0000 40 : Modified By:
0000 41 :
0000 42 :     V03-009 RAS02553      Ron Schaefer      15-Feb-1984
0000 43 :     Initialize the $TRNLNM item-list.
0000 44 :
0000 45 :     V03-008 KB10570      Keith B. Thompson      29-Jul-1983
0000 46 :     Remove ref to ifb$V_nwa
0000 47 :
0000 48 :     V03-007 JAK0104      J A Krycka      22-APR-1983
0000 49 :     Modify NT$NWA_INJT and NT$NWA_FREE to handle large DAP buffers
0000 50 :     described by NWA$Q_BIGBUF.
0000 51 :
0000 52 :     V03-006 KRM0059      K Malik      20-Aug-1982
0000 53 :     Clear IFB$V_NWA bit and IFB$L_NWA_PTR in NT$NWA_FREE.
0000 54 :
0000 55 :     V03-005 KRM0055      K Malik      10-Aug-1982
0000 56 :     Changed NWA$B_UNDERLINE to FWASB_UNDER_NOD.
0000 57 :     Moved old NWA$T_NODEBUF & NWA$T_QUOTEDBUF initialization
0000 58 :     to RMOFWASET. Initialize NWA$B_NODBUFSIZ
0000 59 :
0000 60 :     V02-004 JAK0067      J A Krycka      26-OCT-1981
0000 61 :     Initialize NWA$Q_FLG field.
0000 62 :
0000 63 :     V02-003 SPR37751      J A Krycka      10-JUN-1981
0000 64 :     If a receive QIO is outstanding, process (stall on) the AST
0000 65 :     before deallocating the NWA to avoid a BUGCHECK when the AST
0000 66 :     is delivered.
0000 67 :
0000 68 :     V02-002 JAK0058      J A Krycka      22-MAY-1981
0000 69 :     This module was created from code previously residing in
0000 70 :     RMOXPFN and RMSOCLOSE.
0000 71 :
0000 72 :--
```

```
0000 74      .SBTTL  DECLARATIONS
0000 75
0000 76 :
0000 77 : Include Files:
0000 78 :
0000 79 :
0000 80      $IFBDEF      ; Define IFAB symbols
0000 81      $LNMDEF      ; Define LNM symbols
0000 82      $NWADEF      ; Define Network Work Area symbols
0000 83
0000 84 :
0000 85 : Macros:
0000 86 :
0000 87 :      None
0000 88 :
0000 89 : Equated Symbols:
0000 90 :
0000 91 :
0000 92      ASSUME  NWA$Q_FLG EQ 0
0000 93
0000 94 :
0000 95 : Own Storage:
0000 96 :
0000 97 :      None
0000 98 :
```

```

0000 100      .SBTTL NTSNWA_INIT, Allocate and Initialize NWA
0000 101
0000 102 :++
0000 103 : NTSNWA_INIT - allocates space for a NWA control block and then initializes
0000 104 :   selected fields in it and in the FWA.
0000 105 :
0000 106 : Calling Sequence:
0000 107 :
0000 108 :     BSBW  NTSNWA_INIT
0000 109 :
0000 110 : Input Parameters:
0000 111 :
0000 112 :     R8      FAB address
0000 113 :     R9      IFAB address
0000 114 :     R10     FWA address
0000 115 :     R11     Impure Area address
0000 116 :
0000 117 : Implicit Inputs:
0000 118 :
0000 119 :     None
0000 120 :
0000 121 : Output Parameters:
0000 122 :
0000 123 :     R0      Status code (RMS)
0000 124 :     R1      Destroyed
0000 125 :     R2      NWA size in bytes (rounded to page boundary)
0000 126 :     R3      NWA address
0000 127 :
0000 128 : Implicit Outputs:
0000 129 :
0000 130 :     FWASQ_NODE
0000 131 :     FWASQ_QUOTED
0000 132 :     IFBSL_NWA_PTR
0000 133 :     NWA fields
0000 134 :
0000 135 : Completion Codes:
0000 136 :
0000 137 :     Standard RMS completion codes
0000 138 :
0000 139 : Side Effects:
0000 140 :
0000 141 :     Space is obtained from the RMS buffer manager.
0000 142 :
0000 143 : --
0000 144 :
0000 145 NTSNWA_INIT::      ; Entry point
0000 146 :
0000 147 :
0000 148 : Allocate space for the Network Work Area (NWA) control block and make IFAB
0000 149 : point to it.
0000 150 :
0000 151 :
52  0800 8F  3C 0000 152      MOVZWL #NWASC_BLN,R2      ; Specify size of NWA in bytes
      FFF8  30 0005 153      BSBW  RMSGETPAG          ; Allocate space (NOT ZEROED)
      63 50  E9 0008 154      BLBL  R0,EXIT           ; Branch on failure
3C  A9  53  D0 000B 155      MOVL  R3,IFBSL_NWA_PTR(R9) ; Store NWA address in IFAB
000F 156
    
```

```

000F 157 :
000F 158 : Initialize selected fields in NWA.
000F 159 :
000F 160 :
000F 161 ASSUME NWA$Q_XLTBUF2 EQ NWA$Q_XLTBUF1+8
000F 162 ASSUME NWA$T_ITM_ATTR EQ NWA$T_ITM_LST
000F 163 ASSUME NWA$T_ITM_STRING EQ NWA$T_ITM_ATTR+12
000F 164 ASSUME NWA$T_ITM_MAXINDX EQ NWA$T_ITM_STRING+12
000F 165 ASSUME NWA$T_ITM_END EQ NWA$T_ITM_MAXINDX+12
000F 166 ASSUME NWA$L_XLTBUFFLG EQ NWA$L_XLTCNT+4
000F 167
51 024C C3 7C 000F 168 CLRQ NWA$Q_FLG(R3) ; Initialize status flags
81 024C C3 7E 0011 169 MOVAQ NWA$Q_XLTBUF1(R3),R1 ; Initialize descriptors for use during
81 FF 8F 9A 0016 170 ; logical node name translation
81 02AC C3 9E 001A 171 MOVZBL #NWA$C_XLTBUFSIZ,(R1)+ ; Set-up 1st translation buffer
81 FF 8F 9A 001F 172 MOVAB NWA$T_XLTBUF1(R3),(R1)+ ;
61 03AC C3 9E 0023 173 MOVZBL #NWA$C_XLTBUFSIZ,(R1)+ ; Set-up 2nd translation buffer
026C C3 9E 0028 174 MOVAB NWA$T_XLTBUF2(R3),(R1)+ ;
0248 C3 002C 175 MOVAB NWA$T_ACSBUF(R3),- ; Initialize address for access
0238 C3 D4 002F 176 NWA$Q_ACS+4(R3) ; control string descriptor
0228 C3 7C 0033 177 CLRQ NWA$L_XLTATTR(R3) ; Initialize node translation attributes
0037 178 CLRQ NWA$L_XLTCNT(R3) ; Initialize node translation counter
81 0200 C3 9E 0037 179 ; and translation buffer flag
81 00030004 8F D0 003C 180 MOVAB NWA$T_ITM_LST(R3),R1 ; Initialize $TRNLNM itemlist
81 0238 C3 9E 0043 181 MOVL #<<LNMS_ATTRIBUTES@16>!4>,(R1)+ ; Translation attributes
81 00200FF 8F D4 0048 182 MOVAB NWA$L_XLTCNT(R3),(R1)+ ; Place for attributes
81 00200FF 8F D0 004A 183 CLRQ (R1)+ ; no return length
81 0230 C3 9E 0051 184 MOVL #<<LNMS_STRING@16>!NWA$C_XLTBUFSIZ>,(R1)+ ; Buffer addr setup later
81 00070004 8F D4 0053 185 CLRQ (R1)+ ; Return length
81 0234 C3 9E 0058 186 MOVAB NWA$L_XLTSIZ(R3),(R1)+ ; Max # of translations
81 0234 C3 9E 005F 187 MOVL #<<LNMS_MAX_INDEX@16>!4>,(R1)+ ; Place for count
0168 C3 81 7C 0064 188 CLRQ (R1)+ ; No return len and eol
0170 C3 94 0066 189 CLRQ NWA$B_NODBUFSIZ(R3) ; Initialize NODEBUF size field
006E 190 CLRQ NWA$Q_BIGBUF(R3) ; Zero big DAP buffer descriptor to
006E 191 ; indicate that buffer does not exist
05 006E 192 ; Exit with RMS code in R0
05 006E 193 EXIT: RSB

```



```

006F 195 .SBTTL NT$NWA_FREE, Deallocate NWA
006F 196
006F 197 :++
006F 198 : NT$NWA_FREE - deallocates the NWA control block after all receive QIO ASTs
006F 199 : hav~ been processed.
006F 200 :
006F 201 : Calling Sequence:
006F 202 :
006F 203 : BSBW NT$NWA_FREE
006F 204 :
006F 205 : Input Parameters:
006F 206 :
006F 207 : R8 FAB address
006F 208 : R9 IFAB address
006F 209 : R10 IFAB address
006F 210 : R11 Impure Area address
006F 211 :
006F 212 : Implicit Inputs:
006F 213 :
006F 214 : IFBSL_NWA_PTR
006F 215 : N$WASV_RCVQIO
006F 216 : N$WASV_RCVAST
006F 217 :
006F 218 : Output Parameters:
006F 219 :
006F 220 : R0-R5 Destroyed
006F 221 : AP Destroyed
006F 222 :
006F 223 : Implicit Outputs:
006F 224 :
006F 225 : NWA is deallocated
006F 226 :
006F 227 : Completion Codes:
006F 228 :
006F 229 : None
006F 230 :
006F 231 : Side Effects:
006F 232 :
006F 233 : Space is returned to the RMS buffer manager.
006F 234 :
006F 235 :--
006F 236 :
006F 237 NT$NWA_FREE:: ; Entry point
54 3C A9 D0 006F 238 MOVL IFBSL_NWA_PTR(R9),R4 ; Get NWA address
39 13 0073 239 BEQL 20$ ; Branch if none
0075 240
0075 241 :
0075 242 : If a special receive AST is still pending, then stall on it (i.e., wait for it
0075 243 : to complete) before deallocating the NWA. A special receive QIO is one posted
0075 244 : by NT$TRANSMIT_PKT that references the NWA and uses NT$STALLAST instead of
0075 245 : RMSSTALLAST.
0075 246 :
0075 247 :
13 64 03 E1 0075 248 BBC #N$WASV_RCVQIO,(R4),10$ ; Branch if special receive not posted
OF 64 04 E0 0079 249 BBS #N$WASV_RCVAST,(R4),10$ ; Branch if special received completed
007D 250 $SETBIT #N$WASV_RCVSTALL,(R4) ; Set flag to resume thread after stall
00FC C4 59 D0 0081 251 MOVL R9,N$WASL_THREAD(R4) ; Save IFAB/IRAB address that we are

```

```

00000000'EF 16 0086 252 ; stalling on for use by NT$STALLAST
0086 253 ; before it branches to RM$THREADGO
0086 254 JSB RMSSTALL ; Await completion of special receive
008C 255 ; Note: R0 contains garbage on return
008C 256
008C 257
008C 258 ; Now it's safe to deallocate the NWA and associated DAP buffers.
008C 259
008C 260
0174 C4 D5 008C 261 10$: TSTL NWA$Q_BIGBUF+4(R4) ; Branch if big DAP buffers have not
11 13 0090 262 BEQL 15$ ; been allocated
55 0170 C4 D0 0092 263 MOVL NWA$Q_BIGBUF(R4),R5 ; Get # bytes to return
54 0174 C4 D0 0097 264 MOVL NWA$Q_BIGBUF+4(R4),R4 ; Get address of buffer area
FF61' 30 009C 265 BSBW RMSRETPAG ; Deallocate space used by buffer area
54 3C A9 D0 009F 266 MOVL IFB$NWA_PTR(R9),R4 ; Get NWA address
55 0800 BF 3C 00A3 267 15$: MOVZWL #NWA$C_BLN,R5 ; Get # bytes to return
FF55' 30 00AB 268 BSBW RMSRETPAG ; Deallocate space used by NWA
3C A9 D4 00AB 269 CLRL IFB$NWA_PTR(R9) ; clear in case NWA is deallocated
00AE 270 ; without deallocating the IFB
05 00AE 271 20$: RSB ; Exit with no status in R0
UCAF 272
00AF 273 .END ; End of module

```

NTONWASET  
Symbol table

NWA SETUP AND RELEASE

C 1

16-SEP-1984 00:03:13 VAX/VMS Macro V04-00  
5-SEP-1984 16:20:56 [RMS.SRC]NTONWASET.MAR;1

```

SS.PSECT_EP      = 00000000
SSRMSTEST       = 0000001A
SSRMS_PBUGCHK   = 00000010
SSRMS_TBUGCHK   = 00000008
SSRMS_UMODE     = 00000004
EXIT            = 0000006E R    01
IFBSL_NWA_PTR   = 0000003C
LNMS_ATTRIBUTES = 00000003
LNMS_MAX_INDEX  = 00000007
LNMS_STRING     = 00000002
NTSNWA_FREE     = 0000006F RG   01
NTSNWA_INIT     = 00000000 RG   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
NWASC_XLTBUFSIZ = 000000FF
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       = 000000F0
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
NWAAT_ACSBUF    = 0000026C
NWAAT_AUXBUF    = 000005E0

```

```

NWAAT_DAP       = 00000000
NWAAT_INODEBUF  = 000004AC
NWAAT_ITM_ATTR  = 00000200
NWAAT_ITM_END   = 00000224
NWAAT_ITM_LST   = 00000200
NWAAT_ITM_MAXINDX = 00000218
NWAAT_ITM_STRING = 0000020C
NWAAT_NCBBUF    = 0000052C
NWAAT_NODEBUF   = 00000169
NWAAT_RCVBUF    = 000001A0
NWAAT_SCAN      = 00000100
NWAAT_TEMP      = 00000120
NWAAT_XLTBUF1   = 000002AC
NWAAT_XLTBUF2   = 000003AC
NWAAT_XMTBUF    = 000003C0
NWAAT_V_RCVAST  = 00000004
NWAAT_V_RCVQIO  = 00000003
NWAAT_V_RCVSTALL = 00000005
NWAAT_W_BUILD   = 000000D2
NWAAT_W_DAPBUFSIZ = 000000CA
NWAAT_W_DIR_OFF = 000000CC
NWAAT_W_DISPLAY = 000000D0
NWAAT_W_FIL_OFF = 000000CE
NWAAT_W_JNLXABJOP = 0000011E
RMSGETPAG       = ***** X 01
RMSRETPAG       = ***** X 01
RMSSTALL        = ***** X 01

```

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

PSECT name	Allocation	PSECT No.	Attributes
. ABS	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
NFS\$NETWORK	000000AF ( 175.)	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	32	00:00:00.09	00:00:01.33
Command processing	111	00:00:00.61	00:00:03.53
Pass 1	181	00:00:04.33	00:00:12.99
Symbol table sort	0	00:00:00.44	00:00:00.97
Pass 2	61	00:00:00.96	00:00:03.36
Symbol table output	12	00:00:00.08	00:00:00.08
Psect synopsis output	2	00:00:00.02	00:00:00.10
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	401	00:00:06.54	00:00:22.38

The working set limit was 1200 pages.  
20043 bytes (40 pages) of virtual memory were used to buffer the intermediate code.  
There were 20 pages of symbol table space allocated to hold 334 non-local and 4 local symbols.  
273 source lines were read in Pass 1, producing 13 object records in Pass 2.  
14 pages of virtual memory were used to define 13 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	4
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	9

434 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

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

The image displays a grid of 120 small screenshots, arranged in 10 rows and 12 columns. Each screenshot shows a different VMS utility or system output. Several screenshots are highlighted with larger text labels:

- Row 2, Column 3: NT0DAPRMS LIS
- Row 4, Column 10: NT0GET LIS
- Row 4, Column 12: NT0NWASET LIS
- Row 5, Column 10: NT0EXTEND LIS
- Row 7, Column 3: NT0DECODE LIS
- Row 8, Column 5: NT0ENCODE LIS
- Row 8, Column 6: NT0ERASE LIS
- Row 9, Column 4: NT0DISCON LIS
- Row 9, Column 5: NT0DISPLY LIS
- Row 10, Column 10: NT0MISC LIS

NTOPEN LIS									

NTOPEN LIS

NTSEARCH LIS

NTSEARCH LIS

RMOCACHE LIS

RMOCACHE LIS

RMOCACHE LIS

RMOCACHE LIS