

NNN		NNN	MMM		MMM	LLL
NNN		NNN	MMM		MMM	LLL
NNN		NNN	MMM		MMM	LLL
NNN		NNN	MMMMMM		MMMMMM	LLL
NNN		NNN	MMMMMM		MMMMMM	LLL
NNN		NNN	MMMMMM		MMMMMM	LLL
NNNNNN		NNN	MMM	MMM	MMM	LLL
NNNNNN		NNN	MMM	MMM	MMM	LLL
NNNNNN		NNN	MMM	MMM	MMM	LLL
NNN	NNN	NNN	MMM		MMM	LLL
NNN	NNN	NNN	MMM		MMM	LLL
NNN	NNN	NNN	MMM		MMM	LLL
NNN		NNNNNN	MMM		MMM	LLL
NNN		NNNNNN	MMM		MMM	LLL
NNN		NNNNNN	MMM		MMM	LLL
NNN		NNN	MMM		MMM	LLL
NNN		NNN	MMM		MMM	LLL
NNN		NNN	MMM		MMM	LLL
NNN		NNN	MMM		MMM	LLLLLLLLLLLLLLLL
NNN		NNN	MMM		MMM	LLLLLLLLLLLLLLLL
NNN		NNN	MMM		MMM	LLLLLLLLLLLLLLLL

```

NN      NN  MM      MM  LL      P P P P P P P P  UU      UU  R R R R R R R R  G G G G G G G G  E E E E E E E E
NN      NN  MM      MM  LL      P P P P P P P P  UU      UU  R R R R R R R R  G G G G G G G G  E E E E E E E E
NN      NN  M M M M  M M M M  LL      P P      P P  UU      UU  R R      R R  G G      G G  E E      E E
NN      NN  M M M M  M M M M  LL      P P      P P  UU      UU  R R      R R  G G      G G  E E      E E
N N N N  NN  MM      MM  LL      P P      P P  UU      UU  R R      R R  G G      G G  E E      E E
N N N N  NN  MM      MM  LL      P P      P P  UU      UU  R R      R R  G G      G G  E E      E E
NN      NN  NN      NN  MM      MM  LL      P P P P P P P P  UU      UU  R R R R R R R R  G G      G G  E E      E E
NN      NN  NN      NN  MM      MM  LL      P P P P P P P P  UU      UU  R R R R R R R R  G G      G G  E E      E E
NN      NN  NN      NN  MM      MM  LL      P P      P P  UU      UU  R R      R R  G G      G G  E E      E E
NN      NN  NN      NN  MM      MM  LL      P P      P P  UU      UU  R R      R R  G G      G G  E E      E E
NN      NN  NN      NN  MM      MM  LL      P P      P P  UU      UU  R R      R R  G G      G G  E E      E E
NN      NN  NN      NN  MM      MM  L L L L L L L L L L  P P      P P  U U U U U U U U  R R      R R  G G      G G  E E      E E
NN      NN  NN      NN  MM      MM  L L L L L L L L L L  P P      P P  U U U U U U U U  R R      R R  G G      G G  E E      E E

```

```

LL      I I I I I I  S S S S S S S S
LL      I I I I I I  S S S S S S S S
LL      I I      S S
LL      I I      S S
LL      I I      S S
LL      I I      S S
LL      I I      S S S S S S
LL      I I      S S S S S S
LL      I I      S S
LL      I I      S S
LL      I I      S S
LL      I I      S S
L L L L L L L L L L  I I I I I I  S S S S S S S S
L L L L L L L L L L  I I I I I I  S S S S S S S S

```

```

1 0001 0 %TITLE 'NML PURGE permanent parameter module'
2 0002 0 MODULE NML$PURGE (
3 0003 0     LANGUAGE (BLISS32),
4 0004 0     ADDRESSING_MODE (NONEXTERNAL=GENERAL),
5 0005 0     ADDRESSING_MODE (EXTERNAL=GENERAL),
6 0006 0     IDENT = 'V04-000'
7 0007 0 ) =
8 0008 1 BEGIN
9 0009 1
10 0010 1 *****
11 0011 1 *
12 0012 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
13 0013 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
14 0014 1 *  ALL RIGHTS RESERVED.
15 0015 1 *
16 0016 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
17 0017 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
18 0018 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
19 0019 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
20 0020 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
21 0021 1 *  TRANSFERRED.
22 0022 1 *
23 0023 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
24 0024 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
25 0025 1 *  CORPORATION.
26 0026 1 *
27 0027 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
28 0028 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
29 0029 1 *
30 0030 1 *****
31 0031 1 *****
32 0032 1
33 0033 1
34 0034 1 **
35 0035 1 FACILITY:  DECnet-VAX V2.0 Network Management Listener
36 0036 1
37 0037 1 ABSTRACT:
38 0038 1
39 0039 1     This module contains routines for processing NCP PURGE commands.
40 0040 1
41 0041 1 ENVIRONMENT:  VAX/VMS Operating System
42 0042 1
43 0043 1 AUTHOR:  Distributed Systems Software Engineering
44 0044 1
45 0045 1 CREATION DATE:  30-DEC-1979
46 0046 1
47 0047 1 MODIFIED BY:
48 0048 1
49 0049 1     V03-006 MKP0006      Kathy Perko      2-July-1984
50 0050 1     If doing a PURGE KNOWN NODES, and there is no executor
51 0051 1     don't return an error message to NCP.
52 0052 1
53 0053 1     V03-005 MKP0005      Kathy Perko      5-Feb-1984
54 0054 1     Fix key initialization for PURGE KNOWN entities.
55 0055 1
56 0056 1     V03-004 MKP0004      Kathy Perko      4-Aug-1983
57 0057 1     Make changes to change the node permanent database to use

```

```

: 58      0058 1 | multiple ISAM keys. This is for performance reasons.
: 59      0059 1 |
: 60      0060 1 | V03-003 MKP0003      Kathy Perko      14-Sept-1982
: 61      0061 1 | When the logging sink node id the executor, use a node
: 62      0062 1 | address of zero. This allows the logging database to be
: 63      0063 1 | transportable.
: 64      0064 1 |
: 65      0065 1 | V03-002 MKP0002      Kathy Perko      29-June-1982
: 66      0066 1 | Modify PURGE KNOWN processing to be able to purge multiple
: 67      0067 1 | entries with the same entity ID (for X25-PROTOCOL GROUPS).
: 68      0068 1 | Change qualifier handling to use entity qualifier Parameter
: 69      0069 1 | semantic Table (PST) address instead of Network Management
: 70      0070 1 | parameter code.
: 71      0071 1 |
: 72      0072 1 | V03-001 MKP0001      Kathy Perko      16-June-1982
: 73      0073 1 | Add entity qualifier handling.
: 74      0074 1 | --
: 75      0075 1 |

```

```

77 0076 1 %SBTTL 'Declarations'
78 0077 1
79 0078 1
80 0079 1 : TABLE OF CONTENTS:
81 0080 1
82 0081 1
83 0082 1 FORWARD ROUTINE
84 0083 1     NML$PURGEKNOWN      : NOVALUE,
85 0084 1     NML_PURKNOWN      : NOVALUE,
86 0085 1     NML$PURGE_KNOWN_NODES : NOVALUE,
87 0086 1     NML_PURGE_KNOWN_NODES : NOVALUE,
88 0087 1     NML$PURLOGGING     : NOVALUE,
89 0088 1     NML$PARENTITY      : NOVALUE,
90 0089 1     NML_PARENTITY      : NOVALUE,
91 0090 1     NML_PURLOGGING     : NOVALUE,
92 0091 1     NML_PURLOGALL;
93 0092 1
94 0093 1
95 0094 1 : INCLUDE FILES:
96 0095 1
97 0096 1
98 0097 1 LIBRARY 'LIBS:NMLLIB.L32';
99 0098 1 LIBRARY 'SHRLIBS:NMALIBRY.L32';
100 0099 1 LIBRARY 'SYSS$LIBRARY:STARLET.L32';
101 0100 1
102 0101 1
103 0102 1 : EQUATED SYMBOLS:
104 0103 1
105 0104 1
106 0105 1
107 0106 1 : Data base entity matching function codes.
108 0107 1
109 0108 1 LITERAL
110 0109 1     NML$C_ALWAYS = 0,           ! Always match
111 0110 1     NML$C_MATCH = 1,        ! Match if parameter is present
112 0111 1     NML$C_NOMATCH = 2;     ! Match if parameter is not present
113 0112 1
114 0113 1
115 0114 1 : OWN STORAGE:
116 0115 1
117 0116 1
118 0117 1
119 0118 1 : Event logging parameter buffer.
120 0119 1
121 0120 1 OWN
122 0121 1     NML$T_EVTBUFFER : VECTOR [NML$K_RECBFLEN, BYTE];
123 0122 1 BIND
124 0123 1     NML$Q_EVTBFDSC = UPLIT (NML$K_RECBFLEN, NML$T_EVTBUFFER) : VECTOR [2];
125 0124 1
126 0125 1 : Entity buffer and descriptor.
127 0126 1
128 0127 1 OWN
129 0128 1     NML$T_ENTBUFFER : BBLOCK [NML$K_ENTBUFLLEN],
130 0129 1     NML$Q_ENTBFDSC : VECTOR [2];
131 0130 1
132 0131 1
133 0132 1 : EXTERNAL REFERENCES:
```

NML\$PURGE
V04-000

NML PURGE permanent parameter module
Declarations

F 14
16-Sep-1984 00:27:19
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[NML.SRC]NMLPURGE.B32;1 (2)

NM
VC

```
134 0133 1 !
135 0134 1
136 0135 1 $NML_EXTDEF;
137 0136 1
138 0137 1 EXTERNAL LITERAL
139 0138 1     NML$_RECDELET;
140 0139 1
141 0140 1 EXTERNAL ROUTINE
142 0141 1     NML$SEARCHFLD,
143 0142 1     NML$ADDEVENTS,
144 0143 1     NML$ADDFILTERS,
145 0144 1     NML$BLD_REPLY,
146 0145 1     NML$DEL_FIELDS,
147 0146 1     NML$DELETRECORD,
148 0147 1     NML$GETNXTSNK,
149 0148 1     NML$GETRECORDNER,
150 0149 1     NML$MATCHRECORD,
151 0150 1     NML$READ_KNOWN_NODE_REC,
152 0151 1     NML$READRECORD,
153 0152 1     NML$REMSRC,
154 0153 1     NML$SAVEVENTS,
155 0154 1     NML$SEND,
156 0155 1     NML$WRITERECORD;
157 0156 1
```

```

: 159 0157 1 %SBTTL 'NMLSPURGEKNOWN Purge parameters for known entities'
: 160 0158 1 GLOBAL ROUTINE NMLSPURGEKNOWN (ENTITY, ENTITY_LEN, ENTITY_ADR) : NOVALUE =
: 161 0159 1
: 162 0160 1 !++
: 163 0161 1 ! FUNCTIONAL DESCRIPTION:
: 164 0162 1
: 165 0163 1 ! This routine removes the a set of parameters from the permanent
: 166 0164 1 ! data base entry for each entity of the specified type.
: 167 0165 1
: 168 0166 1 ! FORMAL PARAMETERS:
: 169 0167 1
: 170 0168 1 ! ENTITY Entity type code.
: 171 0169 1 ! ENTITY_LEN Entity ID length (used only for X25-PROTOCOL GROUPS).
: 172 0170 1 ! ENTITY_ADR Entity ID address (used only for X25-PROTOCOL GROUPS).
: 173 0171 1
: 174 0172 1 ! IMPLICIT INPUTS:
: 175 0173 1
: 176 0174 1 ! NML$GL_PRS_FLGS Message parsing flags.
: 177 0175 1
: 178 0176 1 ! SIDE EFFECTS:
: 179 0177 1
: 180 0178 1 ! Signals errors.
: 181 0179 1
: 182 0180 1 ! --
: 183 0181 1
: 184 0182 2 BEGIN
: 185 0183 2
: 186 0184 2 LOCAL
: 187 0185 2 deleteflg, ! Record delete flag
: 188 0186 2 fid, ! File id code
: 189 0187 2 matchflg, ! Record matching flag
: 190 0188 2 prm; ! Parameter code
: 191 0189 2
: 192 0190 2 fid = .nml$ab_entitydata [.entity, eit$b_fil id];
: 193 0191 2 matchflg = nm[$c_always];
: 194 0192 2
: 195 0193 2 ! If it's a PURGE ALL command, set a flag so the entity will be completely
: 196 0194 2 ! removed from the permanent database.
: 197 0195 2
: 198 0196 2 IF .nml$gl_prs_flg [nml$v_prs_all] THEN
: 199 0197 2 deleteflg = true
: 200 0198 2 ELSE
: 201 0199 2 deleteflg = false;
: 202 0200 2
: 203 0201 2 SELECTONEU .fid OF
: 204 0202 2 SET
: 205 0203 2 [nma$c_opn_log]:
: 206 0204 2 BEGIN
: 207 0205 2
: 208 0206 2 ! Purge parameters for all logging sink types.
: 209 0207 2
: 210 0208 2 nml$purlogging (.entity, nma$c_snk_con, 0); ! Console
: 211 0209 2 nml$purlogging (.entity, nma$c_snk_fil, 0); ! File
: 212 0210 2 nml$purlogging (.entity, nma$c_snk_mon, 0); ! Monitor
: 213 0211 2 END;
: 214 0212 2
: 215 0213 2 [nma$c_opn_x25,

```



```

.EXTRN NML$AB_ENTITYDATA
.EXTRN NML$AB_NML_NMV, NML$AB_PRMSEM
.EXTRN NML$AB_RECBUF, NML$AL_ENTINF TAB
.EXTRN NML$AL_PERMINF TAB
.EXTRN NML$AW_PRM DES, NML$GB_CMD_VER
.EXTRN NML$GB_ENTITY_CODE
.EXTRN NML$GB_ENTITY_FORMAT
.EXTRN NML$GL_QUALIFIER_PST
.EXTRN NML$GB_QUALIFIER_FORMAT
.EXTRN NML$GB_FUNCTION
.EXTRN NML$GB_INFO, NML$GB_OPTIONS
.EXTRN NML$GL_PRCODE, NML$GL_PRS_FLGS
.EXTRN NML$GL_NML_ENTITY
.EXTRN NML$GQ_NETRAMDSC
.EXTRN NML$GQ_RECBF DSC
.EXTRN NML$GW_PRMDESCNT
.EXTRN NML$ RECDELET, NML$SEARCHFLD
.EXTRN NML$ADDEVENTS, NML$ADDFILTERS
.EXTRN NML$BLD_REPLY, NML$DEL_FIELDS
.EXTRN NML$DELETRECORD
.EXTRN NML$GETNXTSNK, NML$GETREOWNER
.EXTRN NML$MATCHRECORD
.EXTRN NML$READ_KNOWN_NODE_REC
.EXTRN NML$READRECORD, NML$REMSRC
.EXTRN NML$SAVEEVENTS, NML$SEND
.EXTRN NML$WRITERECORD

```

.PSECT \$CODE\$,NOWRT,2

			00FC 00000	.ENTRY	NML\$PURGEKNOWN, Save R2,R3,R4,R5,R6,R7	: 0158
	57	00000000V	00 9E 00002	MOVAB	NML\$PURLOGGING, R7	
	54	04	AC D0 00009	MOVL	ENTITY, R4	: 0190
52	54		2C C5 0000D	MULL3	#44, R4, R2	
	55	00000000G00	42 9A 00011	MOVZBL	NML\$AB_ENTITYDATA[R2], FID	
			56 D4 00019	CLRL	MATCHFLG	: 0191
05	00000000G		00 01 E1 0001B	BBC	#1, NML\$GL_PRS_FLGS, 1\$: 0196
			53 01 D0 00023	MOVL	#1, DELETEFLG	: 0197
			02 02 11 00026	BRB	2\$	
			53 D4 00028	CLRL	DELETEFLG	: 0199
	02		55 D1 0002A	CPL	FID, #2	: 0203
			19 12 0002D	BNEQ	3\$	
	7E		01 7D 0002F	MOVQ	#1, -(SP)	: 0208
			54 DD 00032	PUSHL	R4	
	67		03 FB 00034	CALLS	#3, NML\$PURLOGGING	
	7E		02 7D 00037	MOVQ	#2, -(SP)	: 0209
			54 DD 0003A	PUSHL	R4	
	67		03 FB 0003C	CALLS	#3, NML\$PURLOGGING	
	7E		03 7D 0003F	MOVQ	#3, -(SP)	: 0210
			54 DD 00042	PUSHL	R4	
	67		03 FB 00044	CALLS	#3, NML\$PURLOGGING	
			04 00047	RET		: 0201
	05		55 D1 00048	CPL	FID, #5	: 0213
			1E 1F 0004B	BLSSU	4\$	
	06		55 D1 0004D	CPL	FID, #6	
			19 1A 00050	BGTRU	4\$	
	56		01 D0 00052	MOVL	#1, MATCHFLG	: 0216
		00000000G0042	9F 00055	PUSHAB	NML\$AB_ENTITYDATA+3[R2]	: 0217

NMLSPURGE
V04-000

NML PURGE permanent parameter module
NMLSPURGEKNOWN Purge parameters for known enti

J 14
16-Sep-1984 00:27:19
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[NML.SRC]NMLPURGE.B32;1

Page 8
(3)

50		9E 3C 0005C	MOVZWL	@(SP)+, PRM	
		50 DD 0005F	PUSHL	PRM	0219
		53 DD 00061	PUSHL	DELETEFLG	
		56 DD 00063	PUSHL	MATCHFLG	
7E	08	AC 7D 00065	MOVQ	ENTITY_LEN, -(SP)	0218
		08 11 00069	BRB	5\$	
		7E D4 0006B	4\$: CLRL	-(SP)	0224
		53 DD 0006D	PUSHL	DELETEFLG	
		56 DD 0006F	PUSHL	MATCHFLG	
		7E 7C 00071	CLRQ	-(SP)	
		54 DD 00073	5\$: PUSHL	R4	
		55 DD 00075	PUSHL	FID	
00000000V	00	07 FB 00077	CALLS	#7, NML_PURKNOWN	
		04 0007E	RET		0228

: Routine Size: 127 bytes, Routine Base: \$CODE\$ + 0000

```

232 0229 1 %SBITL 'NML_PURKNOWN Purge parameters for known entities'
233 0230 1 ROUTINE NML_PURKNOWN (FID, ENTITY, ENTITY_LEN, ENTITY_ADR,
234 0231 1 MAT, DEL, PRM) : NOVALUE =
235 0232 1
236 0233 1 +-
237 0234 1 FUNCTIONAL DESCRIPTION:
238 0235 1 This routine is called for PURGE KNOWN operations for all entities
239 0236 1 except nodes. It deletes permanent data base information for each
240 0237 1 permanent database record that it matches.
241 0238 1
242 0239 1 FORMAL PARAMETERS:
243 0240 1
244 0241 1 FID File id code.
245 0242 1 ENTITY Entity type code.
246 0243 1 ENTITY_LEN Entity ID length (used only for X25-PROTOCOL GROUPS).
247 0244 1 ENTITY_ADR Entity ID address (used only for X25-PROTOCOL GROUPS).
248 0245 1 MAT Qualifier flag (match, nomatch, always).
249 0246 1 DEL Delete flag (TRUE=yes, FALSE=no).
250 0247 1 PRM Parameter code (if applicable).
251 0248 1
252 0249 1 SIDE EFFECTS:
253 0250 1
254 0251 1 Signals errors.
255 0252 1
256 0253 1 --
257 0254 1
258 0255 2 BEGIN
259 0256 2
260 0257 2 LOCAL
261 0258 2 fldadr,
262 0259 2 fldsize,
263 0260 2 msgsize, ! Message size
264 0261 2 key, ! Temporary record key buffer
265 0262 2 recdsc : VECTOR [2], ! Record descriptor
266 0263 2 status;
267 0264 2
268 0265 2 ! Add parameters to every entity in the permanent database file.
269 0266 2
270 0267 2 key = 0;
271 0268 2 WHILE nml$readrecord (.fid, key, 0, nml$gq_recbfdsc, recdsc, 0) DO
272 0269 2 BEGIN
273 0270 2 status = true;
274 0271 2 fldadr = 0;
275 0272 2 fldsize = 0;
276 0273 2
277 0274 3 IF .mat EQL nml$c_match THEN
278 0275 4 BEGIN
279 0276 4 IF NOT nma$searchfld (recdsc, .prm, fldsize, fldadr) THEN
280 0277 4 status = false
281 0278 4 ELSE
282 0279 4
283 0280 4 ! If the entity takes a qualifier (e.g. X25-PROTOCOL GROUPS),
284 0281 4 ! it has more than one entry in the database). Use the
285 0282 4 ! supplied entity ID to identify all of the entity's entries.
286 0283 4
287 0284 5 BEGIN
288 0285 5 IF .entity_len GTR 0 THEN

```


			18	AE	9F	00060	PUSHAB	RECDSC		
			18	AC	DD	00063	PUSHL	DEL	:	
			08	AC	DD	00066	PUSHL	ENTITY	:	
00000000V	00			05	FB	00069	CALLS	#5, NML_PURENTITY	:	
				0C	AE	9F	PUSHAB	MSGSIZE	:	0300
		00000000G		00	9F	00073	PUSHAB	NML\$AB MSGBLOCK	:	
00000000G	00			02	FB	00079	CALLS	#2, NML\$BLD_REPLY	:	
				0C	AE	DD	PUSHL	MSGSIZE	:	0301
		00000000G		00	9F	00083	PUSHAB	NML\$AB SNDBUFFER	:	
00000000G	00			02	FB	00089	CALLS	#2, NML\$SEND	:	
				08	AE	D6	INCL	KEY	:	0304
				FF72	31	00093	BRW	1\$:	0268
					04	00096	RET		:	0306

; Routine Size: 151 bytes, Routine Base: \$CODE\$ + 007F

```

311 0307 1 %SBTTL 'NML$PURGE KNOWN_NODES Purge known nodes'
312 0308 1 GLOBAL ROUTINE NML$PURGE_KNOWN_NODES (ENTITY, INF, DUM1, DUM2) : NOVALUE =
313 0309 1
314 0310 1 +-
315 0311 1 FUNCTIONAL DESCRIPTION:
316 0312 1 This routine purges all entries for nodes that are in the permanent
317 0313 1 data base. Remote nodes are then purged. Loop nodes are purged
318 0314 1 last. The executor node is purged last.
319 0315 1
320 0316 1 FORMAL PARAMETERS:
321 0317 1 ENTITY Entity type code.
322 0318 1 INF Information type code.
323 0319 1 DUM1 Not used.
324 0320 1 DUM2 Not used.
325 0321 1
326 0322 1 --
327 0323 1
328 0324 2 BEGIN
329 0325 2 IF NOT .nml$gl_prs_flg [nml$v_prs_loopg] THEN
330 0326 2
331 0327 2 Purge remote nodes.
332 0328 2
333 0329 2 nml_purge_known_nodes (nml$c_node);
334 0330 2
335 0331 2 Return loop nodes.
336 0332 2
337 0333 2 IF .nml$gl_prs_flg [nml$v_prs_loopg] OR
338 0334 2 .nml$gl_prs_flg [nml$v_prs_all] THEN
339 0335 2 nml_purge_known_nodes (nml$c_loopnode);
340 0336 2 IF NOT .nml$gl_prs_flg [nml$v_prs_loopg] THEN
341 0337 2
342 0338 2 Return executor node.
343 0339 2
344 0340 2 nml$parentity (nml$c_executor, 0, 0, 0, 0, 0);
345 0341 2
346 0342 1 END; ! End of NML$PURGE_KNOWN_NODES

```

			000C 00000	.ENTRY	NML\$PURGE_KNOWN_NODES, Save R2,R3	: 0308
		53 00000000V	00 9E 00002	MOVAB	NML PURGE_KNOWN_NODES, R3	
		52 00000000G	00 9E 00009	MOVAB	NML\$GL_PR5_FLGS, R2	
0E	01	A2	03 E0 00010	BBS	#3, NML\$GL_PR5_FLGS+1, 1\$: 0325
			03 DD 00015	PUSHL	#3	: 0329
		63	01 FB 00017	CALLS	#1, NML PURGE KNOWN_NODES	
04	01	A2	03 E0 0001A	BBS	#3, NML\$GL_PR5_FLGS+1, 1\$: 0333
05		62	01 E1 0001F	BBC	#1, NML\$GL_PR5_FLGS, 2\$: 0334
			05 DD 00023 1\$:	PUSHL	#5	: 0335
		63	01 FB 00025	CALLS	#1, NML PURGE KNOWN_NODES	
0E	01	A2	03 E0 00028 2\$:	BBS	#3, NML\$GL_PR5_FLGS+1, 3\$: 0336
			7E 7C 0002D	CLRQ	-(SP)	: 0340
			7E 7C 0002F	CLRQ	-(SP)	
		7E	07 7D 00031	MOVQ	#7, -(SP)	
		00000000V 00	06 FB 00034	CALLS	#6, NML\$PARENTITY	
			04 0003B 3\$:	RET		: 0342

NML\$PURGE
V04-000

NML PURGE permanent parameter module
NML\$PURGE_KNOWN_NODES Purge known nodes

B 15
16-Sep-1984 00:27:19
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[NML.SRC]NMLPURGE.B32;1 Page 13 (5)

. Routine Size: 60 bytes, Routine Base: \$CODES + 0116

NML
VO

.....

.

```

348 0343 1 %SBTTL 'NML_PURGE_KNOWN_NODES Purge parameters for known nodes'
349 0344 1 ROUTINE NML_PURGE_KNOWN_NODES (ENTITY) : NOVALUE =
350 0345 1
351 0346 1  +-+
352 0347 1  FUNCTIONAL DESCRIPTION:
353 0348 1  This routine deletes parameters from permanent data base information
354 0349 1  for known nodes or loopnodes.
355 0350 1
356 0351 1  FORMAL PARAMETERS:
357 0352 1  ENTITY          Entity type code.
358 0353 1
359 0354 1  SIDE EFFECTS:
360 0355 1  Signals errors.
361 0356 1
362 0357 1  --
363 0358 1
364 0359 2 BEGIN
365 0360 2
366 0361 2 LOCAL
367 0362 2     deleteflg,
368 0363 2     recdsc:   VECTOR [2],
369 0364 2     rewind_flag,
370 0365 2     msgsize;
371 0366 2
372 0367 2 :
373 0368 2 : If processing a PURGE ALL command, set up to delete the entire record from
374 0369 2 : the permanent database. Since the only valid parameter for loopnodes is
375 0370 2 : CIRCUIT, purging CIRCUIT is the same as PURGE ALL for loopnodes.
376 0371 2 :
377 0372 2 IF .nml$gl_prs_flg [nml$pr_all] OR
378 0373 2 .nml$gl_prs_flg [nml$pr_loopg] THEN
379 0374 2     deleteflg = true
380 0375 2 ELSE
381 0376 2     deleteflg = false;
382 0377 2 :
383 0378 2 : Add parameters to every record in the file.
384 0379 2 :
385 0380 2 rewind_flag = true;
386 0381 2 WHILE nml$read_known_node_rec (.entity,
387 0382 2                               nml$gg_recbfdsc,
388 0383 2                               recdsc,
389 0384 2                               .rewind_flag) DO
390 0385 2     BEGIN
391 0386 2     rewind_flag = false;
392 0387 2     nml_purentity (.entity, .deleteflg, recdsc, UPLIT (0), 0);
393 0388 2     :
394 0389 2     : Build and send the response message.
395 0390 2     :
396 0391 2     nml$bld_reply (nml$ab_msgblock, msgsize);
397 0392 2     nml$send (nml$ab_sndbuffer, .msgsize);
398 0393 2     END;
399 0394 1 END;          ! End of NML_PURGE_KNOWN_NODES

```

.PSECT \$PLITS,NOWRT,NOEXE,2

00000000 00008 P.AAB: .LONG 0

.PSECT \$CODE\$,NOWRT,2

		000C 00000	NML_PURGE_KNOWN_NODES:		
			.WORD	Save R2,R3	: 0344
			SUBL2	#12, SP	
08 00000000G	00	0C C2 00002	BBS	#1, NML\$GL_PRS_FLGS, 1\$: 0372
05 00000000G	00	01 E0 00005	BBC	#3, NML\$GL_PRS_FLGS+1, 2\$: 0373
	53	03 E1 0000D	MOVL	#1, DELETEFLG	: 0374
		01 D0 00015 1\$:	BRB	3\$	
		02 11 00018	CLRL	DELETEFLG	: 0376
	52	53 D4 0001A 2\$:	MOVL	#1, REWIND_FLAG	: 0380
		01 D0 0001C 3\$:	PUSHL	REWIND_FLAG	: 0384
		52 DD 0001F 4\$:	PUSHAB	RECDSC	: 0381
		08 AE 9F 00021	PUSHAB	NML\$GQ_RECBFDSC	
	00000000G	00 9F 00024	PUSHL	ENTITY	
	04	AC DD 0002A	CALLS	#4, NML\$READ_KNOWN_NODE_REC	
00000000G	00	04 FB 0002D	BLBC	R0, 5\$	
	39	50 E9 00034	CLRL	REWIND_FLAG	: 0386
		52 D4 00037	CLRL	-(SP)	: 0387
		7E D4 00039	PUSHAB	P.AAB	
	00000000'	00 9F 0003B	PUSHAB	RECDSC	
	0C	AE 9F 00041	PUSHL	DELETEFLG	
		53 DD 00044	PUSHL	ENTITY	
00000000V	00	04 AC DD 00046	CALLS	#5, NML_PURENTITY	
		05 FB 00049	PUSHL	SP	: 0391
		5E DD 00050	PUSHAB	NML\$AB_MSGBLOCK	
00000000G	00	00 9F 00052	CALLS	#2, NML\$BLD_REPLY	: 0392
		02 FB 00058	PUSHL	MSGSIZE	: 0392
		6E DD 0005F	PUSHAB	NML\$AB_SNDBUFFER	
00000000G	00	00 9F 00061	CALLS	#2, NML\$SEND	
		02 FB 00067	BRB	4\$: 0381
		AF 11 0006E	RET		: 0394
		04 00070 5\$:			

: Routine Size: 113 bytes, Routine Base: \$CODE\$ + 0152

```

401 0395 1 %SBTTL 'NML$PURLOGGING Purge logging parameters'
402 0396 1 GLOBAL ROUTINE NML$PURLOGGING (ENTITY, SNK, DUM2) : NOVALUE =
403 0397 1
404 0398 1 !++
405 0399 1 ! FUNCTIONAL DESCRIPTION:
406 0400 1 !
407 0401 1 !     Add parameters to the permanent data base entry for the specified
408 0402 1 !     logging entity.
409 0403 1 !
410 0404 1 ! FORMAL PARAMETERS:
411 0405 1 !
412 0406 1 !     ENTITY      Entity type code.
413 0407 1 !     SNK         Logging sink type.
414 0408 1 !     DUM2       Not used.
415 0409 1 !
416 0410 1 ! IMPLICIT INPUTS:
417 0411 1 !
418 0412 1 !     NML$GL_PRS_FLGS Message parsing flags.
419 0413 1 !     NML$GW_EVT$SNKADR Sink node address.
420 0414 1 !
421 0415 1 ! --
422 0416 1
423 0417 2 BEGIN
424 0418 2
425 0419 2 LOCAL
426 0420 2     FID,           ! File id code
427 0421 2     MSGFLG,      ! Response message flag
428 0422 2     MSGSIZE,     ! Message size
429 0423 2     KEY,        ! Temporary record key buffer
430 0424 2     OWNER,     ! Search key
431 0425 2     RECDSC  : VECTOR [2]; ! Record descriptor
432 0426 2
433 0427 2 !
434 0428 2 ! If this is the event sink (ESI) parameter group then call the normal
435 0429 2 ! routine.
436 0430 2
437 0431 2 IF .NML$GL_PRS_FLGS [NML$V_PRS_ESIPG] THEN
438 0432 2 BEGIN
439 0433 2     NML$PURI:ENTITY (NML$C_SINK, 1, SNK, 0, 0, 0);
440 0434 2     RETURN
441 0435 2 END;
442 0436 2
443 0437 2 !
444 0438 2 ! The event filter parameter group is handled specially.
445 0439 2
446 0440 2 NML$AB_MSGBLOCK [MSB$L_FLAGS] = 0; ! Initialize message flags
447 0441 2 FID = .NML$AB_ENTITYDATA [.ENTITY, EIT$B_FILEID]; ! Get file id
448 0442 2 OWNER = .NML$AB_ENTITYDATA [.ENTITY, EIT$W_KEY]; ! Get search key
449 0443 2 KEY = 0; ! Initialize record key
450 0444 2
451 0445 2 ! If this is all (no parameters) then purge all logging to the specified
452 0446 2 ! sink type (console, monitor, or file).
453 0447 2
454 0448 2 IF .NML$GL_PRS_FLGS [NML$V_PRS_ALL] THEN
455 0449 2 BEGIN
456 0450 2     MSGFLG = NML_PURLOGALL (.SNK);
457 0451 2 END

```

```

458 0452 2 ELSE
459 0453 2
460 0454 2 Purge the event filter (EFI) database.
461 0455 2
462 0456 2 BEGIN
463 0457 2
464 0458 2 If purging logging for the executor node, use a sink node address of
465 0459 2 zero. Using zero for the executor node allows the logging database
466 0460 2 to be transported to another node and have the events logged at the
467 0461 2 new executor.
468 0462 2
469 0463 2 IF .NML$GL_PRS FLGS [NML$V_PRS_EXESNK] THEN
470 0464 2 NML$GW_EVTSNKADR = 0;
471 0465 2
472 0466 2 If there is a record in the logging database for the sink node specified
473 0467 2 in the NICE message, purge the logging filters.
474 0468 2
475 0469 2 IF NML$MATCHRECORD (.FID,
476 0470 2 NML$GQ_RECBFDSC,
477 0471 2 KEY,
478 0472 2 .OWNER, 2, NML$GW_EVTSNKADR,
479 0473 2 0, 0, 0, ! No qualifier
480 0474 2 RECD$C) THEN
481 0475 2 BEGIN
482 0476 2 NML PURLOGGING (.SNK, .NML$GW_EVTSNKADR, RECD$C, KEY);
483 0477 2 MSGFLG = TRUE;
484 0478 2 END;
485 0479 2 END;
486 0480 2
487 0481 2 Set up to add entity id (sink type code) to NICE response message.
488 0482 2
489 0483 2 NML$Q_ENTBFDSC [0] = 1;
490 0484 2 NML$Q_ENTBFDSC [1] = NML$T_ENTBUFFER;
491 0485 2 NML$T_ENTBUFFER<0,8> = .SNR;
492 0486 2 NML$AB_MSGBLOCK [MSB$V_ENTD_FLD] = 1; ! Set entity descriptor flag
493 0487 2 NML$AB_MSGBLOCK [MSB$A_ENTITY] = NML$Q_ENTBFDSC; ! Add entity descriptor pointer
494 0488 2
495 0489 2 Build and send the NICE response message.
496 0490 2
497 0491 2 IF .MSGFLG THEN
498 0492 2 BEGIN
499 0493 2 NML$BLD_REPLY (NML$AB_MSGBLOCK, MSGSIZE);
500 0494 2 NML$SEND (NML$AB_SNDBUFFER, .MSGSIZE);
501 0495 2 END;
502 0496 2
503 0497 1 END; ! End of NML$PURLOGGING

```

		01FC 00000	.ENTRY	NML\$PURLOGGING, Save R2,R3,R4,R5,R6,R7,R8	: 0396
58	00000000G	00 9E 00002	MOVAB	NML\$GL_PRS FLGS, R8	:
57	00000000G	00 9E 00009	MOVAB	NML\$GW_EVTSNKADR, R7	:
56	00000000G	00 9E 00010	MOVAB	NML\$AB_MSGBLOCK, R6	:
55	00000000'	00 9E 00017	MOVAB	NML\$Q_ENTBFDSC, R5	:
5E		10 C2 0001E	SUBL2	#16, 5P	:

13	01	A8	04	E1	00021	BBC	#4, NML\$GL_PRS_FLGS+1, 1\$	0431
			7E	7C	00026	CLRQ	-(SP)	0433
			08	D4	00028	CLRL	-(SP)	
				AC	9F 0002A	PUSHAB	SNK	
				01	DD 0002D	PUSHL	#1	
				02	DD 0002F	PUSHL	#2	
	00000000V	00		06	FB 00031	CALLS	#6, NML\$PURITY	
				04	00038	RET		0432
				66	D4 00039	CLRL	NML\$AB MSGBLOCK	0440
50	04	AC	2C	C5	0003B	MULL3	#4, ENTITY RO	0441
		53	00000000G	0040	9A 00040	MOVZBL	NML\$AB_ENTITYDATA[RO], FID	
			00000000G	0040	9F 00048	PUSHAB	NML\$AB_ENTITYDATA+3[RO]	0442
		52		9E	3C 0C04F	MOVZWL	@(SP)+, OWNER	
				6E	D4 00052	CLRL	KEY	0443
OF		68		01	E1 00054	BBC	#1, NML\$GL_PRS_FLGS, 2\$	0448
			08	AC	DD 00058	PUSHL	SNK	0450
	00000000V	00		01	FB 0005B	CALLS	#1, NML PURLOGALL	
		54		50	D0 00062	MOVL	RO, MSGFLG	
				3D	11 00065	BRB	4\$	0448
		02		01	A8 E9 00067	BLBC	NML\$GL_PRS_FLGS+1, 3\$	0463
				67	B4 0006B	CLRW	NML\$GW_EVT\$NKADR	0464
			08	AE	9F 0C06D	PUSHAB	RECDSC	0469
				7E	7C 00070	CLRQ	-(SP)	
				7E	D4 00072	CLRL	-(SP)	
				57	DD 00074	PUSHL	R7	
				02	DD 00076	PUSHL	#2	
				52	DD 00078	PUSHL	OWNER	0472
			1C	AE	9F 0007A	PUSHAB	KEY	0469
			00000000G	00	9F 0007D	PUSHAB	NML\$GQ_REC\$FDSC	
				53	DD 00083	PUSHL	FID	
	00000000G	00		0A	FB 00085	CALLS	#10, NML\$MATCHRECORD	
		15		50	E9 0008C	BLBC	RO, 4\$	
				5E	DD 0008F	PUSHL	SP	0476
			0C	AE	9F 00091	PUSHAB	RECDSC	
		7E		67	3C 00094	MOVZWL	NML\$GW_EVT\$NKADR, -(SP)	
			08	AC	DD 00097	PUSHL	SNK	
	00000000V	00		04	FB 0009A	CALLS	#4, NML PURLOGGING	
		54		01	D0 000A1	MOVL	#1, MSGFLG	0477
		5		01	D0 000A4	MOVL	#1, NML\$Q_ENT\$FDSC	0483
	04	A5	C0	A5	9E 000A7	MOVAB	NML\$T_ENT\$BUFFER, NML\$Q_ENT\$FDSC+4	0484
	C0	A5	08	AC	90 000AC	MOVAB	SNK, NML\$T_ENT\$BUFFER	0485
		66		10	88 000B1	BISB2	#16, NML\$AB MSGBLOCK	0486
	14	A6		65	9E 000B4	MOVAB	NML\$Q_ENT\$FDSC, NML\$AB_MSGBLOCK+20	0487
		1C		54	E9 000B8	BLBC	MSGFLG, 5\$	0491
			04	AE	9F 000BB	PUSHAB	MSG\$SIZE	0493
				56	DD 000BE	PUSHL	R6	
	00000000G	00		02	FB 000C0	CALLS	#2, NML\$BLD_REPLY	
			04	AE	DD 000C7	PUSHL	MSG\$SIZE	0494
			00000000G	00	9F 000CA	PUSHAB	NML\$AB_SND\$BUFFER	
	00000000G	00		02	FB 000D0	CALLS	#2, NML\$SEND	
				04	000D7	RET		0497

; Routine Size: 216 bytes, Routine Base: \$CODE\$ + 01C3

```

: 505 0498 1 %SBTTL 'NML$PURITY Delete entity parameters'
: 506 0499 1 GLOBAL ROUTINE NML$PURITY (ENTITY, ENTITY_LEN, ENTITY_ADR,
: 507 0500 1 QUAL_PST, QUAL_LEN, QUAL_ADR) : NOVALUE =
: 508 0501 1
: 509 0502 1 |++
: 510 0503 1 | FUNCTIONAL DESCRIPTION:
: 511 0504 1 |
: 512 0505 1 | This routine removes a set of parameters from the permanent data
: 513 0506 1 | base entry for the specified entity.
: 514 0507 1 |
: 515 0508 1 | FORMAL PARAMETERS:
: 516 0509 1 |
: 517 0510 1 | ENTITY Entity type code.
: 518 0511 1 | ENTITY_LEN Byte count of entity id string.
: 519 0512 1 | ENTITY_ADR Address of entity id string.
: 520 0513 1 | QUAL_PST Address of Qualifier's Parameter Semantic Table (PST)
: 521 0514 1 | address
: 522 0515 1 | QUAL_LEN Byte count of qualifier id string
: 523 0516 1 | QUAL_ADR Address of qualifier id string
: 524 0517 1 |
: 525 0518 1 | IMPLICIT INPUTS:
: 526 0519 1 |
: 527 0520 1 | NML$GL_PRS_FLGS Message parsing flags.
: 528 0521 1 |
: 529 0522 1 | SIDE EFFECTS:
: 530 0523 1 |
: 531 0524 1 | Signals errors.
: 532 0525 1 |
: 533 0526 1 | --
: 534 0527 1 |
: 535 0528 2 BEGIN
: 536 0529 2
: 537 0530 2 MAP
: 538 0531 2 nml$gb_entity_format : BYTE SIGNED;
: 539 0532 2
: 540 0533 2 LOCAL
: 541 0534 2 deleteflg,
: 542 0535 2 fid, | File id code
: 543 0536 2 msgsize, | Message size
: 544 0537 2 key, | Temporary record key buffer
: 545 0538 2 key_value_dsc: VECTOR [2], | ID of node being purged (not used
: 546 0539 2 | for other entities.
: 547 0540 2 owner, | Search key
: 548 0541 2 recdsc : VECTOR [2], | Record descriptor
: 549 0542 2 temp,
: 550 0543 2 status;
: 551 0544 2
: 552 0545 2 | Set the flag to indicate record deletion (or not).
: 553 0546 2
: 554 0547 2 IF .nml$gl_prs_flg [nml$v_prs_all] OR
: 555 0548 2 .nml$gl_prs_flg [nml$v_prs_loopg] THEN
: 556 0549 2 deleteflg = true
: 557 0550 2 ELSE
: 558 0551 2 deleteflg = false;
: 559 0552 2
: 560 0553 2 | Add parameters to the entity record in the file.
: 561 0554 2

```

```
562 0555 2 fid = .nml$ab_entitydata [.entity, eit$b_fileid]; ! Get file id
563 0556 2 owner = .nml$ab_entitydata [.entity, eit$b_key]; ! Get search key
564 0557 2 key_value_dsc [0] = .entity_len;
565 0558 2 key_value_dsc [1] = .entity_adr;
566 0559 2
567 0560 2 : Find the record in the file.
568 0561 2
569 0562 2 IF .fid NEQ nma$sc_opn_node THEN
570 0563 2 BEGIN
571 0564 2 key = 0;
572 0565 2 status = nml$matchrecord (.fid,
573 0566 2 nml$gq_recbfdsc,
574 0567 2 key,
575 0568 2 .owner, .entity_len, .entity_adr,
576 0569 2 .qual_pst, .qual_len, .qual_adr,
577 0570 2 recdsc);
578 0571 2 END
579 0572 2 ELSE
580 0573 2
581 0574 2 : Since the node permanent database is much larger than the others, it
582 0575 2 : has a different structure using four ISAM keys instead of one.
583 0576 2
584 0577 2 BEGIN
585 0578 2 key = .owner;
586 0579 2 IF .owner EQL nmn$sc_typ_key_ref THEN
587 0580 2 BEGIN
588 0581 2 key_value_dsc [0] = nmn$sc_typ_key_len;
589 0582 2 key_value_dsc [1] = entity;
590 0583 2 END;
591 0584 2 status = nml$readrecord (.fid, owner, key_value_dsc,
592 0585 2 nml$gq_recbfdsc, recdsc, temp);
593 0586 2 END;
594 0587 2
595 0588 2 IF .status THEN
596 0589 2 nml_purentity (.entity, .deleteflg, recdsc, key, key_value_dsc)
597 0590 2 ELSE
598 0591 2 BEGIN
599 0592 2
600 0593 2 : No such entity found in data base.
601 0594 2 : If processing a PURGE KNOWN NODES ALL, and there is no record for
602 0595 2 : the executor, don't return an error message to NCP. Otherwise,
603 0596 2 : return an error response.
604 0597 2
605 0598 2 IF .entity EQL nml$sc_executor AND
606 0599 2 .status EQL rms$rnf AND
607 0600 2 .nml$gb_entity_format EQL nma$sc_ent_kno THEN
608 0601 2 RETURN;
609 0602 2 nml$ab_msgblock [msb$l_flags] = msb$m_det_fld; ! Message detail only
610 0603 2 nml$ab_msgblock [msb$b_code] = nma$sc_sts_cmp; ! Add error code
611 0604 2 nml$ab_msgblock [msb$b_detail] =
612 0605 2 .nml$ab_entitydata [.entity, eit$b_detail];
613 0606 2 END;
614 0607 2
615 0608 2 : Build and send the response message.
616 0609 2
617 0610 2 nml$bld_reply (nml$ab_msgblock, msgsize);
618 0611 2 nml$send (nml$ab_sndbuffer, .msgsize);
```

: 619
: 620

0612 2
0613 1 END:

! End of NML\$PURITY

				00FC 00000	.ENTRY	NML\$PURITY, Save R2,R3,R4,R5,R6,R7	: 0499
		57	00000000G	00 9E 00002	MOVAB	NML\$GQ_RECBDSC, R7	
		56	0000000JG	00 9E 00009	MOVAB	NML\$AB_ENTITYDATA, R6	
		55	0000000JG	00 9E 00010	MOVAB	NML\$AB_MSGBLOCK, R5	
		5E		20 C2 00017	SUBL2	#32, SP	
08	00000000G	00		01 E0 0001A	BBS	#1, NML\$GL_PRS_FLGS, 1\$: 0547
05	00000000G	00		03 E1 00022	BBC	#3, NML\$GL_PRS_FLGS+1, 2\$: 0548
		54		01 D0 0002A	MOVL	#1, DELETEFLG	: 0549
				02 11 0002D	BRB	3\$	
				54 D4 0002F	CLRL	DELETEFLG	: 0551
52	04	AC		2C C5 00031	MULL3	#44, ENTITY, R2	: 0555
		53		6642 9A 00036	MOVZBL	NML\$AB_ENTITYDATA[R2], FID	
				03 A642 9F 0003A	PUSHAB	NML\$AB_ENTITYDATA+3[R2]	: 0556
	04	AE		9E 3C 0003E	MOVZWL	@(SP)+, OWNER	
	18	AE		08 AC 7D 00042	MOVQ	ENTITY_LEN, KEY_VALUE_DSC	: 0557
				53 D5 00047	TSTL	FID	: 0562
				24 13 00049	BEQL	4\$	
				08 AE D4 0004B	CLRL	KEY	: 0564
				10 AE 9F 0004E	PUSHAB	RECDSC	: 0565
		7E		14 AC 7D 00051	MOVQ	QUAL_LEN, -(SP)	: 0569
		7E		0C AC 7D 00055	MOVQ	ENTITY_ADR, -(SP)	: 0568
				08 AC DD 00059	PUSHL	ENTITY_LEN	
				1C AE DD 0005C	PUSHL	OWNER	
				24 AE 9F 0005F	PUSHAB	KEY	: 0565
			0088	8F BB 00062	PUSHR	#^M<R3,R7>	
00000000G	00			0A FB 00066	CALLS	#10, NML\$MATCHRECORD	
				2A 11 0006D	BRB	6\$: 0562
	08	AE		04 AE D0 0006F	MOVL	OWNER, KEY	: 0578
		01		04 AE D1 00074	CMPL	OWNER, #1	: 0579
				09 12 00078	BNEQ	5\$	
	18	AE		02 D0 0007A	MOVL	#2, KEY_VALUE_DSC	: 0581
	1C	AE		04 AC 9E 0007E	MOVAB	ENTITY, -KEY_VALUE_DSC+4	: 0582
				5E DD 00083	PUSHL	SP	: 0584
				14 AE 9F 00085	PUSHAB	RECDSC	
				57 DD 00088	PUSHL	R7	
				24 AE 9F 0008A	PUSHAB	KEY_VALUE_DSC	
				14 AE 9F 0008D	PUSHAB	OWNER	
				53 DD 00090	PUSHL	FID	
00000000G	00			06 FB 00092	CALLS	#6, NML\$READRECORD	
	17			50 E9 00099	BLBC	STATUS, 7\$: 0588
				18 AE 9F 0009C	PUSHAB	KEY_VALUE_DSC	: 0589
				0C AE 9F 0009F	PUSHAB	KEY	
				18 AE 9F 000A2	PUSHAB	RECDSC	
				54 DD 000A5	PUSHL	DELETEFLG	
				04 AC DD 000A7	PUSHL	ENTITY	
00000000V	00			05 FB 000AA	CALLS	#5, NML_PURITY	
				28 11 000B1	BRB	9\$	
	07			04 AC D1 000B3	CMPL	ENTITY, #7	: 0598
				13 12 000B7	BNEQ	8\$	
000182B2	8F			50 D1 000B9	CMPL	STATUS, #98994	: 0599

NML\$PURGE
V04-000

NML PURGE permanent parameter module
NML\$PURITY Delete entity parameters

K 15
16-Sep-1984 00:27:19
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[NML.SRC]NMLPURGE.B32;1
Page 22
(8)

FF	8F	00000000G	0A	12	000C0	BNEQ	8\$:	0600	
			00	91	000C2	CMPB	NML\$GB_ENTITY_FORMAT, #-1	:		
			2B	13	000CA	BEQL	10\$:		
	65		02	D0	000CC	8\$:	MOVL	#2, NML\$AB_MSGBLOCK	0602	
04	A5		08	8E	000CF	MNEGB	#8, NML\$AB_MSGBLOCK+4	:	0603	
		01	A642	9F	000D3	PUSHAB	NML\$AB_ENTITYDATA+1[R2]	:	0605	
08	A5		9E	B0	000D7	MOVW	@(SP)+, NML\$AB_MSGBLOCK+8	:		
			0C	AE	9F	000DB	9\$:	PUSHAB	MSGSIZE	0610
			55	DD	000DE	PUSHL	R5	:		
00000000G	00		02	FB	000E0	CALLS	#2, NML\$BLD_REPLY	:		
			0C	AE	DD	000E7	PUSHL	MSGSIZE	0611	
		00000000G	00	9F	000EA	PUSHAB	NML\$AB_SNDBUFFER	:		
00000000G	00		02	FB	000F0	CALLS	#2, NML\$SEND	:		
			04	000F7	10\$:	RET		:	0613	

; Routine Size: 248 bytes, Routine Base: \$CODE\$ + 029B


```

622 0614 1 %SBTTL 'NML_PURENTITY Purge entity parameters'
623 0615 1 ROUTINE NML_PURENTITY (ENTITY, DEL, RECDSC, KEY, KEY_VALUE_DSC) : NOVALUE =
624 0616 1
625 0617 1 |++
626 0618 1 | FUNCTIONAL DESCRIPTION:
627 0619 1 |
628 0620 1 |     This routine removes permanent data base parameters for the specified
629 0621 1 |     line.
630 0622 1 |
631 0623 1 | FORMAL PARAMETERS:
632 0624 1 |
633 0625 1 |     ENTITY      Entity type code.
634 0626 1 |     DEL         Record delete flag (TRUE=yes, FALSE=no).
635 0627 1 |     RECDSC     Address of current record descriptor.
636 0628 1 |     KEY        For all but node database - Current record key.
637 0629 1 |     KEY_VALUE_DSC Used only for node database - ID of node being purged
638 0630 1 |
639 0631 1 | --
640 0632 2 BEGIN
641 0633 2
642 0634 2 MAP
643 0635 2     recdsc : REF VECTOR [2];      ! Record descriptor
644 0636 2
645 0637 2 LOCAL
646 0638 2     fid;                          ! File id code
647 0639 2
648 0640 2     fid = .nml$ab_entitydata [.entity, eit$b_fileid]; ! Get file id
649 0641 2     nml$ab_msgblock [msb$_flags] = 0;
650 0642 2     nml$ab_msgblock [msb$_code] = nml$_sts_suc;
651 0643 2
652 0644 2     Delete parameters.  If none specified then delete record.
653 0645 2
654 0646 2 IF NOT .del THEN
655 0647 2     BEGIN
656 0648 2     |
657 0649 2     |     Purge the parameters specified by the NICE message, and then write the
658 0650 2     |     record back to the permanent database file.
659 0651 2     |
660 0652 2     |     nml$del_fields (.recdsc);
661 0653 2     |     nml$writerecord (.fid, .entity, .key, .recdsc, nml$_update_rec);
662 0654 2     |     END
663 0655 2 ELSE
664 0656 2     BEGIN
665 0657 2     |
666 0658 2     |     Delete the entity's record from the permanent database.  This is
667 0659 2     |     generally a PURGE ALL operation.
668 0660 2     |
669 0661 2     |     nml$deleterecord (.fid, .key, .key_value_dsc);
670 0662 2     |     nml$ab_msgblock [msb$_flags] = msb$_m_msg_fld;
671 0663 2     |     nml$ab_msgblock [msb$_text] = nml$_recdelete;
672 0664 2     |     END;
673 0665 2     |
674 0666 2     |     Add entity id to message descriptor.
675 0667 2     |
676 0668 2     |     nml$_q_entbfdsc [0] = nml$_k_entbuflen;
677 0669 2     |     nml$_q_entbfdsc [1] = nml$_t_entbuffer;
678 0670 2

```



```

: 689 0680 1 %SBTTL 'NML_PURLOGGING Purge logging parameters'
: 690 0681 1 ROUTINE NML_PURLOGGING (SNK, SNKADR, RECDSC, KEY) : NOVALUE =
: 691 0682 1
: 692 0683 1
: 693 0684 1 :++
: 694 0685 1 :FUNCTIONAL DESCRIPTION:
: 695 0686 1 :       This routine performs common purge functions for both singular
: 696 0687 1 :       and plural requests.
: 697 0688 1
: 698 0689 1 :FORMAL PARAMETERS:
: 699 0690 1
: 700 0691 1 :       SNK           Logging sink type.
: 701 0692 1 :       SNKADR        Sink node address.
: 702 0693 1 :       RECDSC        Address of current record descriptor.
: 703 0694 1 :       KEY           Address of current record key.
: 704 0695 1
: 705 0696 1 :IMPLICIT INPUTS:
: 706 0697 1
: 707 0698 1 :       NONE
: 708 0699 1
: 709 0700 1 :IMPLICIT OUTPUTS:
: 710 0701 1
: 711 0702 1 :       NONE
: 712 0703 1
: 713 0704 1 :ROUTINE VALUE:
: 714 0705 1 :COMPLETION CODES:
: 715 0706 1
: 716 0707 1 :       NONE
: 717 0708 1
: 718 0709 1 :SIDE EFFECTS:
: 719 0710 1
: 720 0711 1 :       NONE
: 721 0712 1
: 722 0713 1 :--
: 723 0714 1
: 724 0715 2 :BEGIN
: 725 0716 2
: 726 0717 2 :MAP
: 727 0718 2 :       SNKADR : WORD,
: 728 0719 2 :       RECDSC : REF VECTOR;
: 729 0720 2
: 730 0721 2 :LOCAL
: 731 0722 2 :       FID,           ! File id code
: 732 0723 2 :       FLD$SIZE,
: 733 0724 2 :       FLDADR,
: 734 0725 2 :       STATUS,
: 735 0726 2 :       UPDFLG;       ! Data base update flag
: 736 0727 2
: 737 0728 2 :       FID = .NML$AB_ENTITYDATA [NML$C_LOGGING, EIT$B_FILEID]; ! Get file id
: 738 0729 2
: 739 0730 2 :       Add event flags.
: 740 0731 2
: 741 0732 2 :       IF NOT NML$ADDEVENTS (FALSE, .RECDSC, .SNK, .SNKADR, UPDFLG)
: 742 0733 2 :       THEN
: 743 0734 2 :           RETURN;
: 744 0735 2
: 745 0736 2 :       If the events parameter valid then write the record back to the file.

```

B
C
C
O
F
F
E
R
E
D
I
N
G
I
N
F
O
R
M
A
T
I
O
N
I
S
P
R
O
H
I
B
I
T
E
D
B
Y
T
H
E
A
U
T
H
O
R
I
T
Y
O
F
T
H
E
A
M
S
T
R
O
N
O
M
I
C
A
L
S
O
C
I
E
T
Y
O
F
C
O
M
P
U
T
E
R
S
I
N
C
O
R
P
O
R
A
T
E
D
I
N
1
9
8
9

```

: 746      0737 2 ! Otherwise, delete the record.
: 747      0738 ~
: 748      0739 ~
: 749      0740 ~
: 750      0741 ~
: 751      0742 ~
: 752      0743 ~
: 753      0744 ~
: 754      0745 ~
: 755      0746 ~
: 756      0747 ~
: 757      0748 ~
: 758      0749 ~
: 759      0750 ~
: 760      0751 1

```

```

IF .UPDFLG
THEN
    NML$WRITERECORD (.FID,
                    NML$C_LOGGING,
                    .KEY,
                    .RECDSC,
                    NMNSC_UPDATE_REC) ! Write the record
ELSE
    NML$DELETRECORD (.FID, .KEY, 0); ! Delete the record

NML$AB_MSGBLOCK [MSB$B_CODE] = NMASC_STS_SUC;

END; ! End of NML_PURLOGGING

```

		0004 0000 NML_PURLOGGING:				
	5E	04	C2 00002	.WORD	Save R2	: 0681
	52 00000000	00	9A 00005	SUBL2	#4, SP	: 0728
		5E	DD 0000C	MOVZBL	NML\$AB_ENTITYDATA+44, FID	: 0732
	7E	08	AC 3C 0000E	PL'SHL	SP	
		04	AC DD 00012	MOVZWL	SNKADR, -(SP)	
		0C	AC DD 00015	PUSHL	SNK	
		7E	D4 00018	PUSHL	RECDSC	
00000000G	00	05	FB 0001A	CLRL	-(SP)	
	2D	50	E9 0C021	CALLS	#5, NML\$ADDEVENTS	
	15	6E	E9 00024	BLBC	R0, 3\$	
		02	DD 00027	BLBC	UPDFLG, 1\$: 0739
		0C	AC DD 00029	PUSHL	#2	: 0741
		10	AC DD 0002C	PUSHL	RECDSC	: 0744
		01	DD 0002F	PUSHL	KEY	: 0743
		52	DD 00031	PUSHL	#1	: 0741
00000000G	00	05	FB 00033	PUSHL	FID	
		0E	11 0003A	CALLS	#5, NML\$WRITERECORD	
		7E	D4 0003C 1\$:	BRB	2\$	
		10	AC DD 0003E	CLRL	-(SP)	: 0747
		52	DD 00041	PUSHL	KEY	
00000000G	00	03	FB 00043	PUSHL	FID	
00000000G	00	G1	90 0004A 2\$:	CALLS	#3, NML\$DELETRECORD	: 0749
		04	00051 3\$:	MOVB	#1, NML\$AB_MSGBLOCK+4	: 0751
				RET		

: Routine Size: 82 bytes, Routine Base: \$CODE\$ + 0414

```

0752 1 %SBTTL 'NML_PURLOGALL Purge logging sink parameters'
0753 1 ROUTINE NML_PURLOGALL (SNK) =
0754 1
0755 1 ++
0756 1 FUNCTIONAL DESCRIPTION:
0757 1
0758 1     This routine performs common purge functions for both singular
0759 1     and plural requests.
0760 1
0761 1 FORMAL PARAMETERS:
0762 1
0763 1     SNK           Sink type code.
0764 1
0765 1 IMPLICIT INPUTS:
0766 1
0767 1     NONE
0768 1
0769 1 IMPLICIT OUTPUTS:
0770 1
0771 1     NONE
0772 1
0773 1 ROUTINE VALUE:
0774 1 COMPLETION CODES:
0775 1
0776 1     TRUE is returned if any operations have been performed indicating
0777 1     that a status message should be sent.  If no operations were
0778 1     performed then no data for the specified sink type was found so
0779 1     no status message should be sent.
0780 1
0781 1 SIDE EFFECTS:
0782 1
0783 1     NONE
0784 1
0785 1 --
0786 1
0787 2 BEGIN
0788 2
0789 2 LOCAL
0790 2     FID,           ! File id code
0791 2     FLDSIZE,
0792 2     FLDADR,
0793 2     KEY,          ! Record key
0794 2     MSGFLG : BYTE, ! Response message flag
0795 2     OWNER,       ! Search key
0796 2     PRMDSK : VECTOR [2],
0797 2     RECDSC : VECTOR [2],
0798 2     SRCPTR,     ! Pointer to source block
0799 2     STATUS;
0800 2
0801 2
0802 2 Purge the logging sink data.
0803 2
0804 2     MSGFLG = FALSE; ! No response messages
0805 2     FID = .NML$AB_ENTITYDATA [NML$C_SINK, EIT$B_FILEID]; ! Get file id
0806 2     OWNER = .NML$AB_ENTITYDATA [NML$C_SINK, EIT$W_KEY]; ! Get search key
0807 2     KEY = 0; ! Initialize record key
0808 2

```

```

0809 2 : Find the record in the file.
0810 2
0811 2 IF NMLSMATCHRECORD (.FID, NML$GQ_RECBFDSC, KEY,
0812 2     .OWNER, 1, SNK,
0813 2     0, 0, 0, ' No qualifier
0814 2     RECD$C)
0815 2 THEN
0816 2     BEGIN
0817 2
0818 2     MSGFLG = TRUE; ! Set response message flag
0819 2     NML_PURENTITY (NML$C_SINK, TRUE, RECD$C, KEY);
0820 2
0821 2     END;
0822 2
0823 2 : Purge the logging filter data.
0824 2
0825 2 FID = .NML$AB_ENTITYDATA [NML$C_LOGGING, EIT$B_FILEID]; ! Get file id
0826 2 OWNER = .NML$AB_ENTITYDATA [NML$C_LOGGING, EIT$W_KEY]; ! Get search key
0827 2 KEY = 0;
0828 2
0829 2 WHILE NMLSMATCHRECORD (.FID,
0830 2     NML$GQ_RECBFDSC,
0831 2     KEY,
0832 2     .OWNER, 0, 0,
0833 2     0, 0, 0, ' No qualifier
0834 2     RECD$C) DO
0835 2     BEGIN
0836 2
0837 2 : Find event parameter in record.
0838 2
0839 2     FLDADR = 0;
0840 2     STATUS = NML$SEARCHFLD (RECD$C,
0841 2         NML$C_PCLO_EVE,
0842 2         FLDSIZE,
0843 2         FLDADR);
0844 2
0845 2     IF .STATUS
0846 2     THEN
0847 2         BEGIN
0848 2
0849 2         CH$MOVE (.FLDSIZE, .FLDADR, .NML$Q_EVTBFDSC [1]);
0850 2         PRMD$C [0] = .FLDSIZE;
0851 2         PRMD$C [1] = .NML$Q_EVTBFDSC [1];
0852 2
0853 2         END
0854 2     ELSE
0855 2         BEGIN
0856 2
0857 2         PRMD$C [0] = 0;
0858 2         PRMD$C [1] = .NML$Q_EVTBFDSC [1];
0859 2
0860 2         END;
0861 2
0862 2 : Clear event filters.
0863 2
0864 2     SRCPTR = 0;
0865 2     WHILE NML$GETNXTSNK (PRMD$C, .SNK, SRCPTR) DO
0866 2         BEGIN

```

```

876 0866 4
877 0867 4      MSGFLG = TRUE;          Set response message flag
878 0868 4      NML$REMSRC (PRMDSC, .SRCPTR);
879 0869 4
880 0870 3      END;
881 0871 3
882 0872 3      IF .MSGFLG
883 0873 3      THEN
884 0874 4          BEGIN
885 0875 4
886 0876 4          IF NOT NML$SAVEEVENTS (NML$K_MAX_REC_DATA,
887 0877 4              .PRMDSC [0],
888 0878 4              .PRMDSC [1],
889 0879 4              RECDSC)
890 0880 4          THEN
891 0881 4              EXITLOOP;
892 0882 4
893 0883 4      ! If source was cleared successfully and there are still filters
894 0884 4      ! remaining for this sink node then write the record back to file.
895 0885 4      !
896 0886 4      IF .PRMDSC [0] EQLU 0
897 0887 4      THEN
898 0888 4          NML$DELETRECORD (.FID, KEY, 0)
899 0889 4      ELSE
900 0890 4          NML$WRITERECORD (.FID, NML$C_SINK, KEY, RECDSC, NMNSC_UPDATE_REC);
901 0891 4
902 0892 4      NML$AB_MSGBLOCK [MSB$B_CODE] = NMASC_STS_SUC;
903 0893 4
904 0894 3      END;
905 0895 3
906 0896 3      KEY = .KEY + 1;
907 0897 3
908 0898 2      END;
909 0899 2
910 0900 2      RETURN .MSGFLG
911 0901 2
912 0902 1      END;

```

! End of NML_PURLOGALL

OFFC 0000 NML_PURLOGALL:						
5B	0000000G	00	9E 00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 0753
5E		20	C2 00009	MOVAB	NML\$AB_ENTITYDATA+88, R11	:
		58	94 0000C	SUBL2	#32, SP	:
57		6B	9A 0000E	CLRB	MSGFLG	: 0804
59	03	AB	3C 00011	MOVZBL	NML\$AB_ENTITYDATA+88, FID	: 0805
	0C	AE	D4 00015	MOVZWL	NML\$AB_ENTITYDATA+91, OWNER	: 0806
	10	AE	9F 00018	CLRL	KEY	: 0807
		7E	7C 0001B	PUSHAB	RECDSC	: 0811
		7E	D4 0001D	CLRQ	-(SP)	:
	04	AC	9r 0001F	CLRL	-(SP)	:
		01	DD 00022	PUSHAB	SNK	:
		59	DD 00024	PUSHL	#1	:
	28	AE	9F 00026	PUSHL	OWNER	: 0812
				PUSHAB	KEY	: 0811

			00000000G	00	9F	00029		PUSHAB	NML\$GQ_RECBFDSC		
				57	DD	0002F		PUSHL	FID		
00000000G	00			0A	FB	00031		CALLS	#10, NML\$MATCHRECORD		
	12			50	E9	00038		BLBC	RO, 1\$		
	58			01	90	0003B		MOVB	#1, MSGFLG		0818
		0C		AE	9F	0003E		PUSHAB	KEY		0819
		14		AE	9F	00041		PUSHAB	RECDSC		
				01	DD	00044		PUSHL	#1		
				02	DD	00046		PUSHL	#2		
FEE0	CF			04	FB	00048		CALLS	#4, NML_PURENTITY		
	57		D4	AB	9A	0004D	1\$:	MOVZBL	NML\$AB_ENTITYDATA+44, FID		0825
	59		D7	AB	3C	00051		MOVZWL	NML\$AB_ENTITYDATA+47, OWNER		0826
			0C	AE	D4	00055		CLRL	KEY		0827
			10	AE	9F	00058	2\$:	PUSHAB	RECDSC		0829
				7E	7C	0005B		CLRQ	-(SP)		
				7E	7C	0005D		CLRQ	-(SP)		
				7E	D4	0005F		CLRL	-(SP)		
				59	DD	00061		PUSHL	OWNER		0832
		28		AE	9F	00063		PUSHAB	KEY		0829
			00000000G	00	9F	00066		PUSHAB	NML\$GQ_RECBFDSC		
				57	DD	0006C		PUSHL	FID		
00000000G	00			0A	FB	0006E		CALLS	#10, NML\$MATCHRECORD		
	76			50	E9	00075		BLBC	RO, 7\$		
				6E	D4	00078		CLRL	FLDADR		0839
				5E	DD	0007A		PUSHL	SP		0840
		08		AE	9F	0007C		PUSHAB	FLDSIZE		
		7E		8F	9A	0007F		MOVZBL	#201, -(SP)		
		1C		AE	9F	00083		PUSHAB	RECDSC		
00000000G	00			04	FB	00086		CALLS	#4, NML\$SEARCHFLD		
	5A			50	DD	0008D		MOVL	RO, STATUS		
	56	00000000'		00	DD	00090		MOVL	NML\$Q_EVTBFDSC+4, R6		0848
	0D			5A	E9	00097		BLBC	STATUS, 3\$		0844
66	00		04	AE	28	0009A		MOVC3	FLDSIZE, @FLDADR, (R6)		0848
	18		04	AE	DD	000A0		MOVL	FLDSIZE, PRMDSC		0849
				03	11	000A5		BRB	4\$		0850
			18	AE	D4	000A7	3\$:	CLRL	PRMDSC		0856
				56	DD	000AA	4\$:	MOVL	R6, PRMDSC+4		0857
			08	AE	D4	000AE		CLRL	SRCPTR		0863
			08	AE	9F	000B1	5\$:	PUSHAB	SRCPTR		0864
			04	AC	DD	000B4		PUSHL	SNK		
			20	AE	9F	000B7		PUSHAB	PRMDSC		
00000000G	00			03	FB	000BA		CALLS	#3, NML\$GETNXTSNK		
	12			50	E9	000C1		BLBC	RO, 6\$		
	58			01	90	000C4		MOVB	#1, MSGFLG		0867
		08		AE	DD	000C7		PUSHL	SRCPTR		0868
		1C		AE	9F	000CA		PUSHAB	PRMDSC		
00000000G	00			02	FB	000CD		CALLS	#2, NML\$REMSRC		
				DB	11	000D4		BRB	5\$		0864
			47	58	E9	000D6	6\$:	BLBC	MSGFLG, 10\$		0872
			10	AE	9F	000D9		PUSHAB	RECDSC		0876
			20	AE	DD	000DC		PUSHL	PRMDSC+4		0878
			20	AE	DD	000DF		PUSHL	PRMDSC		0877
	7E	03F6		8F	3C	000E2		MOVZWL	#1014, -(SP)		0876
00000000G	00			04	FB	000E7		CALLS	#4, NML\$SAVEVENTS		
	35			50	E9	000EE	7\$:	BLBC	RO, 11\$		
			18	AE	DS	000F1		TSTL	PRMDSC		0886
				10	12	000F4		BNEQ	8\$		

NML\$PURGE
V04-000

NML PURGE permanent parameter module
NML_PURLOGALL Purge logging sink parameters

G 16
16-Sep-1984 00:27:19
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[NML.SRC]NMLPURGE.B32;1 (11) Page 31

		7E	D4	000F6		CLRL	-(SP)		: 0888
		AE	9F	000F8		PUSHAB	KEY		: ..
		57	DD	000FB		PUSHL	FID		: ..
00000000G	00	03	FB	000FD		CALLS	#3, NML\$DELETRECORD		: ..
		13	11	00104		BRB	9\$: ..
		02	DD	00106	8\$:	PUSHL	#2		: .. 0890
		AE	9F	00108		PUSHAB	RECDSC		: ..
		AE	9F	0010B		PUSHAB	KEY		: ..
		02	DD	0010E		PUSHL	#2		: ..
		57	DD	00110		PUSHL	FID		: ..
00000000G	00	05	FB	00112		CALLS	#5, NML\$WRITERECORD		: ..
00000000G	00	01	90	00119	9\$:	MOVB	#1, NML\$AB_MSGBLOCK+4		: .. 0892
		AE	D6	00120	10\$:	INCL	KEY		: .. 0896
		FF	32	31	00123	BRW	2\$: .. 0829
	50	58	9A	00126	11\$:	MOVZBL	MSGFLG, R0		: .. 0900
			04	00129		RET			: .. 0902

: Routine Size: 298 bytes, Routine Base: \$CODE\$ + 0466

NML\$PURGE
V04-000

NML PURGE permanent parameter module
NML_PURLOGALL Purge logging sink parameters

H 16
16-Sep-1984 00:27:19
14-Sep-1984 12:50:16

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[NML.SRC]NMLPURGE.B32;1 (12) Page 32

: 914 0903 1 END
: 915 0904 1
: 916 0905 0 ELUDOM

: End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	1096	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	12	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	1424	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
-\$255\$DUA28:[NML.OBJ]NMLLIB.L32;1	341	44	12	27	00:00.1
-\$255\$DUA28:[SHRLIB]NMLIBRY.L32;1	887	11	1	47	00:00.2
-\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	1	0	581	00:02.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:NMLPURGE/OBJ=OBJ\$:NMLPURGE MSRC\$:NMLPURGE/UPDATE=(ENH\$:NMLPURGE)

: Size: 1424 code + 1108 data bytes
: Run Time: 00:24.0
: Elapsed Time: 00:59.9
: Lines/CPU Min: 2264
: Lexemes/CPU-Min: 11467
: Memory Used: 141 pages
: Compilation Complete

The image displays a grid of 100 terminal windows, arranged in 10 rows and 10 columns. Each window contains text-based output from a VAX/VMS system. The text is dense and includes various system messages, error codes, and diagnostic information. Some windows are more legible than others, showing titles like 'NMLPURGE LIS', 'NMLPARINI LIS', 'NMLNOOFTL LIS', 'NMLREAD LIS', 'NMLPARPRM LIS', and 'NMLPMANTP LIS'. The overall appearance is that of a multi-user terminal session or a system diagnostic tool output.