

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

_S
Ps
NP
NP
SG
SO
NP
PA
_L



```

NN      NN  MM      MM  LL      DDDDDDDD  EEEEEEEEE  FFFFFFFF  IIIIII  NN      NN  EEEEEEEEE
NN      NN  MM      MM  LL      DDDDDDDD  EEEEEEEEE  FFFFFFFF  IIIIII  NN      NN  EEEEEEEEE
NN      NN  MMMM    MMMM LL      DD        EE        FF        II      NN      NN  EE
NN      NN  MMMM    MMMM LL      DD        EE        FF        II      NN      NN  EE
NNNN    NN  MM      MM  LL      DD        DD        EE        FF        II      NNNN   NN  EE
NNNN    NN  MM      MM  LL      DD        DD        EE        FF        II      NNNN   NN  EE
NN  NN   NN  MM      MM  LL      DD        DD  EEEEEEE  FFFFFFFF  II      NN  NN  EEEEEEE
NN  NN   NN  MM      MM  LL      DD        DD  EEEEEEE  FFFFFFFF  II      NN  NN  EEEEEEE
NN      NNNN  MM      MM  LL      DD        DD  EE        FF        II      NN  NNNN  EE
NN      NNNN  MM      MM  LL      DD        DD  EE        FF        II      NN  NNNN  EE
NN      NN   MM      MM  LL      DD        DD  EE        FF        II      NN      NN  EE
NN      NN   MM      MM  LL      DD        DD  EE        FF        II      NN      NN  EE
NN      NN   MM      MM  LLLLLLLLLL DDDDDDDD EEEEEEEEE  FF        IIIIII NN      NN  EEEEEEEEE
NN      NN   MM      MM  LLLLLLLLLL DDDDDDDD EEEEEEEEE  FF        IIIIII NN      NN  EEEEEEEEE

```

```

LL      IIIIII  SSSSSSS
LL      IIIIII  SSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSS
LL      II      SSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLL IIIIII  SSSSSSS
LLLLLLLLL IIIIII  SSSSSSS

```

```

1 0001 0 %TITLE 'NML DEFINE permanent parameter module'
2 0002 0 MODULE NML$DEFINE (
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 Network Management Listener
36 0036 1
37 0037 1 ABSTRACT:
38 0038 1
39 0039 1     This module contains routines to handle the NCP DEFINE command.
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     V03-010 MKP0013      Kathy Perko      20-April-1984
50 0050 1     If DEF NODE nnn NAME mmm supplies a duplicate name, fix
51 0051 1     the error message so it indicates whether the duplicate is
52 0052 1     the exec or a remote.
53 0053 1
54 0054 1     V03-009 MKP0012      Kathy Perko      9-April-1984
55 0055 1     If talking to a Phase III NCP, return those nodes in the
56 0056 1     executor's area with a zero area number. This will make
57 0057 1     them intelligible on the Phase III system. Node numbers

```

```

: 58      0058 1  |
: 59      0059 1  |
: 60      0060 1  |
: 61      0061 1  |
: 62      0062 1  |
: 63      0063 1  |
: 64      0064 1  |
: 65      0065 1  |
: 66      0066 1  |
: 67      0067 1  |
: 68      0068 1  |
: 69      0069 1  |
: 70      0070 1  |
: 71      0071 1  |
: 72      0072 1  |
: 73      0073 1  |
: 74      0074 1  |
: 75      0075 1  |
: 76      0076 1  |
: 77      0077 1  |
: 78      0078 1  |
: 79      0079 1  |
: 80      0080 1  |
: 81      0081 1  |
: 82      0082 1  |
: 83      0083 1  |
: 84      0084 1  |
: 85      0085 1  |
: 86      0086 1  |
: 87      0087 1  |
: 88      0088 1  |
: 89      0089 1  |
: 90      0090 1  |
: 91      0091 1  |
: 92      0092 1  |
: 93      0093 1  |
: 94      0094 1  |
: 95      0095 1  |
: 96      0096 1  |
: 97      0097 1  |
: 98      0098 1  |
: 99      0099 1  |
: 100     0100 1  |
: 101     0101 1  |
: 102     0102 1  |
: 103     0103 1  |
: 104     0104 1  |
: 105     0105 1  |
: 106     0106 1  |
: 107     0107 1  |
: 108     0108 1  |

```

outside the executor's area are returned as is. This means they will be large, not easily understood numbers, but at least they will be unique.

V03-008 MKP0011 Kathy Perko 17-Feb-1984
Fix bug in DEFINE KNOWN NODES - NML\$GETREOWNER is being passed a descriptor incorrectly.

V03-007 MKP0010 Kathy Perko 9-Jan-1984
Add X25-Access Module entity.

V03-006 MKP0009 Kathy Perko 4-Aug-1983
Converting node database to use multiple ISAM keys. Make changes necessary in this module.

V03-005 MKP0008 Kathy Perko 25-April-1983
Add configurator module permanent database operations.

V03-004 MKP0007 Kathy Perko 14-Sept-1982
If a logging sink node is the executor node, save it in the permanent data base with an address of zero. This allows the database to be transported to another node, and still log the events to the local node.
Fix DEFINE KNOWN so that it uses a record owner to determine what records to modify. This allows for X25 and X29 which combine several databases in a single permanent database file.

V03-003 MKP0006 Kathy Perko 9-Sept-1982
Fix entity returned in NICE messages for X29-Server so that a length is included.

V03-002 MKP0005 Kathy Perko 28-June-1982
Redo qualifier handling to use the qualifier's index into the Parameter Semantic Table (PST) instead of the qualifier's Network Management parameter ID.
Add X29-Server entity.

V03-001 MKP0004 Kathy Perko 3-May-1982
Add qualifier matching to permanent data base operations.

V02-003 MKP0003 Kathy Perko 16-Nov-1981
Undo the previous fix. It's wrong.

V02-002 MKP0002 Kathy Perko 17-Sept-1981
Fix NML\$SAVEEVENTS so that a status of NML\$_STS_PMS is returned if there are no events.

V02-001 MKP0001 Kathy Perko 23-July-1981
Add circuit entity and multidrop lines

```

: 110 0109 1 %SBTTL 'Declarations'
: 111 0110 1
: 112 0111 1
: 113 0112 1 : : TABLE OF CONTENTS:
: 114 0113 1 : :
: 115 0114 1 : :
: 116 0115 1 FORWARD ROUTINE
: 117 0116 1     nml$defineknown      : NOVALUE,
: 118 0117 1     nml$defentity      : NOVALUE,
: 119 0118 1     nml_defentity      : NOVALUE,
: 120 0119 1     nml$define_known_nodes: NOVALUE,
: 121 0120 1     nml$define_node     : NOVALUE,
: 122 0121 1     nml_define_node     : NOVALUE,
: 123 0122 1     nml$defknownlog    : NOVALUE,
: 124 0123 1     nml$deflogging     : NOVALUE,
: 125 0124 1     nml_deflogging     : NOVALUE,
: 126 0125 1     nml$addevents,
: 127 0126 1     nml$saveevents,
: 128 0127 1     nml$getrecowner;
: 129 0128 1
: 130 0129 1 : :
: 131 0130 1 : : INCLUDE FILES:
: 132 0131 1 : :
: 133 0132 1
: 134 0133 1 LIBRARY 'LIB$:NMLLIB.L32';
: 135 0134 1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
: 136 0135 1 LIBRARY 'SYSSLIBRARY:STARLET.L32';
: 137 0136 1
: 138 0137 1 : :
: 139 0138 1 : : EQUATED SYMBOLS:
: 140 0139 1 : :
: 141 0140 1 : :
: 142 0141 1 : :
: 143 0142 1 : : OWN STORAGE:
: 144 0143 1 : :
: 145 0144 1 : :
: 146 0145 1 : :
: 147 0146 1 : : Entity buffer and descriptor.
: 148 0147 1 : :
: 149 0148 1 OWN
: 150 0149 1     NML$T_ENTBUFFER : BBLOCK [NML$K_ENTBUFLN],
: 151 0150 1     NML$Q_ENTBFDSC : DESCRIPTOR;
: 152 0151 1
: 153 0152 1 : :
: 154 0153 1 : : EXTERNAL REFERENCES:
: 155 0154 1 : :
: 156 0155 1 : :
: 157 0156 1 $NML_EXTDEF;
: 158 0157 1
: 159 0158 1 EXTERNAL
: 160 0159 1     nml$gb_ncp_version,
: 161 0160 1     nml$gw_perm_exec_addr: BBLOCK [2];
: 162 0161 1
: 163 0162 1 EXTERNAL LITERAL
: 164 0163 1     nml$_badevtupd,
: 165 0164 1     nml$_intevtovf,
: 166 0165 1     nml$_recadded,

```

```
.. 167      0166 1      nml$_norecown;  
.. 168      0167 1  
.. 169      0168 1  EXTERNAL ROUTINE  
.. 170      0169 1      nml$_insertfld,  
.. 171      0170 1      nml$_matchrecord,  
.. 172      0171 1      nml$_searchfld,  
.. 173      0172 1      nml$_add_fields,  
.. 174      0173 1      nml$_addfilters,  
.. 175      0174 1      nml$_getnxtsnk,  
.. 176      0175 1      nml$_bld_reply,  
.. 177      0176 1      nml$_error_1,  
.. 178      0177 1      nml$_openfile,  
.. 179      0178 1      nml$_read_known_node_rec,  
.. 180      0179 1      nml$_readrecord,  
.. 181      0180 1      nml$_writerecord,  
.. 182      0181 1      nml$_send;  
.. 183      0182 1
```

```

: 185 0183 1 %SBTTL 'NML$DEFINEKNOWN Define parameters for known entities'
: 186 0184 1 GLOBAL ROUTINE NML$DEFINEKNOWN (ENTITY, DUM1, DUM2,
: 187 0185 1 DUM3, DUM4, DUM5) : NOVALUE =
: 188 0186 1
: 189 0187 1 |++
: 190 0188 1 | FUNCTIONAL DESCRIPTION:
: 191 0189 1 | This routine defines a set of parameters in the permanent
: 192 0190 1 | data base entry for each entity of the specified type.
: 193 0191 1 |
: 194 0192 1 | FORMAL PARAMETERS:
: 195 0193 1 |
: 196 0194 1 | ENTITY Entity type code.
: 197 0195 1 | DUM1-DUM5 Not used.
: 198 0196 1 |
: 199 0197 1 | IMPLICIT INPUTS:
: 200 0198 1 |
: 201 0199 1 | NML$GL_PRS_FLGS Message parsing flags.
: 202 0200 1 |
: 203 0201 1 | SIDE EFFECTS:
: 204 0202 1 |
: 205 0203 1 | Signals errors.
: 206 0204 1 |
: 207 0205 1 | --
: 208 0206 1 |
: 209 0207 2 BEGIN
: 210 0208 2
: 211 0209 2 LOCAL
: 212 0210 2 fid, | File id code
: 213 0211 2 param_to_find, | Parameter to match in permanent database
: 214 0212 2 | record.
: 215 0213 2 msgsize, | Message size
: 216 0214 2 key, | Temporary record key buffer
: 217 0215 2 recdsc : DESCRIPTOR; | Record descriptor
: 218 0216 2
: 219 0217 2 fid = .nml$ab_entitydata [.entity, eit$b_fileid];
: 220 0218 2 param_to_find = .nml$ab_entitydata [.entity, eit$w_key];
: 221 0219 2
: 222 0220 2 | Some of the permanent database files have more than one entity in them.
: 223 0221 2 | For example, NETX25.DAT has all X25 entities (Access, Protocol, and Server).
: 224 0222 2 | Add parameters to every record in the file which contains a parameter
: 225 0223 2 | matching PARAM_TO_FIND.
: 226 0224 2 |
: 227 0225 2 key = 0;
: 228 0226 2 WHILE nml$matchrecord (.fid, nml$gq_recbfdsc, key,
: 229 0227 2 .param_to_find, 0, 0,
: 230 0228 2 0, 0, 0,
: 231 0229 2 recdsc) DO
: 232 0230 2 BEGIN
: 233 0231 2 nml$ab_msgblock [msb$l_flags] = 0; ! Initialize message flags
: 234 0232 2 |
: 235 0233 2 | Add the parameters to the entity's record, and write the updated record
: 236 0234 2 | back to the permanent database file.
: 237 0235 2 |
: 238 0236 2 nml_defentity (.entity, recdsc, key);
: 239 0237 2 |
: 240 0238 2 | Build and send the NICE response message.
: 241 0239 2

```

```

: 242      0240 3      nml$bld_reply (nml$ab_msgblock, msgsize);
: 243      0241 3      nml$send (nml$ab_sndbuffer, .msgsize);
: 244      0242 2      key = .key + 1;
: 245      0243 2      END;
: 246      0244 1      END;

```

! End of NML\$DEFINEKNOWN

```

.TITLE NML$DEFINE NML DEFINE permanent parameter modul
.IDENT \V04-000\
.PSECT $OWNS,NOEXE,2

```

```

00000 NML$T_ENTBUFFER:
      .BLKB 64
00040 NML$Q_ENTBFDSC:
      .BLKB 8

```

```

.EXTRN NML$GB_EVTSRCTYP
.EXTRN NML$GQ_EVTSRCDSK
.EXTRN NML$GW_EVTCLASS
.EXTRN NML$GB_EVTMSKTYP
.EXTRN NML$GQ_EVTMSKDSK
.EXTRN NML$GW_EVTSNKADR
.EXTRN NML$GW_ACP_CHAN
.EXTRN NML$GL_LOGMASK, NML$GQ_ENTSTRDSC
.EXTRN NML$AB_QIOBUFFER
.EXTRN NML$GQ_QIOBFDSK
.EXTRN NML$AB_EXEBUFFER
.EXTRN NML$GL_EXEDATPTR
.EXTRN NML$GQ_EXEDATDSK
.EXTRN NML$AB_EXEBFDSK
.EXTRN NML$AB_RCVBUFFER
.EXTRN NML$GQ_RCVBFDSK
.EXTRN NML$AB_SNDBUFFER
.EXTRN NML$GQ_SNDBFDSK
.EXTRN NML$GL_RCVDATLEN
.EXTRN NML$AB_CPTABLE, NML$AB_MSGBLOCK
.EXTRN NML$AB_ENTITY_ID
.EXTRN NML$AB_QUALIFIER_ID
.EXTRN NML$AB_ENTITYDATA
.EXTRN NML$AB_NML_NMV, NML$AB_PRMSEM
.EXTRN NML$AB_RECBUF, NML$AL_ENTINFNTAB
.EXTRN NML$AL_PERMINFTAB
.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$GQ_NETNAMDSK
.EXTRN NML$GQ_RECBFDSK
.EXTRN NML$GW_PRMDESCNT
.EXTRN NML$GB_NCP_VERSION

```

				001C 00000	.EXTRN NML\$GW PERM EXEC ADDR	
				54 00000000G 00 9E 00002	.EXTRN NML\$_BADEVTOPD, NML\$ INTEVTOVF	
				5E 10 C2 00009	.EXTRN NML\$_RECADDED, NML\$ NORECOWN	
50	04			AC 2C C5 0000C	.EXTRN NML\$INSERTFLD, NML\$MATCHRECORD	
				53 00000000G0040 9A 00011	.EXTRN NML\$SEARCHFLD, NML\$ADD_FIELDS	
				00000000G0040 9F 00019	.EXTRN NML\$ADDFILTERS, NML\$GETNXTSNK	
				52 9E 3C 00020	.EXTRN NML\$BLD_REPLY, NML\$ERROR 1	
				6E D4 00023	.EXTRN NML\$OPENFILE, NML\$READ KNOWN NODE_REC	
				08 AE 9F 00025 1\$:	.EXTRN NML\$READRECORD, NML\$WRITERECORD	
				7E 7C 00028	.EXTRN NML\$SEND	
				7E 7C 0002A	.PSECT \$CODE\$,NOWRT,2	
				7E D4 0002C	.ENTRY NML\$DEFINEKNOWN, Save R2,R3,R4	: 0184
				52 DD 0002E	MOVAB NML\$AB_MSGBLOCK, R4	
				1C AE 9F 00030	SUBL2 #16, SP	
				00000000G 00 9F 00033	MULL3 #44, ENTITY, R0	: 0217
				53 DD 00039	MOVZBL NML\$AB_ENTITYDATA[R0], FID	
00000000G 00				0A FB 0003B	PUSHAB NML\$AB_ENTITYDATA+3[R0]	: 0218
				31 50 E9 00042	MOVZWL @ (SP)+, PARAM_TO_FIND	
				64 D4 00045	CLRL KEY	: 0225
				5E DD 00047	PUSHAB RECDSC	: 0226
				0C AE 9F 00049	CLRQ -(SP)	
				04 AC DD 0004C	CLRQ -(SP)	
00000000V 00				03 FB 0004F	CLRL -(SP)	
				04 AE 9F 00056	PUSHL PARAM_TO_FIND	: 0227
				54 DD 00059	PUSHAB KEY	: 0226
00000000G 00				02 FB 0005B	PUSHAB NML\$GQ_RECBFDSC	
				04 AE DD 00062	PUSHL FID	
00000000G 00				00 9F 00065	CALLS #10, NML\$MATCHRECORD	
				02 FB 0006B	BLBC R0, 2\$	
				6E D6 00072	CLRL NML\$AB_MSGBLOCK	: 0231
				AF 11 00074	PUSHL SP	: 0236
				04 00076 2\$:	PUSHAB RECDSC	
					PUSHL ENTITY	
					CALLS #3, NML_DEFENTITY	
					PUSHAB MSGSIZE	: 0240
					PUSHL R4	
					CALLS #2, NML\$BLD_REPLY	
					PUSHL MSGSIZE	: 0241
					PUSHAB NML\$AB_SNDBUFFER	
					CALLS #2, NML\$SEND	
					INCL KEY	: 0242
					BRB 1\$: 0226
					RET	: 0244

; Routine Size: 119 bytes, Routine Base: \$CODE\$ + 0000

```

: 248 0245 1 %SBTTL 'NML$DEFENTITY Define entity parameters'
: 249 0246 1 GLOBAL ROUTINE NML$DEFENTITY (ENTITY, ENTITY_LEN, ENTITY_ADR,
: 250 0247 1 QUAL_PST, QUAL_LEN, QUAL_ADR) : NOVALUE =
: 251 0248 1
: 252 0249 1 !++
: 253 0250 1 ! FUNCTIONAL DESCRIPTION:
: 254 0251 1
: 255 0252 1 ! Add parameters to the permanent data base entry for the specified
: 256 0253 1 ! entity.
: 257 0254 1
: 258 0255 1 ! FORMAL PARAMETERS:
: 259 0256 1
: 260 0257 1 ! ENTITY Entity type code.
: 261 0258 1 ! ENTITY_LEN Byte count of entity id string.
: 262 0259 1 ! ENTITY_ADR Address of entity id string.
: 263 0260 1 ! QUAL_PST Qualifier PST entry address
: 264 0261 1 ! QUAL_LEN Qualifier ID byte count.
: 265 0262 1 ! QUAL_ADR Address of qualifier id string.
: 266 0263 1
: 267 0264 1 ! IMPLICIT INPUTS:
: 268 0265 1
: 269 0266 1 ! NML$GL_PRS_FLGS Message parsing flags.
: 270 0267 1
: 271 0268 1 ! --
: 272 0269 1
: 273 0270 2 BEGIN
: 274 0271 2
: 275 0272 2 LOCAL
: 276 0273 2 ! fid, ! File id code
: 277 0274 2 ! fldadr,
: 278 0275 2 ! fldsize,
: 279 0276 2 ! loop,
: 280 0277 2 ! msgsize, ! Message size
: 281 0278 2 ! key, ! Temporary record key buffer
: 282 0279 2 ! owner, ! Search key
: 283 0280 2 ! recdsc : DESCRIPTOR, ! Record descriptor
: 284 0281 2 ! status;
: 285 0282 2
: 286 0283 2 nml$ab_msgblock [msb$l_flags] = 0; ! Initialize message flags
: 287 0284 2
: 288 0285 2 ! Add parameters to every record in the file.
: 289 0286 2
: 290 0287 2 fid = .nml$ab_entitydata [.entity, eit$b_fileid]; ! Get file id
: 291 0288 2 owner = .nml$ab_entitydata [.entity, eit$w_key]; ! Get search key
: 292 0289 2
: 293 0290 2 key = 0; ! Initialize record key
: 294 0291 2 status = nml$matchrecord (.fid,
: 295 0292 2 nml$gq_recbfdsc,
: 296 0293 2 key,
: 297 0294 2 .owner, .entity_len, .entity_adr,
: 298 0295 2 .qual_pst, .qual_len, .qual_adr,
: 299 0296 2 recdsc);
: 300 0297 2 IF NOT .status THEN
: 301 0298 2 BEGIN
: 302 0299 2
: 303 0300 2 ! Initialize record descriptor and add entity ID field to record buffer.
: 304 0301 2

```

```

305 0302 3 recdsc [dsc$w_length] = 0;
306 0303 3 recdsc [dsc$a_pointer] = .nml$gq_recbfdsc [dsc$a_pointer] + 2;
307 0304 3 nma$insertfld (nml$k_max_rec_data,
308 0305 3 .owner, .entity_len, .entity_adr, recdsc);
309 0306 3
310 0307 3 | If there's a qualifier associated with this entity, add that to
311 0308 3 | the record as well.
312 0309 3
313 0310 3 IF .nml$gl_prs_flg [nml$v_prs_qualifier] THEN
314 0311 4 BEGIN
315 0312 4 MAP qual_pst: REF BBLOCK;
316 0313 4 nma$insertfld (nml$k_max_rec_data,
317 0314 4 .qual_pst [pst$w_dataid], .qual_len, .qual_adr,
318 0315 4 recdsc);
319 0316 4 END;
320 0317 3 status = nml$_sts_suc;
321 0318 3 END;
322 0319 3
323 0320 3 | If everything is all right then add the parameters.
324 0321 3
325 0322 3 IF .status THEN
326 0323 3 nml_defentity (.entity, recdsc, key);
327 0324 3
328 0325 3 | Build and send the response message.
329 0326 3
330 0327 3 nml$bld_reply (nml$ab_msgblock, msgsize);
331 0328 3 nml$send (nml$ab_sndbuffer, .msgsize);
332 0329 3
333 0330 1 END; ! End of NML$DEFENTITY

```

				003C 0000	.ENTRY NML\$DEFENTITY, Save R2,R3,R4,R5	: 0246
	55	00000000G	00	9E 00002	MOVAB NML\$AB_MSGBLOCK, R5	
	54	00000000G	00	9E 00009	MOVAB NMA\$INSERTFLD, R4	
	5E		10	C2 00010	SUBL2 #16, SP	
			65	D4 00013	CLRL NML\$AB_MSGBLOCK	: 0283
50	04	AC	2C	C5 00015	MULL3 #44, ENTITY, R0	: 0287
	51	00000000G0040	9A	0001A	MOVZBL NML\$AB_ENTITYDATA[R0], FID	
		00000000G0040	9F	00022	PUSHAB NML\$AB_ENTITYDATA+3[R0]	: 0288
	53		9E	3C 00029	MOVZWL @(SP)+, OWNER	
			6E	D4 0002C	CLRL KEY	: 0290
			08	AE 9F 0002E	PUSHAB RECDSC	: 0291
	7E		14	AC 7D 00031	MOVQ QUAL_LEN, -(SP)	: 0295
	7E		0C	AC 7D 00035	MOVQ ENTITY_ADR, -(SP)	: 0294
			08	AC DD 00039	PUSHL ENTITY_LEN	
				53 DD 0003C	PUSHL OWNER	
			1C	AE 9F 0003E	PUSHAB KEY	: 0291
		00000000G	00	9F 00041	PUSHAB NML\$GQ_RECBFDSC	
			51	DD 00047	PUSHL FID	
		00000000G	00	0A FB 00049	CALLS #10, NML\$MATCHRECORD	
			52	50 D0 00050	MOVL R0, STATUS	
			3E	52 E8 00053	BLBS STATUS, 2\$: 0297
			08	AE B4 00056	CLRW RECDSC	: 0302
0C	AE	00000000G	00	02 C1 00059	ADDL3 #2, NML\$GQ_RECBFDSC+4, RECDSC+4	: 0303

NML\$DEFINE
V04-000

NML DEFINE permanent parameter module
NML\$DEFENTITY Define entity parameters

C 5
16-Sep-1984 00:12:41
14-Sep-1984 12:50:07

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLDEFINE.B32;1

Page 10
(4)

		08	AE	9F	00062	PUSHAB	RECDSC	:	0304		
	7E	08	AC	7D	00065	MOVQ	ENTITY_LEN, -(SP)	:	0305		
			53	DD	00069	PUSHL	OWNER	:			
	7E	03F6	8F	3C	0006B	MOVZWL	#1014, -(SP)	:	0304		
	64		05	FB	00070	CALLS	#5, NML\$INSERTFLD	:			
13	00000000G	00	02	E1	00073	BBC	#2, NML\$GL_PRS_FLGS, 1\$:	0310		
			08	AE	9F	0007B	PUSHAB	RECDSC	:	0313	
	7E		14	AC	7D	0007E	MOVQ	QUAL_LEN, -(SP)	:	0314	
	7E		10	BC	3C	00082	MOVZWL	@QUAL_PST, -(SP)	:		
	7E	03F6	8F	3C	00086	MOVZWL	#1014, -(SP)	:	0313		
	64		05	FB	0008B	CALLS	#5, NML\$INSERTFLD	:			
	52		01	D0	0008E	1\$:	MOVL	#1, STATUS	:	0317	
	0F		52	E9	00091	BLBC	STATUS, 3\$:	0322		
			5E	DD	00094	2\$:	PUSHL	SP	:	0323	
			0C	AE	9F	00096	PUSHAB	RECDSC	:		
			04	AC	DD	00099	PUSHL	ENTITY	:		
00000000V	00		03	FB	0009C	CALLS	#3, NML_DEFENTITY	:			
			04	AE	9F	000A3	3\$:	PUSHAB	MSGSIZE	:	0327
			55	DD	000A6	PUSHL	R5	:			
00000000G	00		02	FB	000AB	CALLS	#2, NML\$BLD_REPLY	:			
			04	AE	DD	000AF	PUSHL	MSGSIZE	:	0328	
		00000000G	00	9F	000B2	PUSHAB	NML\$AB_SNDBUFFER	:			
00000000G	00		02	FB	000B8	CALLS	#2, NML\$SEND	:			
			04	000BF		RET		:	0330		

; Routine Size: 192 bytes, Routine Base: \$CODE\$ + 0077

```

335 0331 1 %SBTTL 'NML_DEFENTITY Define entity parameters'
336 0332 1 ROUTINE NML_DEFENTITY (ENTITY, RECDSC, KEY) : NOVALUE =
337 0333 1
338 0334 1 |++
339 0335 1 | FUNCTIONAL DESCRIPTION:
340 0336 1 |
341 0337 1 |     This routine performs common DEFINE functions for both singular
342 0338 1 |     and plural requests.
343 0339 1 |
344 0340 1 | FORMAL PARAMETERS:
345 0341 1 |
346 0342 1 |     ENTITY      Entity type code.
347 0343 1 |     RECDSC     Address of current record descriptor.
348 0344 1 |     KEY        Address of current record key.
349 0345 1 |
350 0346 1 | --
351 0347 1
352 0348 2 BEGIN
353 0349 2
354 0350 2 MAP
355 0351 2     recdsc : REF DESCRIPTOR;      ! Record descriptor
356 0352 2
357 0353 2 LOCAL
358 0354 2     fid;                          ! File id code
359 0355 2
360 0356 2     fid = .nml$ab_entitydata [.entity, eit$b_fileid]; ! Get file id
361 0357 2
362 0358 2     Add the NICE command parameter fields to the permanent database record.
363 0359 2
364 0360 2 IF nml$add_fields (nml$k_max_rec_data, .recdsc) THEN
365 0361 2     BEGIN
366 0362 2         nml$writerecord (.fid, .entity, .key, .recdsc, 0);
367 0363 2         nml$ab_msgblock [msb$b_code] = nma$c_sts_suc;
368 0364 2     END;
369 0365 2
370 0366 2     Build response message.
371 0367 2
372 0368 2     nml$q_entbfdsc [dsc$w_length] = nml$k_entbuflen; ! Initialize entity descriptor
373 0369 2     nml$q_entbfdsc [dsc$a_pointer] = nml$e_entbuffer;
374 0370 2
375 0371 2     nml$getrecowner (.recdsc,
376 0372 2         .entity,
377 0373 2         nml$q_entbfdsc,
378 0374 2         nml$q_entbfdsc [dsc$w_length]);
379 0375 2
380 0376 2     nml$ab_msgblock [msb$v_entd fld] = 1;      ! Set entity descriptor flag
381 0377 2     nml$ab_msgblock [msb$a_entity] = nml$q_entbfdsc; ! Add entity descriptor pointer
382 0378 2
383 0379 1 END;                                     ! End of NML_DEFENTITY

```

```

                                001C 00000 NML_DEFENTITY:
                                .WORD      Save R2,R3,R4
54 00000000G 00 9E 00002      MOVAB   NML$AB_MSGBLOCK+4, R4

```

```

: 0332
:
```

NML\$DEFINE
V04-000

NML DEFINE permanent parameter module
NML_DEFENTITY Define entity parameters

E 5
16-Sep-1984 00:12:41
14-Sep-1984 12:50:07

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLDEFINE.B32;1

Page 12
(5)

```
50      04      53 00000000' 00 9E 00009
          AC      2C      C5 00010
          52 00000000G0040 9A 00015
              08      AC      DD 0001D
          00000000G 7E      03F6 8F 3C 00020
              00      02      FB 00025
              17      50      E9 0002C
              7E      D4 0002F
              08      AC      DD 00031
              0C      AC      DD 00034
              04      AC      DD 00037
              52      DD 0003A
          00000000G 00      05      FB 0003C
              64      01      90 00043
              63      40      8F 9B 00046 1$:
          04      A3      C0      A3 9E 0004A
              53      DD 0004F
              53      DD 00051
              04      AC      DD 00053
              08      AC      DD 00056
          00000000V 00      04      FB 00059
              FC      A4      10      88 00060
              10      A4      63      9E 00064
              04      00068
```

```
MOVAB NML$Q_ENTBFDSC, R3
MULL3 #4, ENTITY, R0
MOVZBL NML$AB_ENTITYDATA[R0], FID
PUSHL RECDSC
MOVZWL #1014, -(SP)
CALLS #2, NML$ADD_FIELDS
BLBC R0, 1$
CLRL -(SP)
PUSHL RECDSC
PUSHL KEY
PUSHL ENTITY
PUSHL FID
CALLS #5, NML$WRITERECORD
MOVB #1, NML$AB_MSGBLOCK+4
MOVZBW #6, NML$Q_ENTBFDSC
MOVAB NML$T_ENTBUFFER, NML$Q_ENTBFDSC+4
PUSHL R3
PUSHL R3
PUSHL ENTITY
PUSHL RECDSC
CALLS #4, NML$GETRECOWNER
BISB2 #16, NML$AB_MSGBLOCK
MOVAB NML$Q_ENTBFDSC, NML$AB_MSGBLOCK+20
RET
```

```
..:
..: 0356
..:
..: 0360
..:
..:
..: 0362
..:
..:
..:
..: 0363
..: 0368
..: 0369
..: 0374
..: 0371
..: 0374
..:
..:
..: 0376
..: 0377
..: 0379
```

; Routine Size: 105 bytes, Routine Base: \$CODE\$ + 0137

NM
VO

```

: 385 0380 1 %SBTTL 'NML$DEFINE_KNOWN_NODES Define parameters for known entities'
: 386 0381 1 GLOBAL ROUTINE NML$DEFINE_KNOWN_NODES (ENTITY, DUM1, DUM2,
: 387 0382 1 DUM3, DUM4, DUM5) : NOVALUE =
: 388 0383 1
: 389 0384 1 !++
: 390 0385 1 FUNCTIONAL DESCRIPTION:
: 391 0386 1 This routine defines a set of parameters in the permanent
: 392 0387 1 data base entry for each entity in the node database. The
: 393 0388 1 nodes updated in this case are the executor and the remotes.
: 394 0389 1
: 395 0390 1 FORMAL PARAMETERS:
: 396 0391 1 ENTITY Entity type code.
: 397 0392 1 DUM1-DUM5 Not used.
: 398 0393 1
: 399 0394 1 IMPLICIT INPUTS:
: 400 0395 1 NML$GL_PRS_FLGS Message parsing flags.
: 401 0396 1
: 402 0397 1 SIDE EFFECTS:
: 403 0398 1 Signals errors.
: 404 0399 1
: 405 0400 1
: 406 0401 1
: 407 0402 1
: 408 0403 1
: 409 0404 1
: 410 0405 2 BEGIN
: 411 0406 2
: 412 0407 2 LOCAL
: 413 0408 2 node_type,
: 414 0409 2 recdsc: DESCRIPTOR,
: 415 0410 2 rewind_flag,
: 416 0411 2 msgsize;
: 417 0412 2
: 418 0413 2
: 419 0414 2 Loopnodes can only have the parameter CIRCUIT, which must be unique. So
: 420 0415 2 DEFINE KNOWN never applies to loopnodes. Apply the DEFINE to the executor
: 421 0416 2 first, and then to the remote nodes.
: 422 0417 2
: 423 0418 2 nml$define_node (nml$c_executor, 2, 0);
: 424 0419 2 rewind_flag = true;
: 425 0420 2 WHILE nml$read_known_node_rec (nml$c_node,
: 426 0421 2 nml$gq_recbfdsc,
: 427 0422 2 recdsc,
: 428 0423 2 .rewind_flag) DO
: 429 0424 2
: 430 0425 2 BEGIN
: 431 0426 2 rewind_flag = false;
: 432 0427 2 nml$ab_msgblock [msb$l_flags] = 0;
: 433 0428 2
: 434 0429 2 Add the parameters to the node's record, and write the updated
: 435 0430 2 record back to the node permanent database file.
: 436 0431 2 nml_define_node (.entity, recdsc, nm$c_update_rec);
: 437 0432 2
: 438 0433 2 Set up the NICE message block so that the modified node's id is
: 439 0434 2 included in the NICE response message. This is done because the
: 440 0435 2 architecture requires that, for KNOWN entity operations, the entitie IDs
: 441 0436 2 of the modified entities be returned in the NICE responses.

```

```

: 442 0437 !
: 443 0438 nml$q_entbfdsc [dsc$w_length] = nml$k_entbuflen;
: 444 0439 nml$q_entbfdsc [dsc$a_pointer] = nml$e_entbuffer;
: 445 0440 nml$getrecowner (recdsc,
: 446 0441     .entity,
: 447 0442     nml$q_entbfdsc,
: 448 0443     nml$q_entbfdsc [dsc$w_length]);
: 449 0444 nml$ab_msgblock [msb$w_entd fld] = 1;
: 450 0445 nml$ab_msgblock [msb$a_entity] = nml$q_entbfdsc;
: 451 0446
: 452 0447 | Build and send NICE response message - one for each node updated.
: 453 0448 |
: 454 0449 nml$bld_reply (nml$ab_msgblock, msgsize);
: 455 0450 nml$send (nml$ab_sndbuffer, .msgsize);
: 456 0451 END;
: 457 0452 | END;

```

! End of NML\$DEFINE_KNOWN_NODES

			001C 00000	.ENTRY	NML\$DEFINE_KNOWN_NODES, Save R2,R3,R4	: 0381
	54	00000000G	00 9E 00002	MOVAB	NML\$AB_MSGBLOCK, R4	
	53	00000000'	00 9E 00009	MOVAB	NML\$Q_ENTBFDSC, R3	
	5E		0C C2 00010	SUBL2	#12, SP	
	7E		02 7D 00013	MOVQ	#2, -(SP)	: 0418
			07 DD 00016	PUSHL	#7	
00000000V	00		03 FB 00018	CALLS	#3, NML\$DEFINE_NODE	
	52		01 D0 0001F	MOVL	#1, REWIND_FLAG	: 0419
			52 DD 00022 1\$:	PUSHL	REWIND_FLAG	: 0423
		08	AE 9F 00024	PUSHAB	RECDSC	: 0420
		00000000G	00 9F 00027	PUSHAB	NML\$GQ_RECBFDSC	
			03 DD 0002D	PUSHL	#3	
00000000G	00		04 FB 0002F	CALLS	#4, NML\$READ_KNOWN_NODE_REC	
	50		50 E9 00036	BLBC	R0, 2\$	
			52 D4 00039	CLRL	REWIND_FLAG	: 0425
			64 D4 0003B	CLRL	NML\$AB_MSGBLOCK	: 0426
			02 DD 0003D	PUSHL	#2	: 0431
		08	AE 9F 0003F	PUSHAB	RECDSC	
		04	AC DD 00042	PUSHL	ENTITY	
00000000V	00		03 FB 00045	CALLS	#3, NML DEFINE_NODE	
	63	40	8F 9B 0004C	MOVZBW	#64, NML\$Q_ENTBFDSC	: 0438
04	A3	C0	A3 9E 00050	MOVAB	NML\$T_ENTBUFFER, NML\$Q_ENTBFDSC+4	: 0439
			53 DD 00055	PUSHL	R3	: 0443
			53 DD 00057	PUSHL	R3	: 0440
		04	AC DD 00059	PUSHL	ENTITY	: 0443
		10	AE 9F 0005C	PUSHAB	RECDSC	: 0440
00000000V	00		04 FB 0005F	CALLS	#4, NML\$GETRECOWNER	: 0443
	64		10 88 00066	BISB2	#16, NML\$AB_MSGBLOCK	: 0444
14	A4		63 9E 00069	MOVAB	NML\$Q_ENTBFDSC, NML\$AB_MSGBLOCK+20	: 0445
		4010	8F BB 0006D	PUSHR	#^M<R7, SP>	: 0449
00000000G	00		02 FB 00071	CALLS	#2, NML\$BLD_REPLY	
			6E DD 00078	PUSHL	MSGSIZE	: 0450
		00000000G	00 9F 0007A	PUSHAB	NML\$AB_SNDBUFFER	
00000000G	00		02 FB 00080	CALLS	#2, NML\$SEND	
			99 11 00087	BRB	1\$: 0420
			04 00089 2\$:	RET		: 0452

NML\$DEFINE
V04-000

NML_DEFINE permanent parameter module
NML\$DEFINE_KNOWN_NODES Define parameters for k

H 5
16-Sep-1984 00:12:41
14-Sep-1984 12:50:07

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLDEFINE.B32;1

Page 15
(6)

; Routine Size: 138 bytes, Routine Base: \$CODES + 01A0

NMI
VO

.....

```

: 459 0453 1 %SBTTL 'NML$DEFINE_NODE Define entity parameters'
: 460 0454 1 GLOBAL ROUTINE NML$DEFINE_NODE (ENTITY, ENTITY_LEN, ENTITY_ADR) : NOVALUE =
: 461 0455 1
: 462 0456 1 !++
: 463 0457 1 | FUNCTIONAL DESCRIPTION:
: 464 0458 1 |     Add a node's parameters to the permanent data base entry for the
: 465 0459 1 |     specified node.
: 466 0460 1
: 467 0461 1 | FORMAL PARAMETERS:
: 468 0462 1
: 469 0463 1 |     ENTITY           Entity type code.
: 470 0464 1 |     ENTITY_LEN      Byte count of entity id string.
: 471 0465 1 |     ENTITY_ADR      Address of entity id string.
: 472 0466 1
: 473 0467 1 | IMPLICIT INPUTS:
: 474 0468 1 |     NML$SL_PRS_FLGS Message parsing flags.
: 475 0469 1
: 476 0470 1 | --
: 477 0471 1
: 478 0472 2 BEGIN
: 479 0473 2
: 480 0474 2 LOCAL
: 481 0475 2     fid,           ! Node permanent database file ID
: 482 0476 2     msgsize,      ! Message size
: 483 0477 2     key,          ! Record key buffer
: 484 0478 2     key_value_dsc_addr, ! Address of node's key value descriptor
: 485 0479 2     node_type,     ! Record's node type (remote, loopnode, or
: 486 0480 2                 ! executor).
: 487 0481 2     write_type,    ! Indicates whether updating an already existing
: 488 0482 2                 ! node record, or creating a new one.
: 489 0483 2     recdsc : DESCRIPTOR, ! Record descriptor
: 490 0484 2     status;
: 491 0485 2
: 492 0486 2 |
: 493 0487 2 | Try to get the permanent database record for the node in question.
: 494 0488 2 |
: 495 0489 2 | key = .nml$ab_entitydata [.entity, eit$w_key]; ! Get search key
: 496 0490 2 | fid = .nml$ab_entitydata [.entity, eit$b_fileid]; ! Get file id
: 497 0491 2 |
: 498 0492 2 | Set up the key value.
: 499 0493 2 |
: 500 0494 2 | IF .entity EQL nml$c_executor THEN
: 501 0495 2 |     key_value_dsc_addr = UPLIT (nmn$c_typ_key_len, UPLIT (nml$c_executor))
: 502 0496 2 | ELSE
: 503 0497 2 |     key_value_dsc_addr = entity_len;
: 504 0498 2 |     status = nml$readrecord (.fid, ! Node DB file ID
: 505 0499 2 |         key, ! Key of reference ID
: 506 0500 2 |         .key_value_dsc_addr, ! Key value of node
: 507 0501 2 |         nml$gq_recbfdsc, ! Read buffer descriptor
: 508 0502 2 |         recdsc, ! Return data descriptor
: 509 0503 2 |         node_type); ! Entity type of node read
: 510 0504 2 | IF .status THEN
: 511 0505 2 |     BEGIN
: 512 0506 2 |         write_type = nmn$c_update_rec;
: 513 0507 2 |
: 514 0508 2 |         ! If this is not a loop node but the parameter group is for loop nodes
: 515 0509 2 |         ! then return a parameter not applicable error for the line parameter.

```

```

516 0510 2 | (This assumes that the line parameter is the only one applicable to
517 0511 2 | loop nodes.)
518 0512 2 |
519 0513 2 | IF .node type NEQ nml$c_loopnode AND .nml$gl_prs_flg [nml$v_prs_loopg] THEN
520 0514 2 | BEGIN
521 0515 2 |     nml$ab_msgblock [msb$l_flags] = msb$m_det_fld;
522 0516 2 |     nml$ab_msgblock [msb$b_code] = nma$c_sts_pna;
523 0517 2 |     nml$ab_msgblock [msb$w_detail] = nma$c_pcno_nli;
524 0518 2 |     status = nml$_sts_pna;
525 0519 2 | END;
526 0520 2 |
527 0521 2 | If the node is a loop node but the parameter group is not for loop nodes
528 0522 2 | then return an unrecognized component error. (There must be something
529 0523 2 | better!)
530 0524 2 |
531 0525 2 | IF .node_type EQL nml$c_loopnode AND NOT .nml$gl_prs_flg [nml$v_prs_loopg]
532 0526 2 | THEN
533 0527 2 | BEGIN
534 0528 2 |     nml$ab_msgblock [msb$l_flags] = msb$m_det_fld;
535 0529 2 |     nml$ab_msgblock [msb$b_code] = nma$c_sts_cmp;
536 0530 2 |     nml$ab_msgblock [msb$w_detail] = nma$c_ent_nod;
537 0531 2 |     status = nml$_sts_cmp;
538 0532 2 | END;
539 0533 2 | END
540 0534 2 | ELSE
541 0535 2 | BEGIN
542 0536 2 |     write_type = nmn$c_put_rec;
543 0537 2 |
544 0538 2 |     Initialize record descriptor, leaving room for the ISAM keys at the
545 0539 2 |     beginning of the record, and add entity ID field to record buffer.
546 0540 2 |
547 0541 2 |     recdsc [dsc$w_length] = 0;
548 0542 2 |     recdsc [dsc$a_pointer] = .nml$gq_recbfdsc [dsc$a_pointer] +
549 0543 2 |                               nmn$c_node_keys_len;
550 0544 2 |
551 0545 2 |     IF .entity NEQ nml$c_executor THEN
552 0546 2 |         nma$insertfld (nml$k_max_rec_data,
553 0547 2 |                       .key, .entity_len, .entity_adr, recdsc);
554 0548 2 |
555 0549 2 |     All nodes are in the same permanent database file.
556 0550 2 |     Loop nodes are uniquely identifiable because they have a circuit
557 0551 2 |     (NMA$C_PCNO_NLI) parameter and nothing else. If the NICE command
558 0552 2 |     is defining a loop node, set the entity type accordingly, so the
559 0553 2 |     loopnode will be written to the file with the correct node type key.
560 0554 2 |
561 0555 2 |     IF .nml$gl_prs_flg [nml$v_prs_loopg] THEN
562 0556 2 |         entity = nml$c_loopnode;
563 0557 2 |     status = nml$_sts_suc;
564 0558 2 |     END;
565 0559 2 |
566 0560 2 | If everything is all right then add the parameters.
567 0561 2 | IF .status THEN
568 0562 2 | BEGIN
569 0563 2 |     nml$ab_msgblock [msb$l_flags] = 0;
570 0564 2 |     nml_define_node (.entity, recdsc, .write_type);
571 0565 2 | END;
572 0566 2 |

```

```

: 573      0567 2 ! Build and send the response message.
: 574      0568 2 !
: 575      0569 2 nml$bld_reply (nml$ab_msgblock, msgsize);
: 576      0570 2 nml$send (nml$ab_sndbuffer, .msgsize);
: 577      0571 2
: 578      0572 1 END;
                                ! End of NML$DEFINE_NODE

```

.PSECT SPLITS,NOWRT,NOEXE,2

```

00000007 00000 P.AAB: .LONG 7
00000002 00004 P.AAA: .LONG 2
00000000 00008 .ADDRESS P.AAB

```

.PSECT \$CODE\$,NOWRT,2

				003C 00000	.ENTRY NML\$DEFINE_NODE, Save R2,R3,R4,R5	0454
		55	00000000G	00 9E 00002	MOVAB NML\$GL_PRS_FLGS, R5	
		54	00000000G	00 9E 00009	MOVAB NML\$AB_MSGBLOCK, R4	
		5E		14 C2 00010	SUBL2 #20, SP	
50	04	AC		2C C5 00013	MULL3 #44, ENTITY, R0	0489
			00000000G	0040 9F 00018	PUSHAB NML\$AB_ENTITYDATA+3[R0]	
	04	AE		9E 3C 0001F	MOVZWL @(SP)+, KEY	
		51	00000000G	0040 9A 00023	MOVZBL NML\$AB_ENTITYDATA[R0], FID	0490
		07	04	AC D1 0002B	CMPL ENTITY, #7	0494
				09 12 0002F	BNEQ 1\$	
		50	00000000'	00 9E 00031	MOVAB P.AAA, KEY_VALUE_DSC_ADDR	0495
				04 11 00038	BRB 2\$	
		50	08	AC 9E 0003A	MOVAB ENTITY_LEN, KEY_VALUE_DSC_ADDR	0497
				5E DD 0003E	PUSHL SP	0498
			10	AE 9F 00040	PUSHAB RECDSC	
			00000000G	00 9F 00043	PUSHAB NML\$GQ_RECBFDSC	
				50 DD 00049	PUSHL KEY_VALUE_DSC_ADDR	0500
			14	AE 9F 0004B	PUSHAB KEY	0498
				51 DD 0004E	PUSHL FID	
		00000000G	00	06 FB 00050	CALLS #6, NML\$READRECORD	
			52	50 D0 00057	MOVL R0, STATUS	
			36	52 E9 0005A	BLBC STATUS, 4\$	0504
			53	02 D0 0005D	MOVL #2, WRITE_TYPE	0506
			05	6E D1 00060	CMPL NODE_TYPE, #5	0513
				15 13 00063	BEQL 3\$	
10	01	A5		03 E1 00065	BBC #3, NML\$GL_PRS_FLGS+1, 3\$	
		64		02 D0 0006A	MOVL #2, NML\$AB_MSGBLOCK	0515
	04	A4		16 8E 0006D	MNEGB #22, NML\$AB_MSGBLOCK+4	0516
	08	A4	01F5	8F B0 00071	MOVW #501, NML\$AB_MSGBLOCK+8	0517
		52		2C CE 00077	MNEGL #44, STATUS	0518
		05		6E D1 0007A	CMPL NODE_TYPE, #5	0525
				4B 12 0007D	BNEQ 7\$	
46	01	A5		03 E0 0007F	BBS #3, NML\$GL_PRS_FLGS+1, 7\$	
		64		02 D0 00084	MOVL #2, NML\$AB_MSGBLOCK	0528
	04	A4		08 8E 00087	MNEGB #8, NML\$AB_MSGBLOCK+4	0529
			08	A4 B4 0008B	CLRW NML\$AB_MSGBLOCK+8	0530
		52		10 CE 0008E	MNEGL #16, STATUS	0531
				37 11 0C091	BRB 7\$	0504

NML\$DEFINE
V04-000

NML DEFINE permanent parameter module
NML\$DEFINE_NODE Define entity parameters

L 5
16-Sep-1984 00:12:41
14-Sep-1984 12:50:07

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLDEFINE.B32;1

Page 19
(7)

NM
VO

		53		01	D0	00093	4\$:	MOVL	#1, WRITE_TYPE	:	0536
			0C	AE	B4	00096		CLRW	RECDSC	:	0541
10	AE	00000000G	00	0A	C1	00099		ADDL3	#10, NML\$GQ_RECBFDSC+4, RECDSC+4	:	0542
			07	AC	D1	000A2		CMPL	ENTITY, #7	:	0544
				16	13	000A6		BEQL	5\$:	
				0C	AE	9F	000A8	PUSHAB	RECDSC	:	0545
		7E		08	AC	7D	000AB	MOVQ	ENTITY_LEN, -(SP)	:	0546
				10	AE	DD	000AF	PUSHL	KEY	:	
		7E	03F6	8F	3C	000B2		MOVZWL	#1014, -(SP)	:	0545
	00000000G	00		05	FB	000B7		CALLS	#5, NMA\$INSERTFLD	:	
04	01	A5		03	E1	000BE	5\$:	BBC	#3, NML\$GL_PRS_FLGS+1, 6\$:	0554
	04	AC		05	D0	000C3		MOVL	#5, ENTITY	:	0555
		52		01	D0	000C7	6\$:	MOVL	#1, STATUS	:	0556
		11		52	E9	000CA	7\$:	BLBC	STATUS, 8\$:	0561
				64	D4	000CD		CLRL	NML\$AB_MSGBLOCK	:	0563
				53	DD	000CF		PUSHL	WRITE_TYPE	:	0564
			10	AE	9F	000D1		PUSHAB	RECDSC	:	
			04	AC	DD	000D4		PUSHL	ENTITY	:	
	00000000V	00		03	FB	000D7		CALLS	#3, NML_DEFINE_NODE	:	
			08	AE	9F	000DE	8\$:	PUSHAB	MSGSIZE	:	0569
				54	DD	000E1		PUSHL	R4	:	
	00000000G	00		02	FB	000E3		CALLS	#2, NML\$BLD_REPLY	:	
			08	AE	DD	000EA		PUSHL	MSGSIZE	:	0570
			00000000G	00	9F	000ED		PUSHAB	NML\$AB_SNDBUFFER	:	
	00000000G	00		02	FB	000F3		CALLS	#2, NML\$SEND	:	
				04	000FA			RET		:	0572

; Routine Size: 251 bytes, Routine Base: \$CODE\$ + 022A

```

: 580 0573 1 %SBTTL 'NML_DEFINE_NODE Define entity parameters'
: 581 0574 1 ROUTINE NML_DEFINE_NODE (ENTITY, RECDSC, WRITE_TYPE) : NOVALUE =
: 582 0575 1
: 583 0576 1 !++
: 584 0577 1 ! FUNCTIONAL DESCRIPTION:
: 585 0578 1 ! This routine performs DEFINE NODE functions for both singular
: 586 0579 1 ! and plural requests.
: 587 0580 1
: 588 0581 1 ! FORMAL PARAMETERS:
: 589 0582 1
: 590 0583 1 ! ENTITY Entity type code.
: 591 0584 1 ! RECDSC Address of current record descriptor.
: 592 0585 1 ! WRITE_TYPE Indicates whether updating an already existing node
: 593 0586 1 ! record, or if creating a new one.
: 594 0587 1
: 595 0588 1 ! --
: 596 0589 1
: 597 0590 2 BEGIN
: 598 0591 2
: 599 0592 2 MAP
: 600 0593 2 recdsc : REF DESCRIPTOR; ! Record descriptor
: 601 0594 2
: 602 0595 2 LOCAL
: 603 0596 2 fid,
: 604 0597 2 fidsize,
: 605 0598 2 fldadr,
: 606 0599 2 node_name_dsc : VECTOR [2],
: 607 0600 2 node_type,
: 608 0601 2 original_rec : BBLOCK [nml$k_recbflen],
: 609 0602 2 original_rec_dsc : VECTOR [2],
: 610 0603 2 status;
: 611 0604 2
: 612 0605 2 !
: 613 0606 2 ! Save the original contents of the node's database record so that, if the
: 614 0607 2 ! record hasn't been modified after the changes have been applied, it isn't
: 615 0608 2 ! written back to disk. This should improve performance of command procedures
: 616 0609 2 ! that are reapplied after only a few changes (as is commonly done to update
: 617 0610 2 ! the node database).
: 618 0611 2
: 619 0612 2 original_rec_dsc [0] = .recdsc [dsc$w_length];
: 620 0613 2 original_rec_dsc [1] = original_rec;
: 621 0614 2 CH$MOVE (.recdsc [dsc$w_length], .recdsc [dsc$a_pointer], original_rec);
: 622 0615 2
: 623 0616 2 ! Add new parameter fields to the permanent database record for the node
: 624 0617 2 ! being DEFINEd.
: 625 0618 2
: 626 0619 2 IF nml$add_fields (nml$k_max_rec_data, .recdsc) THEN
: 627 0620 2 BEGIN
: 628 0621 2 !
: 629 0622 2 ! Node must have either address or line parameter specified.
: 630 0623 2 !
: 631 0624 2 status = nml$_sts_suc;
: 632 0625 2 fldadr = 0;
: 633 0626 2 IF NOT (nma$searchfld ( .recdsc,
: 634 0627 2 nma$c_pcno_nli,
: 635 0628 2 fldsize,
: 636 0629 2 fldadr)

```

```

637      OR nma$searchfld ( .recdsc,
638                          nma$c_pcno_add,
639                          fldsize,
640                          fldadr)
641      OR .entity EQL nml$c_executor) THEN
642      BEGIN
643      nml$ab_msgblock [msb$l_flags] = msb$m_det_fld;
644      nml$ab_msgblock [msb$b_code] = nma$c_sts_pms;
645      nml$ab_msgblock [msb$w_detail] = nma$c_pcno_add;
646      status = nml$sts_pms;
647      END;
648
649      !
650      ! DEFINE fields added to the node's permanent database record, check
651      ! to see if the record has changed at all. If it has, write the record
652      ! back to file.
653      !
654      IF .status THEN
655      BEGIN
656      IF CH$NEQ (.recdsc [dsc$w_length], .recdsc [dsc$a_pointer],
657                .original_rec_dsc [0], .original_rec_dsc [1], 0) THEN
658      BEGIN
659      fid = .nml$ab_entitydata [.entity, eit$b_fileid]; ! Get file id
660      status = nml$writerecord (.fid, .entity, 0, .recdsc, .write_type);
661      END;
662      IF .status EQL rms$_dup THEN
663      BEGIN
664      !
665      ! Node name is not unique. (Address was checked in
666      ! NML$DEF_NODE_ADDR since the address key to the node
667      ! database allows duplicates). So, the DEFINE cannot be
668      ! performed.
669      !
670      node_name_dsc [1] = 0;
671      !
672      ! Get the duplicated node name from node record that the NICE
673      ! command is attempting to modify.
674      !
675      nma$searchfld (.recdsc, nma$c_pcno_nna,
676                    node_name_dsc [0],
677                    node_name_dsc [1]);
678      !
679      ! Read in the record for the duplicated node so that it's ID is
680      ! returned in the NICE error response.
681      !
682      nml$readrecord (.fid,
683                    UPLIT (nma$c_pcno_nna),
684                    node_name_dsc,
685                    nml$gq_recbfdsc,
686                    .recdsc,
687                    node_type);
688      !
689      ! File ID
690      ! Key type
691      ! Key value = node name
692      ! Read buffer descriptor
693      ! Return record descriptor
694      ! Node type
695      nml$ab_msgblock [msb$l_flags] = msb$m_det_fld;
696      nml$ab_msgblock [msb$b_code] = nma$c_sts_pva;
697      nml$ab_msgblock [msb$w_detail] = nma$c_pcno_nna;
698      !
699      ! Set up the NICE message block so that the duplicate node id
700      ! is included in the NICE response message.

```



```
709 0701 1 %SBTTL 'NML$DEFKNOWLOG Define parameters for known logging entities'
710 0702 1 GLOBAL ROUTINE NML$DEFKNOWLOG (ENTITY, DUM1, DUM2,
711 0703 1 DUM3, DUM4, DUM5) : NOVALUE =
712 0704 1 !++
713 0705 1 ! FUNCTIONAL DESCRIPTION:
714 0706 1 ! This routine defines a set of parameters in the permanent
715 0707 1 ! data base entry for each logging entity.
716 0708 1 !
717 0709 1 ! FORMAL PARAMETERS:
718 0710 1 ! ENTITY Entity type code.
719 0711 1 ! DUM1-DUM5 Not used.
720 0712 1 !
721 0713 1 ! IMPLICIT INPUTS:
722 0714 1 ! NML$GL_PRS_FLGS Message parsing flags.
723 0715 1 !
724 0716 1 ! SIDE EFFECTS:
725 0717 1 ! Signals errors.
726 0718 1 !
727 0719 1 ! --
728 0720 1
729 0721 2 BEGIN
730 0722 2
731 0723 2 LOCAL
732 0724 2 fid, ! File id code
733 0725 2 key, ! Index key into logging file.
734 0726 2 exec_addr, ! Address of executor node.
735 0727 2 recdsc : DESCRIPTOR, ! Descriptor of logging database record.
736 0728 2 source_blkdsc: DESCRIPTOR, ! Descriptor for event source block.
737 0729 2 source_ptr; ! Pointer to event source block.
738 0730 2
739 0731 2
740 0732 2 fid = .nml$ab_entitydata [.entity, eit$b_fileid];
741 0733 2
742 0734 2 ! Add logging parameters only for sinks that currently exist, or
743 0735 2 ! for sinks that are logging events at the executor node.
744 0736 2
745 0737 2 INCR snk FROM nma$sc_snk_con TO nma$sc_snk_mon DO
746 0738 3 BEGIN
747 0739 3 IF .nml$gl_prs_flg [nml$pr_s_esipg] THEN
748 0740 4 BEGIN
749 0741 4 key = 0;
750 0742 4
751 0743 4 ! If the sink already has an entry in the sink (ESI) permanent
752 0744 4 ! database, modify it.
753 0745 4
754 0746 4 IF nml$matchrecord (.fid, nml$gg_recbfdsc, key,
755 0747 4 nml$sc_key_snk, 1, snk,
756 0748 4 0, 0, 0, ! No qualifier
757 0749 4 recdsc) THEN
758 0750 4 nml$deflogging (.entity, .snk, 0)
759 0751 4 ELSE
760 0752 4 !
761 0753 4 ! If the filter database (EFI) specifies events to be
762 0754 4 ! logged to this sink on the executor node, modify the
763 0755 4 ! sink (ESI) database.
764 0756 4 !
765 0757 5 BEGIN
```

```

: 766      0758 5      exec_addr = 0;
: 767      0759 5      key = 0;
: 768      0760 5
: 769      0761 5      Get the event filters for the executor node from the EFI
: 770      0762 5      database. Then search through the source blocks (which
: 771      0763 5      specify the sink type, source, if any, and filters) to see
: 772      0764 5      if any of the events are specified for the sink currently
: 773      0765 5      being processed. If so, do the define in the ESI database
: 774      0766 5      for this sink.
: 775      0767 5
: 776      0768 5      IF nml$matchrecord (.fid, nml$gq_recbfdsc, key,
: 777      0769 5      nma$c_pclo_sin, 2, exec_addr,
: 778      0770 5      0, 0, 0, ! No qualifier
: 779      0771 5      recdsc) THEN
: 780      0772 6          BEGIN
: 781      0773 6          |
: 782      0774 6          | Find the events in the permanent database record, and check
: 783      0775 6          | to see if any of them are being logged to the sink, SNK. If
: 784      0776 6          | so, update the permanent database.
: 785      0777 6          |
: 786      0778 6          source blkdsc [dsc$a_pointer] = 0;
: 787      0779 6          IF nma$searchfld (recdsc,
: 788      0780 6          nma$c_pclo_eve,
: 789      0781 6          source_blkdsc [dsc$w_length],
: 790      0782 6          source_blkdsc [dsc$a_pointer]) THEN
: 791      0783 7              BEGIN
: 792      0784 7              source_ptr = 0;
: 793      0785 7              IF nml$getnxtsnk (source_blkdsc, .snk, source_ptr) THEN
: 794      0786 7                  nml$deflogging (.entity, .snk, 0);
: 795      0787 6              END;
: 796      0788 5          END;
: 797      0789 4      END;
: 798      0790 3      END;
: 799      0791 2      END;
: 800      0792 1      END;

```

! End of NML\$DEFKOWNLOG

			001C 00000		.ENTRY NML\$DEFKOWNLOG, Save R2,R3,R4	: 0702
	54	00000000G	00 9E 00002		MOVAB NML\$MATCHRECORD, R4	
	53	00000000G	00 9E 00009		MOVAB NML\$GQ RECBFDSC, R3	
	5E		1C C2 00010		SUBL2 #28, SP	
50	04	AC	2C C5 00013		MULL3 #44, ENTITY, R0	: 0732
	52	00000000G00	40 9A 00018		MOVZBL NML\$AB_ENTITYDATA[R0], FID	
			01 DD 00020		PUSHL #1	: 0737
7A	00000000G	00	04 E1 00022	1\$:	BBC #4, NML\$GL_PRS_FLGS+1, 3\$: 0739
			08 AE D4 0002A		CLRL KEY	: 0741
			18 AE 9F 0002D		PUSHAB RECDSC	: 0746
			7E 7C 00030		CLRQ -(SP)	
			7E D4 00032		CLRL -(SP)	
			10 AE 9F 00034		PUSHAB SNK	
			01 DD 00037		PUSHL #1	
	7E		02 CE 00039		MNEGL #2, -(SP)	
			24 AE 9F 0003C		PUSHAB KEY	
			0C BB 0003F		PUSHR #*M<R2,R3>	

64		0A	FB	00041	CALLS	#10, NML\$MATCHRECORD	:	
4E		50	E8	00044	BLBS	R0, 2\$:	
	04	AE	7C	00047	CLRQ	EXEC_ADDR	: 0758	
	18	AE	9F	0004A	PUSHAB	RECDSC	: 0768	
		7E	7C	0004D	CLRQ	-(SP)	:	
		7E	D4	0004F	CLRL	-(SP)	:	
	14	AE	9F	00051	PUSHAB	EXEC_ADDR	:	
		02	DD	00054	PUSHL	#2	:	
7E	C8	8F	9A	00056	MOVZBL	#200, -(SP)	:	
	24	AE	9F	0005A	PUSHAB	KEY	:	
		0C	BB	0005D	PUSHR	#*M<R2,R3>	:	
64		0A	FB	0005F	CALLS	#10, NML\$MATCHRECORD	:	
3F		50	E9	00062	BLBC	R0, 3\$:	
	14	AE	D4	00065	CLRL	SOURCE_BLKDSC+4	: 0778	
	14	AE	9F	00068	PUSHAB	SOURCE_BLKDSC+4	: 0782	
	14	AE	9F	0006B	PUSHAB	SOURCE_BLKDSC	: 0781	
7E	C9	8F	9A	0006E	MOVZBL	#201, -(SP)	:	
	24	AE	9F	00072	PUSHAB	RECDSC	: 0779	
00000000G	00	04	FB	00075	CALLS	#4, NMA\$SEARCHFLD	: 0781	
	25	50	E9	0007C	BLBC	R0, 3\$:	
		0C	AE	D4	0007F	CLRL	SOURCE_PTR	: 0784
		0C	AE	9F	00082	PUSHAB	SOURCE_PTR	: 0785
		04	AE	DD	00085	PUSHL	SNK	:
00000000G	00	18	AE	9F	00088	PUSHAB	SOURCE_BLKDSC	:
	0F	03	FB	0008B	CALLS	#3, NML\$GETNXTSNK	:	
		50	E9	00092	BLBC	R0, 3\$:	
		7E	D4	00095	CLRL	-(SP)	: 0786	
		04	AE	DD	00097	PUSHL	SNK	:
		04	AC	DD	0009A	PUSHL	ENTITY	:
00000000V	00	03	FB	0009D	CALLS	#3, NML\$DEFLOGGING	:	
FF78	6E	03	F1	000A4	ACBL	#3, #1, SNK, 1\$: 0737	
		04	000AA	RET			: 0792	

; Routine Size: 171 bytes, Routine Base: \$CODE\$ + 0443

```

: 802 0793 1 %SBTTL 'NML$DEFLOGGING Define logging parameters'
: 803 0794 1 GLOBAL ROUTINE NML$DEFLOGGING (ENTITY, SNK, DUM2,
: 804 0795 1 DUM3, DUM4, DUM5) : NOVALUE =
: 805 0796 1
: 806 0797 1 !++
: 807 0798 1 FUNCTIONAL DESCRIPTION:
: 808 0799 1
: 809 0800 1 Add parameters to the permanent data base entry for the specified
: 810 0801 1 logging entity.
: 811 0802 1
: 812 0803 1 FORMAL PARAMETERS:
: 813 0804 1
: 814 0805 1 ENTITY Entity type code.
: 815 0806 1 SNK Logging sink type.
: 816 0807 1 DUM2-DUM5 Not used.
: 817 0808 1
: 818 0809 1 IMPLICIT INPUTS:
: 819 0810 1
: 820 0811 1 NML$GL_PRS_FLGS Message parsing flags.
: 821 0812 1 NML$GW_EVTSNKADR Sink node address.
: 822 0813 1
: 823 0814 1 !--
: 824 0815 1
: 825 0816 2 BEGIN
: 826 0817 2
: 827 0818 2 LOCAL
: 828 0819 2 sink_node_addr, ! Address of node where events get
: 829 0820 2 ! logged.
: 830 0821 2 fid, ! File id code
: 831 0822 2 msgsize, ! Message size
: 832 0823 2 key, ! Temporary record key buffer
: 833 0824 2 owner, ! Search key
: 834 0825 2 recdsc : DESCRIPTOR; ! Record descriptor
: 835 0826 2
: 836 0827 2
: 837 0828 2 ! Decide if the parameter group is for filters (EFI) or sinks (ESI).
: 838 0829 2
: 839 0830 2 IF .nml$gl_prs_flg [nml$v_prs_esipg] THEN
: 840 0831 2 BEGIN
: 841 0832 2 nml$defentity (nml$c_sink, 1, snk,
: 842 0833 2 0, 0, 0); ! No qualifier for logging.
: 843 0834 2 RETURN
: 844 0835 2 END;
: 845 0836 2
: 846 0837 2 nml$ab_msgblock [msb$l_flags] = 0; ! Initialize message flags
: 847 0838 2 fid = .nml$ab_entitydata [nml$c_logging, eit$b_fileid]; ! Get file id
: 848 0839 2 owner = .nml$ab_entitydata [nml$c_logging, eit$b_key]; ! Get search key
: 849 0840 2
: 850 0841 2 ! If the sink node (node to log the events at) is the executor node,
: 851 0842 2 store the node's address as zero in the permanent database.
: 852 0843 2
: 853 0844 2 IF .nml$gl_prs_flg [nml$v_prs_exesnk] THEN
: 854 0845 2 sink_node_addr = 0
: 855 0846 2 ELSE
: 856 0847 2 sink_node_addr = .nml$gw_evtsnkadr;
: 857 0848 2
: 858 0849 2 key = 0; ! Initialize record key

```

```

859 0850 2 |
860 0851 | If there is already a record in the event filter (EFI) database for
861 0852 | this sink node, read it in and modify it. Otherwise, create a new
862 0853 | record in the logging database for the sink node.
863 0854 |
864 0855 | IF NOT nml$matchrecord (.fid,
865 0856 |     nml$gq_recbfdsc,
866 0857 |     key,
867 0858 |     .owner, 2, sink_node_addr,
868 0859 |     0, 0, 0, ! No qualifier.
869 0860 |     recdsc) THEN
870 0861 |
871 0862 | BEGIN
872 0863 |     Record not found so create a new one by adding the record owner field.
873 0864 |     In the case of the EFI database, the sink node address is the owner
874 0865 |     field.
875 0866 |
876 0867 |     recdsc [dsc$w_length] = 0; ! Initial descriptor
877 0868 |     recdsc [dsc$a_pointer] = .nml$gq_recbfdsc [dsc$a_pointer] + 2;
878 0869 |     nml$insertfld (nml$k_max_rec_data, .owner, 2, sink_node_addr, recdsc);
879 0870 |
880 0871 |     nml$ab_msgblock [msb$l_flags] = msb$m_msg_fld;
881 0872 |     nml$ab_msgblock [msb$l_text] = nml$_recadded;
882 0873 | END;
883 0874 |
884 0875 |
885 0876 | Add the event filters to the record.
886 0877 |
887 0878 | nml_deflogging (.snk, .sink_node_addr, recdsc, key);
888 0879 |
889 0880 | Add entity id (sink type code) to entity buffer for NICE response message.
890 0881 |
891 0882 | nml$q_entbfdsc [dsc$w_length] = 1;
892 0883 | nml$q_entbfdsc [dsc$a_pointer] = nml$t_entbuffer;
893 0884 | nml$t_entbuffer<0,8> = .snk;
894 0885 |
895 0886 | Set up NICE response message information to add the entity to the NICE
896 0887 | response message.
897 0888 |
898 0889 | nml$ab_msgblock [msb$v_entd_fld] = 1;
899 0890 | nml$ab_msgblock [msb$a_entity] = nml$q_entbfdsc;
900 0891 |
901 0892 | Build and send the NICE response message.
902 0893 |
903 0894 | nml$bld_reply (nml$ab_msgblock, msgsize);
904 0895 | nml$send (nml$ab_sndbuffer, .msgsize);
905 0896 | 1 END; ! End of NML$DEFLOGGING

```

		001C 00000	.ENTRY	NML\$DEFLOGGING, Save R2,R3,R4	: 0794
54	00000000'	00 9E 00002	MOVAB	NML\$Q_ENTBFDSC, R4	:
53	00000000G	00 9E 00009	MOVAB	NML\$AB_MSGBLOCK, R3	:
5E		14 C2 00010	SUBL2	#20, SP	:
11	00000000G	00 04 E1 00013	BBC	#4, NML\$GL_PRS_FLGS+1, 1\$: 0830

			7E 7C 0001B	CLRQ	-(SP)	0832
			7E D4 0001D	CLRL	-(SP)	
		08	AC 9F 0001F	PUSHAB	SNK	
			01 DD 00022	PUSHL	#1	
			02 DD 00024	PUSHL	#2	
	FB5E	CF	06 FB 00026	CALLS	#6, NML\$DEFENTITY	
				RET		0831
			63 D4 0002C	CLRL	NML\$AB_MSGBLOCK	0837
	50	00000000G	00 9A 0002E	MOVZBL	NML\$AB_ENTITYDATA+44, FID	0838
	52	00000000G	00 3C 00035	MOVZWL	NML\$AB_ENTITYDATA+47, OWNER	0839
	04	00000000G	00 E9 0003C	BLBC	NML\$GL_PRS_FLGS+1, 2\$	0844
			6E D4 00043	CLRL	SINK_NODE_ADDR	0845
			07 11 00045	BRB	3\$	
	6E	00000000G	00 3C 00047	MOVZWL	NML\$GW_EVTSNKADR, SINK_NODE_ADDR	0847
			04 AE D4 0004E	CLRL	KEY	0849
			0C AE 9F 00051	PUSHAB	RECDSC	0855
			7E 7C 00054	CLRQ	-(SP)	
			7E D4 00056	CLRL	-(SP)	
		10	AE 9F 00058	PUSHAB	SINK_NODE_ADDR	
			02 DD 0005B	PUSHL	#2	
			52 DD 0005D	PUSHL	OWNER	0858
		20	AE 9F 0005F	PUSHAB	KEY	0855
		00000000G	00 9F 00062	PUSHAB	NML\$GQ_RECBFDS	
			50 DD 00068	PUSHL	FID	
	00000000G	00	0A FB 0006A	CALLS	#10, NML\$MATCHRECORD	
		2D	50 E8 00071	BLBS	RO, 4\$	
			0C AE B4 00074	CLRW	RECDSC	0867
	10	AE 00000000G	00	ADDL3	#2, NML\$GQ_RECBFDS+4, RECDSC+4	0868
			0C AE 9F 00080	PUSHAB	RECDSC	0869
			04 AE 9F 00083	PUSHAB	SINK_NODE_ADDR	
			02 DD 00086	PUSHL	#2	
			52 DD 00088	PUSHL	OWNER	
		7E 03F6	8F 3C 0008A	MOVZWL	#1014, -(SP)	
	00000000G	00	05 FB 0008F	CALLS	#5, NML\$INSERTFLD	
		63	04 D0 00096	MOVL	#4, NML\$AB_MSGBLOCK	0871
		0C A3 00000000G	8F D0 00099	MOVL	#NML\$_RECADED, NML\$AB_MSGBLOCK+12	0872
			04 AE 9F 000A1	PUSHAB	KEY	0878
			10 AE 9F 000A4	PUSHAB	RECDSC	
			08 AE DD 000A7	PUSHL	SINK_NODE_ADDR	
			08 AC DD 000AA	PUSHL	SNK	
	00000000V	00	04 FB 000AD	CALLS	#4, NML DEFLOGGING	
		64	01 B0 000B4	MOVW	#1, NML\$Q_ENTBFDS	0882
	04	A4 C0	A4 9E 000B7	MOVAB	NML\$T_ENTBUFFER, NML\$Q_ENTBFDS+4	0883
	C0	A4 08	AC 90 000BC	MOVW	SNK, NML\$T_ENTBUFFER	0884
		63	10 88 000C1	BISB2	#16, NML\$AB_MSGBLOCK	0889
	14	A3	64 9E 000C4	MOVAB	NML\$Q_ENTBFDS, NML\$AB_MSGBLOCK+20	0890
			08 AE 9F 000C8	PUSHAB	MSGSIZE	0894
			53 DD 000CB	PUSHL	R3	
	00000000G	00	02 FB 000CD	CALLS	#2, NML\$BLD_REPLY	
		08	AE DD 000D4	PUSHL	MSGSIZE	0895
		00000000G	00 9F 000D7	PUSHAB	NML\$AB_SNDBUFFER	
	00000000G	00	02 FB 000DD	CALLS	#2, NML\$SEND	
			04 000E4	RET		0896

; Routine Size: 229 bytes, Routine Base: \$CODE\$ + 04EE

```

: 907 0897 1 %SBTTL 'NML_DEFLOGGING Define entity parameters'
: 908 0898 1 ROUTINE NML_DEFLOGGING (SNK, SNKADR, RECDSC, KEY) : NOVALUE =
: 909 0899 1
: 910 0900 1 !++
: 911 0901 1 FUNCTIONAL DESCRIPTION:
: 912 0902 1
: 913 0903 1 This routine performs common DEFINE functions for both singular
: 914 0904 1 and plural requests.
: 915 0905 1
: 916 0906 1 FORMAL PARAMETERS:
: 917 0907 1
: 918 0908 1 SNK Logging sink type.
: 919 0909 1 SNKADR Sink node address.
: 920 0910 1 RECDSC Address of current record descriptor.
: 921 0911 1 KEY Address of current record key.
: 922 0912 1
: 923 0913 1 IMPLICIT INPUTS:
: 924 0914 1
: 925 0915 1 NONE
: 926 0916 1
: 927 0917 1 IMPLICIT OUTPUTS:
: 928 0918 1
: 929 0919 1 NONE
: 930 0920 1
: 931 0921 1 ROUTINE VALUE:
: 932 0922 1 COMPLETION CODES:
: 933 0923 1
: 934 0924 1 NONE
: 935 0925 1
: 936 0926 1 SIDE EFFECTS:
: 937 0927 1
: 938 0928 1 NONE
: 939 0929 1
: 940 0930 1 !--
: 941 0931 1
: 942 0932 2 BEGIN
: 943 0933 2
: 944 0934 2 MAP
: 945 0935 2 snkadr : WORD,
: 946 0936 2 recdsc : REF DESCRIPTOR;
: 947 0937 2
: 948 0938 2 LOCAL
: 949 0939 2 fid, ! File id code
: 950 0940 2 fldsize,
: 951 0941 2 fldadr,
: 952 0942 2 status,
: 953 0943 2 updflg;
: 954 0944 2
: 955 0945 2 fid = .nml$ab_entitydata [nml$c_logging, eit$b_fileid]; ! Get file id
: 956 0946 2
: 957 0947 2 Add event to record.
: 958 0948 2
: 959 0949 2 status = nml$addevents (true, .recdsc, .snk, .snkadr, updflg);
: 960 0950 2 IF NOT .status THEN
: 961 0951 2 RETURN;
: 962 0952 2
: 963 0953 2 IF NOT .updflg THEN

```

```

: 964      0954      BEGIN
: 965      0955      nml$ab_msgblock [msb$l_flags] = msb$m_msg_fld;
: 966      0956      nml$ab_msgblock [msb$b_code] = nma$c_sts_mpr;
: 967      0957      nml$ab_msgblock [msb$l_text] = nml$_badevtupd;
: 968      0958      RETURN
: 969      0959      END;
: 970      0960      !
: 971      0961      ! If event field was added successfully, then write the record back to file.
: 972      0962      !
: 973      0963      nml$writerecord (.fid, .snk, .key, .recdsc, 0);
: 974      0964      nml$ab_msgblock [msb$b_code] = nma$c_sts_suc;
: 975      0965      !
: 976      0966      1 END;

```

! End of NML_DEFLOGGING

		000C 00000 NML_DEFLOGGING:				
				.WORD	Save R2,R3	: 0898
	53	00000000G	00 9E 00002	MOVAB	NML\$AB_MSGBLOCK+4, R3	
	5E		04 C2 00009	SUBL2	#4, SP	
	52	00000000G	00 9A 0000C	MOVZBL	NML\$AB_ENTITYDATA+44, FID	: 0945
			5E DD 00013	PUSHL	SP	: 0949
	7E	08	AC 3C 00015	MOVZWL	SNKADR, -(SP)	
		04	AC DD 00019	PUSHL	SNK	
		0C	AC DD 0001C	PUSHL	RECDSC	
			01 DD 0001F	PUSHL	#1	
	00000000V	00	05 FB 00021	CALLS	#5, NML\$ADDEVENTS	
		2A	50 E9 00028	BLBC	STATUS, 2\$: 0950
		10	6E E8 0002B	BLBS	UPDFLG, 1\$: 0953
	FC	A3	04 D0 0002E	MOVL	#4, NML\$AB_MSGBLOCK	: 0955
		63	05 8E 00032	MNEGB	#5, NML\$AB_MSGBLOCK+4	: 0956
	08	A3 00000000G	8F D0 00035	MOVL	#NML\$_BADEVTPD, NML\$AB_MSGBLOCK+12	: 0957
			04 0003D	RET		: 0954
			7E D4 0003E 1\$:	CLRL	-(SP)	: 0963
		0C	AC DD 00040	PUSHL	RECDSC	
		10	AC DD 00043	PUSHL	KEY	
		04	AC DD 00046	PUSHL	SNK	
			52 DD 00049	PUSHL	FID	
	00000000G	00	05 FB 0004B	CALLS	#5, NML\$WRITERECORD	
		63	01 90 00052	MOVB	#1, NML\$AB_MSGBLOCK+4	: 0964
			04 00055 2\$:	RET		: 0966

; Routine Size: 86 bytes, Routine Base: \$CODE\$ + 05D3

```

: 978 0967 1 %SBTTL 'NML$ADDEVENTS Add events to the volatile data base'
: 979 0968 1 GLOBAL ROUTINE NML$ADDEVENTS (FCT, RECDSC, SNK, SNKADR, UPDFLG) =
: 980 0969 1
: 981 0970 1 |++
: 982 0971 1 | FUNCTIONAL DESCRIPTION:
: 983 0972 1 |
: 984 0973 1 |     This routine adds the filters specified in the SET command
: 985 0974 1 |     to the volatile data base entry for the specified sink node
: 986 0975 1 |     and sink type.  If an entry does not exist then a new one
: 987 0976 1 |     is created.
: 988 0977 1 |
: 989 0978 1 | FORMAL PARAMETERS:
: 990 0979 1 |
: 991 0980 1 |     FCT           Function code. (TRUE=>set, FALSE=>clear).
: 992 0981 1 |     RECDSC       Descriptor of temporary data base record
: 993 0982 1 |                 containing event filters to be added.
: 994 0983 1 |     SNK          Logging sink type.
: 995 0984 1 |     SNKADR       Sink node address.
: 996 0985 1 |     UPDFLG       Address of update flag. (TRUE=>add,
: 997 0986 1 |                 FALSE=>delete).
: 998 0987 1 |
: 999 0988 1 | IMPLICIT INPUTS:
: 1000 0989 1 |
: 1001 0990 1 |     NML$GB_EVTSRCTYP Event source type code.
: 1002 0991 1 |     NML$GQ_EVTSRCDSC Descriptor of event source id string.
: 1003 0992 1 |     NML$GQ_EVTMSKDSC Descriptor of event filter mask.
: 1004 0993 1 |     NML$GW_EVTCLASS Event class code.
: 1005 0994 1 |
: 1006 0995 1 | IMPLICIT OUTPUTS:
: 1007 0996 1 |
: 1008 0997 1 |     NONE
: 1009 0998 1 |
: 1010 0999 1 | ROUTINE VALUE:
: 1011 1000 1 | COMPLETION CODES:
: 1012 1001 1 |
: 1013 1002 1 |     NONE
: 1014 1003 1 |
: 1015 1004 1 | SIDE EFFECTS:
: 1016 1005 1 |
: 1017 1006 1 |     NONE
: 1018 1007 1 |
: 1019 1008 1 | --
: 1020 1009 1 |
: 1021 1010 2 BEGIN
: 1022 1011 2
: 1023 1012 2 MAP
: 1024 1013 2     recdsc : REF DESCRIPTOR,
: 1025 1014 2     snkadr : WORD;
: 1026 1015 2
: 1027 1016 2 OWN
: 1028 1017 2     evtbuf : VECTOR [nml$k_recbflen, BYTE];
: 1029 1018 2 BIND
: 1030 1019 2     evtdsc = UPLIT (nml$k_recbflen, evtbuf) : DESCRIPTOR;
: 1031 1020 2
: 1032 1021 2 LOCAL
: 1033 1022 2     fldadr,
: 1034 1023 2     fldsize,

```

```

: 1035      1024 2      prmdsc : DESCRIPTOR,
: 1036      1025      status;
: 1037      1026      :
: 1038      1027      : Get the event field from the record. If one does not exist then create
: 1039      1028      : a new one.
: 1040      1029      :
: 1041      1030      fldadr = 0;
: 1042      1031      status = nma$searchfld (.recdsc,
: 1043      1032      nma$c_pclo_eve,
: 1044      1033      fldsize,
: 1045      1034      fldadr);
: 1046      1035      IF .status THEN
: 1047      1036      BEGIN
: 1048      1037      CH$MOVE (.fldsize, .fldadr, .evtdsc [dsc$a_pointer]);
: 1049      1038      prmdsc [dsc$w_length] = .f[dsi];
: 1050      1039      prmdsc [dsc$a_pointer] = .evtdsc [dsc$a_pointer];
: 1051      1040      END
: 1052      1041      ELSE
: 1053      1042      BEGIN
: 1054      1043      prmdsc [dsc$w_length] = 0;
: 1055      1044      prmdsc [dsc$a_pointer] = .evtdsc [dsc$a_pointer];
: 1056      1045      END;
: 1057      1046      :
: 1058      1047      : Add the filters to those found in the temporary data base record.
: 1059      1048      :
: 1060      1049      status = nml$addfilters (.fct,
: 1061      1050      evtdsc,
: 1062      1051      .snk,
: 1063      1052      .nml$gb_evtsrctyp,
: 1064      1053      nml$gq_evtsrcdsc,
: 1065      1054      .nml$gw_evtclass,
: 1066      1055      .nml$gq_evtmskdsc [dsc$w_length],
: 1067      1056      .nml$gq_evtmskdsc [dsc$a_pointer],
: 1068      1057      prmdsc);
: 1069      1058      :
: 1070      1059      IF NOT .status THEN
: 1071      1060      BEGIN
: 1072      1061      nml$ab_msgblock [msb$l_flags] = msb$m_msg_fld;
: 1073      1062      nml$ab_msgblock [msb$b_code] = nma$c_sts_mpr;
: 1074      1063      nml$ab_msgblock [msb$l_text] = nml$_badevtupd;
: 1075      1064      RETURN .status;
: 1076      1065      END;
: 1077      1066      :
: 1078      1067      : If the filters were successfully added then replace the temporary
: 1079      1068      : data base entry with the modified one.
: 1080      1069      :
: 1081      1070      status = nml$saveevents (nml$k_max_rec data,
: 1082      1071      .prmdsc [dsc$w_length],
: 1083      1072      .prmdsc [dsc$a_pointer],
: 1084      1073      .recdsc);
: 1085      1074      IF NOT .status THEN
: 1086      1075      RETURN .status;
: 1087      1076      :
: 1088      1077      : If there are still filters remaining then indicate that they should be
: 1089      1078      : replaced in the data base.
: 1090      1079      :
: 1091      1080      2 IF .prmdsc [dsc$w_length] NEQ 0 THEN

```

```

: 1092
: 1093
: 1094
: 1095
: 1096
: 1097
: 1098
1081 2 .updflg = TRUE
1082 2 ELSE
1083 2 .updflg = FALSE;
1084 2
1085 2 RETURN nml$_sts_suc
1086 2
1087 1 END;

```

! End of NML\$ADDEVENTS

.PSECT \$SPLITS,NOWRT,NOEXE,2

```

00000400 00010 P.AAD: .LONG 1024
00000000' 00014 .ADDRESS EVTBUF

```

.PSECT \$OWNS,NOEXE,2

00048 EVTBUF: .BLKB 1024

EVTDSC= P.AAD

.PSECT \$CODE\$,NOWRT,2

```

                                01FC 00000
58 00000000G 00 9E 00002 .ENTRY NML$ADDEVENTS, Save R2,R3,R4,R5,R6,R7,R8 : 0968
5E          0C C2 00009 MOVAB NML$AB_MSGBLOCK, R8
                                7E D4 0000C SUBL2 #12, SP
                                5E DD 0000E CLRL FLDADR : 1030
                                08 AE 9F 00010 PUSHL SP : 1031
7E          C9 8F 9A 00013 PUSHAB FLDSIZE
                                08 AC DD 00017 MOVZBL #201, -(SP)
00000000G 00 04 FB 0001A CALLS #4, NMASSEARCHFLD
57          50 D0 00021 MOVL R0, STATUS
56 00000000' 00 D0 00024 MOVL EVTDC+4, R6 : 1037
66 00 BE 04 AE 28 0002E BLBC STATUS, 1$ : 1035
08 AE 04 AE B0 00034 MOVV3 FLDSIZE, @FLDADR, (R6) : 1037
                                03 11 00039 MOVW FLDSIZE, PRMDSC : 1038
                                08 AE B4 0003B 1$: BRB 2$ : 1039
0C AE          56 D0 0003E 2$: CLRW PRMDSC : 1043
                                08 AE 9F 00042 MOVL R6, PRMDSC+4 : 1044
                                00 DD 00045 PUSHAB PRMDSC : 1049
7E 00000000G 00 3C 0004B PUSHL NML$GQ_EVTMSKDSC+4 : 1056
7E 00000000G 00 3C 00052 MOVZWL NML$GQ_EVTMSKDSC, -(SP) : 1055
00000000G 00 9F 00059 MOVZWL NML$GW_EVTCLASS, -(SP) : 1054
7E 00000000G 00 9A 0005F PUSHAB NML$GQ_EVTSRCDC : 1049
                                0C AC DD 00066 MOVZBL NML$GB_EVTSRCTYP, -(SP) : 1052
                                00 9F 00069 PUSHL SNK : 1051
00000000' 04 AC DD 0006F PUSHAB EVTDC : 1049
00000000G 00 09 FB 00072 PUSHL FCT
57          50 D0 00079 CALLS #9, NML$ADDFILTERS
11          57 E8 0007C MOVL R0, STATUS
68          04 D0 0007F BLBS STATUS, 3$ : 1059
04 A8          05 8E 00082 MOVL #4, NML$AB_MSGBLOCK : 1061
0C A8 00000000G 8F D0 00086 MNEGB #5, NML$AB_MSGBLOCK+4 : 1062
                                1C 11 0008E MOVL #NML$_BADEVTPD, NML$AB_MSGBLOCK+12 : 1063
                                BRB 4$ : 1064

```

NML\$DEFINE
V04-000

NML DEFINE permanent parameter module
NML\$ADDEVENTS Add events to the volatile data

B 7
16-Sep-1984 00:12:41
14-Sep-1984 12:50:07

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLDEFINE.B32;1

Page 35
(12)

		08	AC	DD	00090	3\$:	PUSHL	RECDSC	:	1073
		10	AE	DD	00093		PUSHL	PRMDSC+4	:	1072
	7E	10	AE	3C	00096		MOVZWL	PRMDSC, -(SP)	:	1071
	7E	03F6	8F	3C	0009A		MOVZWL	#1014, -(SP)	:	1070
00000000V	00		04	FB	0009F		CALLS	#4, NML\$SAVEVENTS	:	
	57		50	D0	000A6		MOVL	R0, STATUS	:	
	04		57	E8	000A9		BLBS	STATUS, 5\$:	1074
	50		57	D0	000AC	4\$:	MOVL	STATUS, R0	:	1075
				04	000AF		RET		:	
		08	AE	B5	000B0	5\$:	TSTW	PRMDSC	:	1080
			06	13	000B3		BEQL	6\$:	
14	BC		01	D0	000B5		MOVL	#1, @UPDFLG	:	1081
			03	11	000B9		BRB	7\$:	
		14	BC	D4	000BB	6\$:	CLRL	@UPDFLG	:	1083
	50		01	D0	000BE	7\$:	MOVL	#1, R0	:	1085
				04	000C1		RET		:	1087

; Routine Size: 194 bytes, Routine Base: \$CODE\$ + 0629

```

: 1100 1088 1 %SBTTL 'NML$SAVEEVENTS Save events'
: 1101 1089 1 GLOBAL ROUTINE NML$SAVEEVENTS (BUFSIZE, LEN, ADR, RTNDSC) =
: 1102 1090 1
: 1103 1091 1 !++
: 1104 1092 1 FUNCTIONAL DESCRIPTION:
: 1105 1093 1
: 1106 1094 1 This routine stores events in a structure resembling a permanent
: 1107 1095 1 data base record. It is used for both volatile and permanent
: 1108 1096 1 event filter modifications.
: 1109 1097 1
: 1110 1098 1 FORMAL PARAMETERS:
: 1111 1099 1
: 1112 1100 1 BUFSIZE Maximum size of buffer.
: 1113 1101 1 LEN Length of events parameter.
: 1114 1102 1 ADR Address of events parameter string.
: 1115 1103 1 RTNDSC Descriptor of resulting data.
: 1116 1104 1
: 1117 1105 1 IMPLICIT INPUTS:
: 1118 1106 1
: 1119 1107 1 NONE
: 1120 1108 1
: 1121 1109 1 IMPLICIT OUTPUTS:
: 1122 1110 1
: 1123 1111 1 NONE
: 1124 1112 1
: 1125 1113 1 ROUTINE VALUE:
: 1126 1114 1 COMPLETION CODES:
: 1127 1115 1
: 1128 1116 1 NONE
: 1129 1117 1
: 1130 1118 1 SIDE EFFECTS:
: 1131 1119 1
: 1132 1120 1 NONE
: 1133 1121 1
: 1134 1122 1 --
: 1135 1123 1
: 1136 1124 2 BEGIN
: 1137 1125 2
: 1138 1126 2 MAP
: 1139 1127 2 rtdsc : REF DESCRIPTOR;
: 1140 1128 2
: 1141 1129 2 LOCAL
: 1142 1130 2 status;
: 1143 1131 2
: 1144 1132 2 status = nma$insertfld (.bufsize,
: 1145 1133 2 nma$c_pclo_eve,
: 1146 1134 2 .len,
: 1147 1135 2 .adr,
: 1148 1136 2 .rtdsc);
: 1149 1137 2 IF NOT .status THEN
: 1150 1138 2 BEGIN
: 1151 1139 2 nml$ab_msgblock [msb$l_flags] = msb$m_msg_fld;
: 1152 1140 2 nml$ab_msgblock [msb$b_code] = nma$c_sts_mpr;
: 1153 1141 2 nml$ab_msgblock [msb$l_text] = nml$_intevtovf;
: 1154 1142 2 END;
: 1155 1143 2
: 1156 1144 2 RETURN .status

```



```

: 1160      1147 1 %SBTTL 'NML$GETRECOWNER Get record owner string'
: 1161      1148 1 GLOBAL ROUTINE NML$GETRECOWNER (RECDSC, ENTITY, BUFDSC, RESLEN) =
: 1162      1149 1
: 1163      1150 1 |++
: 1164      1151 1 | FUNCTIONAL DESCRIPTION:
: 1165      1152 1 |
: 1166      1153 1 |     This routine returns the entity id string based on information supplied
: 1167      1154 1 |     by the current permanent data base record.
: 1168      1155 1 |
: 1169      1156 1 | FORMAL PARAMETERS:
: 1170      1157 1 |
: 1171      1158 1 |     RECDSC      Address of the record descriptor.
: 1172      1159 1 |     ENTITY      Entity type code.
: 1173      1160 1 |     BUFDSC      Address of owner string buffer descriptor.
: 1174      1161 1 |     RESLEN      Address of the resulting string length.
: 1175      1162 1 |
: 1176      1163 1 | IMPLICIT INPUTS:
: 1177      1164 1 |
: 1178      1165 1 |     NONE
: 1179      1166 1 |
: 1180      1167 1 | IMPLICIT OUTPUTS:
: 1181      1168 1 |
: 1182      1169 1 |     The buffer descriptor pointed to by BUFDSC describes the record
: 1183      1170 1 |     owner string.
: 1184      1171 1 |
: 1185      1172 1 | ROUTINE VALUE:
: 1186      1173 1 | COMPLETION CODES:
: 1187      1174 1 |
: 1188      1175 1 |     If no owner field is present for the record then an error
: 1189      1176 1 |     indicating invalid file contents (NML$_STS_FCO) is returned
: 1190      1177 1 |     otherwise success (NML$_STS_SUC) is returned.
: 1191      1178 1 |
: 1192      1179 1 | SIDE EFFECTS:
: 1193      1180 1 |
: 1194      1181 1 |     NONE
: 1195      1182 1 |
: 1196      1183 1 | --
: 1197      1184 1 |
: 1198      1185 2 BEGIN
: 1199      1186 2
: 1200      1187 2 MAP
: 1201      1188 2     recdsc : REF DESCRIPTOR,      ! Descriptor of record data
: 1202      1189 2     bufdsc : REF DESCRIPTOR;  ! Descriptor of entity buffer
: 1203      1190 2
: 1204      1191 2 LOCAL
: 1205      1192 2     code      : WORD,          ! Entity id parameter code
: 1206      1193 2     fldsize,    ! Temporary record owner string length
: 1207      1194 2     fldadr,     ! Temporary record owner string pointer
: 1208      1195 2     ptr,        ! Output string pointer
: 1209      1196 2     msgsize;   ! Message size
: 1210      1197 2
: 1211      1198 2 ptr = .bufdsc [dsc$a_pointer];
: 1212      1199 2
: 1213      1200 2 ! Get entity information.
: 1214      1201 2
: 1215      1202 2 code = .nml$ab_entitydata [.entity, eit$w_key]; ! Permanent database key
: 1216      1203 2

```

```

: 1217      1204 2 SELECTONEU .entity OF
: 1218      1205 2 SET
: 1219      1206 2 [nml$c_node,
: 1220      1207 2 nml$c_nodebyname,
: 1221      1208 2 nml$c_executor]:
: 1222      1209 2 BEGIN
: 1223      1210 2 fldadr = 0; ! Search all fields
: 1224      1211 2 IF nma$searchfld (.recdsc, nma$c_pcno_add,
: 1225      1212 2 fldsize,
: 1226      1213 2 fldadr) THEN
: 1227      1214 2 BEGIN
: 1228      1215 2 |
: 1229      1216 2 | If talking to a Phase III node, clear area numbers that match
: 1230      1217 2 | the executors, so they are displayed intelligibly on the Phase
: 1231      1218 2 | III system. Node numbers outside the executor's area will
: 1232      1219 2 | not look sensible, but they will be unique.
: 1233      1220 2 |
: 1234      1221 2 | IF CH$RCHAR (nml$gb_ncp_version) LEQ 3 THEN
: 1235      1222 2 | BEGIN
: 1236      1223 2 | BIND node_addr = fldadr: REF BBLOCK;
: 1237      1224 2 |
: 1238      1225 2 | IF .node_addr [nma$v_area] EQL
: 1239      1226 2 | .nml$gw_perm_exec_addr [nma$v_area] THEN
: 1240      1227 2 | node_addr [nma$v_area] = 0;
: 1241      1228 2 | END;
: 1242      1229 2 | ptr = CH$MOVE (2, .fldadr, .ptr);
: 1243      1230 2 | END
: 1244      1231 2 | ELSE
: 1245      1232 2 | BEGIN
: 1246      1233 2 | (.ptr)<0,16> = 0; ! Zero node address
: 1247      1234 2 | ptr = .ptr + 2; ! Advance pointer
: 1248      1235 2 | END;
: 1249      1236 2 | code = nma$c_pcno_nna; ! Look for name
: 1250      1237 2 | END;
: 1251      1238 2 |
: 1252      1239 2 | [nml$c_loopnode]:
: 1253      1240 2 | BEGIN
: 1254      1241 2 | code = nma$c_pcno_nna; ! Look for name
: 1255      1242 2 | (.ptr)<0,16> = 0; ! Loop node address is zero
: 1256      1243 2 | ptr = .ptr + 2; ! Advance pointer
: 1257      1244 2 | END;
: 1258      1245 2 | TES;
: 1259      1246 2 |
: 1260      1247 2 | fldadr = 0; ! Search all fields
: 1261      1248 2 | IF nma$searchfld (.recdsc, .code, fldsize, fldadr) THEN
: 1262      1249 2 | BEGIN
: 1263      1250 2 | SELECTONEU .entity OF
: 1264      1251 2 | SET
: 1265      1252 2 | [nml$c_node,
: 1266      1253 2 | nml$c_nodebyname,
: 1267      1254 2 | nml$c_loopnode,
: 1268      1255 2 | nml$c_line,
: 1269      1256 2 | nml$c_circuit,
: 1270      1257 2 | nml$c_object,
: 1271      1258 2 | nml$c_x25_serv,
: 1272      1259 2 | nml$c_trace,
: 1273      1260 2 |

```

```

: 1274      1261      3      nml$c_x29_serv]:
: 1275      1262      3      CH$WCHAR_A (.fldsize, ptr); ! Add count
: 1276      1263      3
: 1277      1264      3      [nml$c_executor]:
: 1278      1265      3      CH$WCHAR_A (.fldsize OR nma$m_ent_exe, ptr);
: 1279      1266      3
: 1280      1267      3      [nml$c_x25_access]:
: 1281      1268      4      BEGIN
: 1282      1269      4      $MOVE_ASCII ('X25-ACCESS', ptr);
: 1283      1270      4      ptr = CH$MOVE (2, UPLIT (nma$c_pcx_a_net), .ptr);
: 1284      1271      4      CH$WCHAR_A (nma$m_pty_asc, ptr);
: 1285      1272      4      CH$WCHAR_A (.fldsize, ptr); ! Add count
: 1286      1273      4      END;
: 1287      1274      4
: 1288      1275      4      [nml$c_prot_net,
: 1289      1276      3      nml$c_prot_dte,
: 1290      1277      3      nml$c_prot_grp]:
: 1291      1278      4      BEGIN
: 1292      1279      4      $MOVE_ASCII ('X25-PROTOCOL', ptr);
: 1293      1280      4      ptr = CH$MOVE (2, code, .ptr);
: 1294      1281      4      CH$WCHAR_A (nma$m_pty_asc, ptr);
: 1295      1282      4      CH$WCHAR_A (.fldsize, ptr); ! Add count
: 1296      1283      4      END;
: 1297      1284      4
: 1298      1285      4      [nml$c_x25_serv_dest,
: 1299      1286      4      nml$c_tracepnt,
: 1300      1287      4      nml$c_x29_serv_dest,
: 1301      1288      3      nml$c_ni_config]:
: 1302      1289      4      BEGIN
: 1303      1290      4      SELECTONEU .ENTITY OF
: 1304      1291      4      SET
: 1305      1292      4      [nml$c_x25_serv_dest]: $MOVE_ASCII ('X25-SERVER', ptr);
: 1306      1293      4      [nml$c_tracepnt]: $MOVE_ASCII ('X25-TRACE', ptr);
: 1307      1294      4      [nml$c_x29_serv_dest]: $MOVE_ASCII ('X29-SERVER', ptr);
: 1308      1295      4      [nml$c_ni_config]: $MOVE_ASCII ('CONFIGURATOR', ptr);
: 1309      1296      4      TES;
: 1310      1297      4      ptr = CH$MOVE (2, code, .ptr);
: 1311      1298      4      CH$WCHAR_A (nma$m_pty_asc, ptr);
: 1312      1299      4      CH$WCHAR_A (.fldsize, ptr); ! Add count
: 1313      1300      4      END;
: 1314      1301      4      TES;
: 1315      1302      4
: 1316      1303      4      |
: 1317      1304      4      | If it will fit, move the entity string into the buffer for the
: 1318      1305      4      | record owner. The contents of this buffer are later used as
: 1319      1306      4      | the entity ID in the NICE response message.
: 1320      1307      4      |
: 1321      1308      4      IF .bufdsc [dsc$w_length] GEQU
: 1322      1309      4      (.fldsize + (.ptr - .bufdsc [dsc$a_pointer])) THEN
: 1323      1310      4      ptr = CH$MOVE (.fldsize, .fldadr, .ptr);
: 1324      1311      4
: 1325      1312      4      END
: 1326      1313      4      ELSE
: 1327      1314      4      SELECTONEU .entity OF
: 1328      1315      4      SET
: 1329      1316      4      [nml$c_node,
: 1330      1317      4      nml$c_nodebyname]:

```

```

: 1331      1318      2          CH$WCHAR_A (0, ptr);
: 1332      1319      2
: 1333      1320      2          [nml$c_executor]:
: 1334      1321      2          ch$wchar_A (nma$m_ent_exe, ptr);
: 1335      1322      2
: 1336      1323      2          [OTHERWISE] :
: 1337      1324      2          BEGIN
: 1338      1325      2          nml$ab_msgblock [msb$l_flags] =
: 1339      1326      2          msb$m_det_fld OR msb$m_msg_fld; ! Set message flags
: 1340      1327      2          nml$ab_msgblock [msb$b_code] = nma$c_sts_fco;
: 1341      1328      2          nml$ab_msgblock [msb$w_detail] = nma$c_fopdtl_pdb;
: 1342      1329      2          nml$ab_msgblock [msb$l_text] = nml$norecown;
: 1343      1330      2          nml$bl_d_reply (nml$ab_msgblock, msgsize); ! Build message
: 1344      1331      2          $SIGNAL_MSG (nml$ab_sndbuffer, .msgsize); ! Signal it
: 1345      1332      2          END;
: 1346      1333      2          TES;
: 1347      1334      2
: 1348      1335      2          .reslen = .ptr - .bufdsc [dsc$a_pointer];
: 1349      1336      2          RETURN nml$_sts_suc
: 1350      1337      1          END;
! End of NML$GETRECOWNER

```

												.PSECT		SPLITS, NOWRT, NOEXE, 2											
		53	53	45	43	43	41	2D	35	32	58	0A	00018	P.AAE:	.ASCII	<10>\X25-ACCESS\	:								
													00023		.BLKB	1	:								
													00000456	P.AAF:	.LONG	1110	:								
4C	4F	43	4F	54	4F	52	50	2D	35	32	58	0C	00028	P.AAG:	.ASCII	<12>\X25-PROTOCOL\	:								
		52	45	56	52	45	53	2D	35	32	58	0A	00035	P.AAH:	.ASCII	<10>\X25-SERVER\	:								
			45	43	41	52	54	2D	35	32	58	09	00040	P.AAI:	.ASCII	<9>\X25-TRACE\	:								
		52	45	56	52	45	53	2D	39	32	58	0A	0004A	P.AAJ:	.ASCII	<10>\X29-SERVER\	:								
52	4F	54	41	52	55	47	49	46	4E	4F	43	0C	00055	P.AAK:	.ASCII	<12>\CONFIGURATOR\	:								
												.PSECT		\$CODE\$, NOWRT, 2											
														OFFC 00000											
												.ENTRY		NML\$GETRECOWNER, Save R2,R3,R4,R5,R6,R7,R8,-;		1148									
												5B		00000000G		00 9E 00002		MOVAB		NMA\$SEARCHFLD, R11					
												5A		00000000G		00 9E 00009		MOVAB		NML\$AB_MSGBLOCK, R10					
												59		00000000'		00 9E 00010		MOVAB		P.AAE, R9					
												5E				0C C2 00017		SUBL2		#12, SP					
												56		08		AC 7D 0001A		MOVQ		ENTITY, R6		1202			
												53		04		A7 D0 0001E		MOVL		4(R7), PTR		1198			
												56				2C C5 00022		MULL3		#44, R6, R0		1202			
												58		00000000G00		40 9F 00026		PUSHAB		NML\$AB_ENTITYDATA+3[R0]					
												58				9E B0 0002D		MOVW		@ (SP)+, CODE					
												03				56 D1 00030		CPL		R6, #3		1206			
												04				05 1F 00033		BLSSU		1\$					
												04				56 D1 00035		CPL		R6, #4					
												07				05 1B 00038		BLEQU		2\$					
																56 D1 0003A		1\$:		CPL		R6, #7			
																47 12 0003D		2\$:		BNEQ		6\$			
																6E D4 0003F				CLRL		FLDADR		1210	
																5E DD 00041				PUSHL		SP		1211	

			08	AE	9F	00043		PUSHAB	FLDSIZE		
		7E	01F6	8F	3C	00046		MOVZWL	#502, -(SP)		
			04	AC	DD	00048		PUSHL	RECD\$C		
		6B		04	FB	0004E		CALLS	#4, NMASSEARCHFLD		
		26		50	E9	00051		BLBC	R0, 4\$		
		03	00000000G	00	91	00054		CMPB	NML\$GB_NCP_VERSION, #3		1222
				17	1A	0005B		BGTRU	3\$		
50	00000000G	00		02	EF	0005D		EXTZV	#2, #6, NML\$GW_PERM_EXEC_ADDR+1, R0		1227
50		00	BE	0A	ED	00066		CMPZV	#10, #6, @NODE_ADDR, R0		
				06	12	0006C		BNEQ	3\$		
		00	BE	8F	AA	0006E		BICW2	#64512, @NODE_ADDR		1228
				63	BE	00074	3\$:	MOVW	@FLDADR, (PTR)		1230
					02	11	00078	BRB	5\$		
					63	B4	0007A	4\$:	CLRW	(PTR)	1234
		53		02	C0	0007C	5\$:	ADDL2	#2, PTR		1235
		58	01F4	8F	B0	0007F		MOVW	#500, CODE		1237
				0C	11	00084		BRB	7\$		1204
		05		56	D1	00086	6\$:	CMPL	R6, #5		1240
				07	12	00089		BNEQ	7\$		
		58	01F4	8F	B0	0008B		MOVW	#500, CODE		1242
				83	B4	00090		CLRW	(PTR)+		1243
				6E	D4	00092	7\$:	CLRL	FLDADR		1248
				5E	DD	00094		PUSHL	SP		1249
			08	AE	9F	00096		PUSHAB	FLDSIZE		
		7E		58	3C	00099		MOVZWL	CODE, -(SP)		
			04	AC	DD	0009C		PUSHL	RECD\$C		
		6B		04	FB	0009F		CALLS	#4, NMASSEARCHFLD		
		03		50	E8	000A2		BLBS	R0, 8\$		
				00C1	31	000A5		BRW	26\$		
				56	D5	000A8	8\$:	TSTL	R6		1253
				24	13	000AA		BEQL	13\$		
		03		56	D1	000AC		CMPL	R6, #3		
				05	1F	000AF		BLSSU	9\$		
		05		56	D1	000B1		CMPL	R6, #5		
				08	1B	000B4		BLEQU	10\$		
		08		56	D1	000B6	9\$:	CMPL	R6, #8		
				08	1F	000B9		BLSSU	12\$		
		09		56	D1	000BB		CMPL	R6, #9		
				03	1A	000BE	10\$:	BGTRU	12\$		
				0088	31	000C0	11\$:	BRW	23\$		
		11		56	D1	000C3	12\$:	CMPL	R6, #17		
				F8	13	000C6		BEQL	11\$		
		13		56	D1	000C8		CMPL	R6, #19		
				7E	13	000CB		BEQL	23\$		
		15		56	D1	000CD		CMPL	R6, #21		
				79	13	000D0	13\$:	BEQL	23\$		
		07		56	D1	000D2		CMPL	R6, #7		1264
				08	12	000D5		BNEQ	14\$		
	63	04	AE	80	8F	89	000D7	BISB3	#128, FLDSIZE, (PTR)		1265
				70	11	000DD		BRB	24\$		
			0D	56	D1	000DF	14\$:	CMPL	R6, #13		1267
				0A	12	000E2		BNEQ	15\$		
	63		69	0B	28	000E4		MOV3	#11, P.AAE, (PTR)		1269
			63	0C	A9	B0	000E8	MOVW	P.AAF, (PTR)		1270
				56	11	000EC		BRB	22\$		
			0E	56	D1	000EE	15\$:	CMPL	R6, #14		1275
				0C	1F	000F1		BLSSU	16\$		

		10		56	D1	000F3			CMPL	R6	#16			
				07	1A	000F6			BGTRU	16\$				
63	10	A9		0D	28	000F8			MOV C3	#13,	P.AAG,	(PTR)	1279	
				42	11	000FD			BRB	21\$			1280	
		12		56	D1	000FF	16\$:		CMPL	R6	#18		1285	
				0F	13	00102			BEQL	17\$				
		14		56	D1	00104			CMPL	R6	#20			
				0A	13	00107			BEQL	17\$				
		16		56	D1	00109			CMPL	R6	#22			
				43	1F	0010C			BLSSU	25\$				
		17		56	D1	0010E			CMPL	R6	#23			
				3E	1A	00111			BGTRU	25\$				
		12		56	D1	00113	17\$:		CMPL	R6	#18		1292	
63	1D	A9		07	12	00116			BNEQ	18\$				
				0B	28	00118			MOV C3	#11,	P.AAH,	(PTR)		
		14		22	11	0011D			BRB	21\$				
				56	D1	0011F	18\$:		CMPL	R6	#20		1293	
63	28	A9		07	12	00122			BNEQ	19\$				
				0A	28	00124			MOV C3	#10,	P.AAI,	(PTR)		
		16		16	11	00129			BRB	21\$				
				56	D1	0012B	19\$:		CMPL	R6	#22		1294	
63	32	A9		07	12	0012E			BNEQ	20\$				
				0B	28	00130			MOV C3	#11,	P.AAJ,	(PTR)		
		17		0A	11	00135			BRB	21\$				
				56	D1	00137	20\$:		CMPL	R6	#23		1295	
63	3D	A9		05	12	0013A			BNEQ	21\$				
				0D	28	0013C			MOV C3	#13,	P.AAK,	(PTR)		
		63		58	B0	00141	21\$:		MOVW	CODE,	(PTR)		1297	
		53		02	C0	00144	22\$:		ADDL2	#2,	PTR			
		83	40	8F	90	00147			MOV B	#64,	(PTR)+		1298	
		63	04	AE	90	0014B	23\$:		MOV B	FLDSIZE,	(PTR)		1299	
				53	D6	0014F	24\$:		INCL	PTR				
50	53		04	A7	C3	00151	25\$:		SUBL3	4(R7),	PTR, R0		1309	
	50		04	AE	C0	00156			ADDL2	FLDSIZE,	R0			
50	67	10		00	ED	0015A			CMPZV	#0,	#16,	(R7), R0		
				57	1F	0015F			BLSSU	30\$				
63	00	BE	04	AE	28	00161			MOV C3	FLDSIZE,	@FLDADR,	(PTR)	1310	
				4F	11	00167			BRB	30\$			1249	
		03		56	D1	00169	26\$:		CMPL	R6	#3		1316	
				09	1F	0016C			BLSSU	27\$				
		04		56	D1	0016E			CMPL	R6	#4			
				04	1A	00171			BGTRU	27\$				
				63	94	00173			CLRB	(PTR)			1318	
				09	11	00175			BRB	28\$				
		07		56	D1	00177	27\$:		CMPL	R6	#7		1320	
				08	12	0017A			BNEQ	29\$				
		63	80	8F	90	0017C			MOV B	#-128,	(PTR)		1321	
				53	D6	00180	28\$:		INCL	PTR				
				34	11	00182			BRB	30\$			1314	
		6A		06	D0	00184	29\$:		MOVL	#6,	NML\$AB	MSGBLOCK	1326	
	04	AA		0E	8E	00187			MNEGB	#14,	NML\$AB	MSGBLOCK+4	1327	
			08	AA	B4	0018B			CLRW	NML\$AB	MSGBLOCK+8		1328	
	0C	AA	00000000G	8F	D0	0018E			MOVL	#NML\$	NORECOWN,	NML\$AB	MSGBLOCK+12	1329
				08	AE	0F	00194		PUSHAB	MSGSIZE			1330	
				5A	DD	00199			PUSHL	R10				
00000000G	00			02	FB	0019B			CALLS	#2,	NML\$BLD	_REPLY		
			08	AE	DD	001A2			PUSHL	MSGSIZE			1331	

NML\$DEFINE
V04-000

NML DEFINE permanent parameter module
NML\$GETREOWNER Get record owner string

K 7
16-Sep-1984 00:12:41
14-Sep-1984 12:50:07

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLDEFINE.B32;1

Page 44
(14)

		00000000G	00	9F	001A5	
		01F90000	8F	DD	001AB	
10	BC	00000000G	00	03	FB	001B1
			53	A7	C3	001B8
			50	01	D0	001BE
				04	001C1	

PUSHAB	NML\$AB	SNDBUFFER
PUSHL	#33095680	
CALLS	#3	LIB\$SIGNAL
SUBL3	4(R7)	PTR, @RESLEN
MOVL	#1,	R0
RET		

.....
: 1335
: 1336
: 1337

; Routine Size: 450 bytes, Routine Base: \$CODE\$ + 071C

NML
V04

NML\$DEFINE
V04-000

NML DEFINE permanent parameter module
NML\$GETREOWNER Get record owner string

L 7
16-Sep-1984 00:12:41
14-Sep-1984 12:50:07

VAX-11 Bliss-32 V4.0-742
[NML.SRC]NMLDEFINE.B32;1

Page 45
(15)

: 1352
: 1353
: 1354

1338 1 END
1339 1
1340 0 ELUDOM

! End of module

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	1096	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	2270	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	98	NOVEC, NOWRT, RD, NOEXE, 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	68	19	27	00:00.1
-\$255\$DUA28:[SHRLIB]NMALIBRY.L32;1	887	20	2	47	00:00.2
-\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	3	0	581	00:02.2

COMMAND QUALIFIERS

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

: Size: 2270 code + 1194 data bytes
: Run Time: 00:37.3
: Elapsed Time: 01:18.9
: Lines/CPU Min: 2157
: Lexemes/CPU-Min: 13330
: Memory Used: 201 pages
: Compilation Complete

