

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

_S
Ps
NP
NP
SG
SOI
NP
PA
_L

```

NN      NN      MM      MM      AAAAAA      FFFFFFFF      IIIIII      EEEEEEEEE      LL      DDDDDDD      SSSSSSS
NN      NN      MM      MM      AAAAAA      FFFFFFFF      IIIIII      EEEEEEEEE      LL      DDDDDDD      SSSSSSS
NN      NN      MMMM     MMMM     AA      AA      FF      FF      III      EE      LL      DD      DD      SS
NN      NN      MMMM     MMMM     AA      AA      FF      FF      III      EE      LL      DD      DD      SS
NNNN     NN      MM      MM      AA      AA      FF      FF      III      EE      LL      DD      DD      SS
NNNN     NN      MM      MM      AA      AA      FF      FF      III      EE      LL      DD      DD      SS
NN      NN      NN      MM      MM      AA      AA      FFFFFFFF      III      EEEEEEE      LL      DD      DD      SSSSS
NN      NN      NN      MM      MM      AA      AA      FFFFFFFF      III      EEEEEEE      LL      DD      DD      SSSSS
NN      NNNN     MM      MM      AAAAAAAAAA      FF      FF      III      EE      LL      DD      DD      SS
NN      NNNN     MM      MM      AAAAAAAAAA      FF      FF      III      EE      LL      DD      DD      SS
NN      NN      MM      MM      AA      AA      FF      FF      III      EE      LL      DD      DD      SS
NN      NN      MM      MM      AA      AA      FF      FF      III      EE      LL      DD      DD      SS
NN      NN      MM      MM      AA      AA      FF      FF      IIIIII     EEEEEEEEE      LLLLLLL'LL      DDDDDDD      SSSSSSS
NN      NN      MM      MM      AA      AA      FF      FF      IIIIII     EEEEEEEEE      LLLLLLLLLL      DDDDDDD      SSSSSSS

```

```

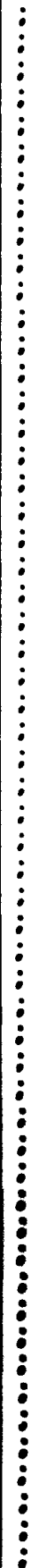
LL      IIIIII     SSSSSSS
LL      IIIIII     SSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSS
LL      II      SSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII     SSSSSSS
LLLLLLLLLL      IIIIII     SSSSSSS

```

```

....
....
....
....

```



```

1 0001 0 %TITLE ' Field Support Routines'
2 0002 0 MODULE NMAFIELDS (
3 0003 0 LANGUAGE (BLISS32),
4 0004 0 ADDRESSING_MODE (NONEXTERNAL=LONG_RELATIVE),
5 0005 0 ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE),
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 *
13 0013 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
14 0014 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
15 0015 1 * ALL RIGHTS RESERVED.
16 0016 1 *
17 0017 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
18 0018 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
19 0019 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
20 0020 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
21 0021 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
22 0022 1 * TRANSFERRED.
23 0023 1 *
24 0024 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
25 0025 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
26 0026 1 * CORPORATION.
27 0027 1 *
28 0028 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
29 0029 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
30 0030 1 *
31 0031 1 *
32 0032 1 *****
33 0033 1
34 0034 1
35 0035 1 ++
36 0036 1 FACILITY: Network Management Layer (NMA)
37 0037 1
38 0038 1 ABSTRACT:
39 0039 1
40 0040 1 These routines provide support for maintaining permanent
41 0041 1 data bases in NML and other network management layer components.
42 0042 1
43 0043 1 ENVIRONMENT: VAX/VMS Operating System
44 0044 1
45 0045 1 AUTHOR: Darrell Duffy , CREATION DATE: 15-October-1979
46 0046 1
47 0047 1 MODIFIED BY:
48 0048 1
49 0049 1 V03-001 MKP0001 Kathy Perko 6-Aug-1983
50 0050 1 Remove all knowledge of key sizes from this module.
51 0051 1 This is necessary because the keys for the new node
52 0052 1 database records are different than for the other files.
53 0053 1 Also, modify field insertion to only insert a field if
54 0054 1 the parameter value has changed.
55 0055 1 --

```

```
57 0056 1 %SBTTL 'Definitions'
58 0057 1
59 0058 1
60 0059 1 !! TABLE OF CONTENTS:
61 0060 1 !!
62 0061 1
63 0062 1 FORWARD ROUTINE
64 0063 1     NMA$SEARCHFLD,
65 0064 1     NMA$INSERTFLD,
66 0065 1     NMA$DELETEFLD;
67 0066 1
68 0067 1 !!
69 0068 1 !! INCLUDE FILES:
70 0069 1 !!
71 0070 1
72 0071 1 LIBRARY 'SHRLIBS:NMALIBRY.L32';
73 0072 1 LIBRARY 'SYSSLIBRARY:STARLET.L32';
74 0073 1
75 0074 1 !!
76 0075 1 !!           Fields for handling descriptors
77 0076 1 !!
78 0077 1
79 0078 1     FIELD
80 0079 1     DSC_FLDS =
81 0080 1     SET
82 0081 1     DSC_SIZ = [0, 0, 16, 0],
83 0082 1     DSC_ADR = [4, 0, 32, 0],
84 0083 1     TES
85 0084 1     ;
86 0085 1
87 0086 1 !!
88 0087 1 !! EQUATED SYMBOLS:
89 0088 1 !!
90 0089 1 !!
91 0090 1 !!
92 0091 1 !! OWN STORAGE:
93 0092 1 !!
94 0093 1 !!
95 0094 1 !!
96 0095 1 !! EXTERNAL REFERENCES:
97 0096 1 !!
98 0097 1 !!
99 0098 1 !! EXTERNAL ROUTINE
100 0099 1 !! ;
```

```

102 0100 1 %SBTTL 'NMASEARCHFLD Find Field by Code'
103 0101 1 GLOBAL ROUTINE NMASEARCHFLD (RTN_DSC, FIELD_CODE,
104 0102 1 FIELD_SIZE, FIELD_ADR) = !
105 0103 1
106 0104 1
107 0105 1 ++
108 0106 1 FUNCTIONAL DESCRIPTION:
109 0107 1 Return size and address of data portion of a field with a specified
110 0108 1 code.
111 0109 1
112 0110 1 FORMAL PARAMETERS:
113 0111 1
114 0112 1 RTN_DSC Address of descriptor of data portion of record
115 0113 1 FIELD_CODE Value of word that is the code for the field
116 0114 1 FIELD_SIZE Address of size of field data to be returned.
117 0115 1 Returned as a longword.
118 0116 1 FIELD_ADR Address of longword to hold address of data.
119 0117 1 On input may point within the data record as
120 0118 1 the start of the next field to compare.
121 0119 1 After a call here,
122 0120 1 FIELD_ADR = .FIELD_ADR + .FIELD_SIZE
123 0121 1 sets up for the next call here to obtain next field.
124 0122 1
125 0123 1 IMPLICIT INPUTS:
126 0124 1
127 0125 1 NONE
128 0126 1
129 0127 1 IMPLICIT OUTPUTS:
130 0128 1
131 0129 1 NONE
132 0130 1
133 0131 1 ROUTINE VALUE:
134 0132 1 COMPLETION CODES:
135 0133 1
136 0134 1 NMA$_SUCCESS Field found
137 0135 1 NMA$_FLDNOTFND Field with specified code not found or
138 0136 1 no more fields in record.
139 0137 1
140 0138 1 SIDE EFFECTS:
141 0139 1
142 0140 1 NONE
143 0141 1
144 0142 1 --
145 0143 1
146 0144 1 BEGIN
147 0145 1
148 0146 1 MAP
149 0147 1 FIELD_CODE : WORD,
150 0148 1 RTN_DSC : REF BLOCK [8, 1]
151 0149 1 FIELD (DSC_FLDS)
152 0150 1 ;
153 0151 1
154 0152 1 LOCAL
155 0153 1 PTR, ! Pointer into record
156 0154 1 END_REC, ! End of record data
157 0155 1 CODE : WORD, ! Code value
158 0156 1 SIZE ! Size of a field

```

```

: 159 0157 2
: 160 0158 2
: 161 0159 2 PTR = ..FIELD_ADR; ! Obtain the address of the field
: 162 0160 2
: 163 0161 2 ! We add two to the descriptor address
: 164 0162 2 ! To skip the two byte key at the
: 165 0163 2 ! beginning of the record
: 166 0164 2
: 167 0165 2 IF (.PTR LSSA (.RTN_DSC [DSC_ADR]) ) ! If its a good address, use it
: 168 0166 2 OR ! and look on from there
: 169 0167 4 (.PTR GTRA (.RTN_DSC [DSC_ADR] +
: 170 0168 4 .RTN_DSC [DSC_SIZE]
: 171 0169 4 )
: 172 0170 2 )
: 173 0171 2 THEN
: 174 0172 2 PTR = .RTN_DSC [DSC_ADR]; ! if its bad, use start of record
: 175 0173 2
: 176 0174 2 END_REC = .RTN_DSC [DSC_ADR] + ! compute end of the record
: 177 0175 2 .RTN_DSC [DSC_SIZE] ;
: 178 0176 2
: 179 0177 2 WHILE .PTR LSSA .END_REC ! Until there is no more data
: 180 0178 2 DO
: 181 0179 2 BEGIN
: 182 0180 2 CODE = . (.PTR) <0, 16, 0> ; ! Look at the field code
: 183 0181 2 SIZE = . (.PTR) <16, 16, 0> ; ! And the size of the field
: 184 0182 2 IF (.CODE EQLU .FIELD_CODE) ! If the codes match, or
: 185 0183 2 THEN
: 186 0184 2 BEGIN
: 187 0185 2 .FIELD_ADR = .PTR + 4; ! Return position of data
: 188 0186 2 .FIELD_SIZE = .SIZE; ! and size
: 189 0187 2 RETURN NMA$_SUCCESS ! And a warm, fuzzy code
: 190 0188 2 END
: 191 0189 2 ELSE
: 192 0190 2 PTR = .PTR + .SIZE + 4 ! Otherwise, keep looking
: 193 0191 2 END
: 194 0192 2 ;
: 195 0193 2
: 196 0194 2 RETURN NMA$_FLDNOTFND ! A not so warm fuzzy code
: 197 0195 2
: 198 0196 1 END;

```

.TITLE NMAFIELDS Field Support Routines
.IDENT \V04-000\

.PSECT \$CODE\$,NOWRT,2

			000C 00000
	51	10	BC D0 00002
	50	04	AC D0 00006
04	A0		51 D1 0000A
			0C 1F 0000E
	52		60 3C 00010
	52	04	A0 C0 00013
	52		51 D1 00017

```

.ENTRY NMA$SEARCHFLD, Save R2,R3
MOVL @FIELD_ADR, PTR
MOVL RTN_DSC, R0
CML PTR, 4(R0)
BLSSU 1$
MOVZWL (R0), R2
ADDL2 4(R0), R2
CML PTR, R2

```

```

: 0101
: 0159
: 0165
:
: 0168
:
: 0167

```

	51	04	04	1B	0001A		BLEQU	2\$:	0172
	53		A0	D0	0001C	1\$:	MOVL	4(R0), PTR		:	0175
	53	04	60	3C	00020	2\$:	MOVZWL	(R0), END_REC		:	
	53		A0	C0	00023		ADDL2	4(R0), END_REC		:	
			51	D1	00027	3\$:	CMPL	PTR, END_REC		:	0177
	52		21	1E	0002A		BGEQU	5\$:	
	50		61	B0	0002C		MOVW	(PTR), CODE		:	0180
08	AC	02	A1	3C	0002F		MOVZWL	2(PTR), SIZE		:	0181
			52	B1	00033		CMPW	CODE, FIELD_CODE		:	0182
			0D	12	00037		BNEQ	4\$:	
10	BC	04	A1	9E	00039		MOVAB	4(R1), @FIELD_ADR		:	0185
0C	BC		50	D0	0003E		MOVL	SIZE, @FIELD_SIZE		:	0186
	50		01	D0	00042		MOVL	#1, R0		:	0187
			04	04	00045		RET			:	
	51	04	A041	9E	00046	4\$:	MOVAB	4(SIZE)[PTR], PTR		:	0190
			DA	11	0004B		BRB	3\$:	0182
	50		20	D0	0004D	5\$:	MOVL	#32, R0		:	0194
			04	04	00050		RET			:	0196

; Routine Size: 81 bytes, Routine Base: \$CODE\$ + 0000

```

: 200 0197 1 %SBTTL 'NMA$INSERTFLD Insert a Field Into a Record'
: 201 0198 1 GLOBAL ROUTINE NMA$INSERTFLD (BUF_SIZE, FIELD_CODE, FIELD_SIZE,
: 202 0199 1 FIELD_ADR, RTN_DSC) =
: 203 0200 1
: 204 0201 1 !++
: 205 0202 1 ! FUNCTIONAL DESCRIPTION:
: 206 0203 1
: 207 0204 1 ! Insert a field into a record. All other fields by that code are
: 208 0205 1 ! removed from the record first.
: 209 0206 1
: 210 0207 1 ! FORMAL PARAMETERS:
: 211 0208 1
: 212 0209 1 !     BUF_SIZE      Maximum size of parameter buffer
: 213 0210 1 !     FIELD_CODE   Value of the word containing the code
: 214 0211 1 !     FIELD_SIZE   Size of field to be inserted
: 215 0212 1 !     FIELD_ADR   Address of field to be inserted
: 216 0213 1 !     RTN_DSC     Descriptor of data in buffer.
: 217 0214 1
: 218 0215 1 ! IMPLICIT INPUTS:
: 219 0216 1
: 220 0217 1 !     NONE
: 221 0218 1
: 222 0219 1 ! IMPLICIT OUTPUTS:
: 223 0220 1
: 224 0221 1 !     NONE
: 225 0222 1
: 226 0223 1 ! ROUTINE VALUE:
: 227 0224 1 ! COMPLETION CODES:
: 228 0225 1
: 229 0226 1 !     NMA$_SUCCESS      A warm cuddly feeling
: 230 0227 1 !     NMA$_SUCCFLDRPL  A warm feeling with a replaced field
: 231 0228 1 !     NMA$_BUFTOOSMALL Not enough space in the buffer, buffer is
: 232 0229 1 !                       not corrupted, and the field is not inserted.
: 233 0230 1
: 234 0231 1 ! SIDE EFFECTS:
: 235 0232 1
: 236 0233 1 !     NONE
: 237 0234 1
: 238 0235 1 ! --
: 239 0236 1
: 240 0237 2 ! BEGIN
: 241 0238 2
: 242 0239 2 ! MAP
: 243 0240 2 !     RTN_DSC : REF BLOCK [8, 1]
: 244 0241 2 !             FIELD (DSC_FLDS)
: 245 0242 2 !     ;
: 246 0243 2
: 247 0244 2 ! LOCAL
: 248 0245 2 !     SUCCOD,      ! Hold the success code for return
: 249 0246 2 !     SIZE,
: 250 0247 2 !     ADR,
: 251 0248 2 !     END_FLD     ! End of a field to replace
: 252 0249 2 !     ;
: 253 0250 2
: 254 0251 2 ! ++
: 255 0252 2 ! If the contents of the field haven't changed, just return success.
: 256 0253 2 ! --

```



```

: 257 0254 2 ADR = 0;
: 258 0255 2 SUCCOD = NMA$SEARCHFLD (.RTN_DSC, .FIELD_CODE, SIZE, ADR);
: 259 0256 2 IF .SUCCOD AND
: 260 0257 2 CH$EQL (.FIELD_SIZE, .FIELD_ADR, .SIZE, .ADR, 0) THEN
: 261 0258 2 RETURN .SUCCOD;
: 262 0259 2 !+
: 263 0260 2 Delete existing fields with specified code.
: 264 0261 2 !-
: 265 0262 2 SUCCOD = NMA$DELETEFLD (.RTN_DSC, .FIELD_CODE);
: 266 0263 2 !* END_FLD = .RTN_DSC [DSC_ADR] + ! End of data in record
: 267 0264 2 !* .RTN_DSC [DSC_SIZE];
: 268 0265 2 END_FLD = .RTN_DSC [DSC_ADR]; ! End of data in record
: 269 0266 2 END_FLD = .END_FLD + .RTN_DSC [DSC_SIZE];
: 270 0267 2
: 271 0268 3 IF (.FIELD_SIZE + 4) ! Do we have enough room
: 272 0269 2 GTRU
: 273 0270 3 (.BUF_SIZE - .RTN_DSC [DSC_SIZE])
: 274 0271 2 THEN
: 275 0272 2 RETURN NMA$_BUFTOOSMALL ! Nope, return error
: 276 0273 2 ;
: 277 0274 2
: 278 0275 2 (.END_FLD) <0, 16, 0> = .FIELD_CODE; ! Store control data
: 279 0276 2 (.END_FLD) <16, 16, 0> = .FIELD_SIZE;
: 280 0277 2 END_FLD = .END_FLD + 4;
: 281 0278 2
: 282 0279 2 RTN_DSC [DSC_SIZE] = ! Adjust descriptor for size
: 283 0280 2 CH$MOVE (.FIELD_SIZE, .FIELD_ADR, .END_FLD) ! and copy the data itself
: 284 0281 2 - .RTN_DSC [DSC_ADR];
: 285 0282 2
: 286 0283 2 RETURN .SUCCOD ! Return the code
: 287 0284 2
: 288 0285 1 END;

```

						00FC 00000	.ENTRY	NMA\$INSERTFLD, Save R2,R3,R4,R5,R6,R7	: 0198
		SE				04 C2 00002	SUBL2	#4, SP	: 0254
						7E D4 00005	CLRL	ADR	: 0255
						5E DD 00007	PUSHL	SP	
				08		AE 9F 00009	PUSHAB	SIZE	
				08		AC DD 0000C	PUSHL	FIELD_CODE	
		56		14		AC D0 0000F	MOVL	RTN_DSC, R6	
						56 DD 00013	PUSHL	R6	
		96	AF			04 FB 00015	CALLS	#4, NMA\$SEARCHFLD	
			57			50 D0 00019	MOVL	R0, SUCCOD	
			0C			57 E9 0001C	BLBC	SUCCOD, 1\$: 0256
04	AE		10	BC	0C	AC 2D 0001F	CMPCS	FIELD_SIZE, @FIELD_ADR, #0, SIZE, @ADR	: 0257
					00	BE			
						42 13 00029	BEQL	3\$	
					08	AC DD 0002B 1\$:	PUSHL	FIELD_CODE	: 0262
						56 DD 0002E	PUSHL	R6	
		00000000V	EF			02 FB 00030	CALLS	#2, NMA\$DELETEFLD	
			57			50 D0 00037	MOVL	R0, SUCCOD	
			51		04	A6 D0 0003A	MOVL	4(R6), END_FLD	: 0265
			50			66 3C 0003E	MOVZWL	(R6), R0	: 0266

NMAFIELDS
V04-000

Field Support Routines
NMA\$INSERTFLD Insert a Field Into a Record

G 15
16-Sep-1984 00:42:05
14-Sep-1984 12:50:02

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

Page 8
(4)

		51		50	C0	00041	ADDL2	R0, END_FLD	
52	0C	AC		04	C1	00044	ADDL3	#4, FIELD_SIZE, R2	: 0268
		50		66	3C	00049	MOVZWL	(R6), R0	: 0270
50	04	AC		50	C3	0004C	SUBL3	R0, BUF_SIZE, R0	
		50		52	D1	00051	CMP	R2, R0	
				04	1B	00054	BLEQU	2\$	
		50		18	D0	00056	MOVL	#24, R0	: 0272
					04	00059	RET		
		81	08	AC	B0	0005A	MOVW	FIELD_CODE, (END_FLD)+	: 0275
		81	0C	AC	B0	0005E	MOVW	FIELD_SIZE, (END_FLD)+	: 0276
61	10	BC	0C	AC	28	00062	MOV3	FIELD_SIZE, @FIELD_ADR, (END_FLD)	: 0280
66		53	04	A6	A3	00068	SUBW3	4(R6), R3, (R6)	: 0281
		50		57	D0	0006D	MOVL	SUCCOD, R0	: 0283
				04	00070		RET		: 0285

; Routine Size: 113 bytes, Routine Base: \$CODE\$ + 0051

```

290 0286 1 %SBTTL 'NMA$DELETEFLD Delete a Field From a Record'
291 0287 1 GLOBAL ROUTINE NMA$DELETEFLD (RTN_DSC, FIELD_CODE) = !
292 0288 1
293 0289 1 +-
294 0290 1 FUNCTIONAL DESCRIPTION:
295 0291 1
296 0292 1 Delete a field from a record. All other fields by that code are
297 0293 1 deleted as well.
298 0294 1
299 0295 1 FORMAL PARAMETERS:
300 0296 1
301 0297 1 RTN_DSC Address of descriptor of data portion of record.
302 0298 1 FIELD_CODE Value of the word containing the code
303 0299 1
304 0300 1 IMPLICIT INPUTS:
305 0301 1
306 0302 1 NONE
307 0303 1
308 0304 1 IMPLICIT OUTPUTS:
309 0305 1
310 0306 1 NONE
311 0307 1
312 0308 1 ROUTINE VALUE:
313 0309 1 COMPLETION CODES:
314 0310 1
315 0311 1 NMA$_SUCCESS A warm cuddly feeling
316 0312 1 NMA$SUCCFLDRPL A warm feeling with a replaced field
317 0313 1
318 0314 1 SIDE EFFECTS:
319 0315 1
320 0316 1 NONE
321 0317 1
322 0318 1 --
323 0319 1
324 0320 2 BEGIN
325 0321 2
326 0322 2 MAP
327 0323 2 RTN_DSC : REF BLOCK [8, 1]
328 0324 2 FIELD (DSC_FLDS)
329 0325 2 ;
330 0326 2
331 0327 2 LOCAL
332 0328 2 SUCCOD, ! Hold the success code for return
333 0329 2 END_FLD, ! End of a field to replace
334 0330 2 ADR, ! Address of field to replace
335 0331 2 SIZE, ! Size of replaced field
336 0332 2 END_BUF ! End of buffer address
337 0333 2 ;
338 0334 2
339 0335 2
340 0336 2 ADR = 0; ! Start at beginning of buffer
341 0337 2 SUCCOD = NMA$SUCCESS; ! Assume success with no replace
342 0338 2
343 0339 2 WHILE ! Find all the fields to replace
344 0340 2 NMA$SEARCHFLD (.RTN_DSC, .FIELD_CODE, SIZE, ADR)
345 0341 2 DO
346 0342 2 BEGIN

```

```

: 347 0343 3      END_FLD = .ADR + .SIZE;      ! End of replaced field
: 348 0344 3      ADR = .ADR - 4;        ! Address of control and data portion
: 349 0345 3      SIZE = .SIZE + 4;    ! Total size including control
: 350 0346 3      END_BUF = .RTN_DSC [DSC_ADR] ! End of data record
: 351 0347 3      + .RTN_DSC [DSC_SIZE];
: 352 0348 3      CHSMOVE (.END_BUF - .END_FLD, ! Copy data over field
: 353 0349 3      .END_FLD, .ADR
: 354 0350 3      );
: 355 0351 3      RTN_DSC [DSC_SIZE] =      ! Adjust size of record
: 356 0352 3      .RTN_DSC [DSC_SIZE] - .SIZE;
: 357 0353 3      SUCCOD = NMA$_SUCCFLDRPL ! Set the success code
: 358 0354 3      END
: 359 0355 3      :
: 360 0356 3      :
: 361 0357 3      RETURN .SUCCOD          ! Return the code
: 362 0358 3
: 363 0359 3      END;

```

			03FC 00000	.ENTRY	NMA\$DELETEFLD, Save R2,R3,R4,R5,R6,R7,R8,R9	: 0287
	SE		04 C2 00002	SUBL2	#4, SP	
			7E D4 00005	CLRL	ADR	: 0336
	59		01 D0 00007	MOVL	#1, SUCCOD	: 0337
	57	04	AC D0 0000A	MOVL	RTN_DSC, R7	: 0340
			5E DD 0000E 1\$:	PUSHL	SP	
		08	AE 9F 00010	PUSHAB	SIZE	
		08	AC DD 00013	PUSHL	FIELD_CODE	
			57 DD 00016	PUSHL	R7	
	FF21	CF	04 FB 00018	CALLS	#4, NMA\$SEARCHFLD	
		25	50 E9 0001D	BLBC	R0, 2\$	
	58	6E	04 AE C1 00020	ADDL3	SIZE, ADR, END_FLD	: 0343
		6E	04 C2 00025	SUBL2	#4, ADR	: 0344
		04	AE C0 00028	ADDL2	#4, SIZE	: 0345
		56	67 3C 0002C	MOVZWL	(R7), END_BUF	: 0347
		56	04 A7 C0 0002F	ADDL2	4(R7), END_BUF	
	00	50	58 C3 00033	SUBL3	END_FLD, END_BUF, R0	: 0348
	BE	68	50 28 00037	MOVCL3	R0, (END_FLD), @ADR	: 0349
		67	04 AE A2 0003C	SUBW2	SIZE, (R7)	: 0352
		59	09 D0 00040	MOVL	#9, SUCCOD	: 0353
			C9 11 00043	BRB	1\$	
		50	59 D0 00045 2\$:	MOVL	SUCCOD, R0	: 0357
			04 00048	RET		: 0359

: Routine Size: 73 bytes, Routine Base: \$CODE\$ + 00C2

NMAFIELDS
V04-000

Field Support Routines
NMA\$DELETEFLD Delete a Field From a Record

J 15
16-Sep-1984 00:42:05
14-Sep-1984 12:50:02

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

Page 11
(6)

: 365 0360 1 END !End of module
: 366 0361 0 ELUDOM

PSECT SUMMARY

Name Bytes Attributes
\$CODE\$ 267 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SHRLIB]NMLIBRY.L32;1	887	4	0	47	00:00.2
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	7776	0	0	581	00:01.7

COMMAND QUALIFIERS

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

: Size: 267 code + 0 data bytes
: Run Time: 00:08.5
: Elapsed Time: 00:21.7
: Lines/CPU Min: 2542
: Lexemes/CPU-Min: 8154
: Memory Used: 60 pages
: Compilation Complete

001	002	003	004	005	006	007	008	009	010	011	012	013	014	015	016	017	018	019	020	021	022	023	024	025	026	027	028	029	030	031	032	033	034	035	036	037	038	039	040	041	042	043	044	045	046	047	048	049	050	051	052	053	054	055	056	057	058	059	060	061	062	063	064	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079	080	081	082	083	084	085	086	087	088	089	090	091	092	093	094	095	096	097	098	099	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------