


```

FFFFFFFFF 000000 RRRRRRRR 000000 PPPPPPPP NN NN KK KK EEEEEEEEE YY YY
FFFFFFFFF 000000 RRRRRRRR 000000 PPPPPPPP NN NN KK KK EEEEEEEEE YY YY
FF 00 00 RR RR 00 00 PP PP NN NN KK KK EE YY YY
FF 00 00 RR RR 00 00 PP PP NN NN KK KK EE YY YY
FF 00 00 RR RR 00 00 PP PP NN NN KK KK EE YY YY
FF 00 00 RR RR 00 00 PP PP NN NN KK KK EE YY YY
FFFFFFFFF 00 00 RRRRRRRR 00 00 PPPPPPPP NN NN NN NN KK KK EE YY YY
FFFFFFFFF 00 00 RRRRRRRR 00 00 PPPPPPPP NN NN NN NN KK KK EE YY YY
FF 00 00 RR RR 00 00 PP PP NN NN NN NN KK KK EE YY YY
FF 00 00 RR RR 00 00 PP PP NN NN NN NN KK KK EE YY YY
FF 00 00 RR RR 00 00 PP PP NN NN NN NN KK KK EE YY YY
FF 00 00 RR RR 00 00 PP PP NN NN NN NN KK KK EE YY YY
FF 000000 RR RR 000000 PP NN NN KK KK EEEEEEEEE YY
FF 000000 RR RR 000000 PP NN NN KK KK EEEEEEEEE YY

```

```

LL 111111 SSSSSSSS
LL 111111 SSSSSSSS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SSSSSS
LL 11 SSSSSS
LL 11 SS
LL 11 SS
LL 11 SS
LL 11 SS
LLLLLLLLLL 111111 SSSSSSSS
LLLLLLLLLL 111111 SSSSSSSS

```

```
1 0001 0 MODULE FOR$$OPEN_KEYWD (%TITLE'Translate OPEN keyword value strings'  
2 0002 0 IDENT = '1-003' ! file: FOROPNKEY.B32 Edit: SBL1003  
3 0003 0 ) =  
4 0004 1 BEGIN  
5 0005 1  
6 0006 1  
7 0007 1 *****  
8 0008 1 *  
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *  
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *  
11 0011 1 * ALL RIGHTS RESERVED. *  
12 0012 1 *  
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *  
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *  
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *  
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *  
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *  
18 0018 1 * TRANSFERRED. *  
19 0019 1 *  
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *  
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *  
22 0022 1 * CORPORATION. *  
23 0023 1 *  
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *  
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *  
26 0026 1 *  
27 0027 1 *  
28 0028 1 *****  
29 0029 1  
30 0030 1  
31 0031 1 **  
32 0032 1 FACILITY: FORTRAN Support Library - Not user callable  
33 0033 1  
34 0034 1 ABSTRACT:  
35 0035 1  
36 0036 1 This module implements OPEN/CLOSE keyword values passed  
37 0037 1 by string descriptor rather than by literal value. It  
38 0038 1 scans its table for matching keywords and returns the  
39 0039 1 equivalent literal value.  
40 0040 1  
41 0041 1 ENVIRONMENT: User access mode, AST reentrant  
42 0042 1  
43 0043 1 AUTHOR: Steven B. Lionel, CREATION DATE: 18-April-1979  
44 0044 1  
45 0045 1 MODIFIED BY:  
46 0046 1  
47 0047 1 Steven B. Lionel, 18-April-1979 : Version 1  
48 0048 1 1-001 - Original  
49 0049 1 1-002 - Male tables PIC. SBL 1-May-1979  
50 0050 1 1-003 - Add new keywords for stream recordtype. Use prologue file.  
51 0051 1 SBL 1-March-1983  
52 0052 1 --
```

```
54 0053 1 |
55 0054 1 | PROLOGUE FILE:
56 0055 1 |
57 0056 1 |
58 0057 1 | REQUIRE 'RTLIN:FORPROLOG';      ! FOR$ declarations
59 0123 1 |
60 0124 1 |
61 0125 1 | TABLE OF CONTENTS:
62 0126 1 |
63 0127 1 |
64 0128 1 | FORWARD ROUTINE
65 0129 1 |     FOR$$OPEN_KEYWD      ! Searches keyword table
66 0130 1 |     ;
67 0131 1 |
68 0132 1 |
69 0133 1 | MACROS:
70 0134 1 |
71 0135 1 |
72 0136 1 |     MACRO
73 0137 1 |         KWD (A,X) = WORD (%CHARCOUNT (X)), (UPLIT BYTE (X) - A)%,
74 0138 1 |         KWDSW_KEY = 0,0,16,0%,
75 0139 1 |         KWDSW_COUNT = 0,16,16,0%,
76 0140 1 |         KWDSA_KEYWD = 4,0,32,0%;
77 0141 1 |
78 0142 1 |
79 0143 1 | EQUATED SYMBOLS:
80 0144 1 |
81 0145 1 |
82 0146 1 |
83 0147 1 | OWN STORAGE:
84 0148 1 |
85 0149 1 |
86 0150 1 | OWN
87 0151 1 |     KEYWDSACCESS: VECTOR [9] PSECT ( FOR$CODE) INITIAL (
88 0152 1 |         WORD (OPENS$K_ACC_DIR), KWD (KEYWDSACCESS, 'DIRECT'),
89 0153 1 |         WORD (OPENS$K_ACC_SEQ), KWD (KEYWDSACCESS, 'SEQUENTIAL'),
90 0154 1 |         WORD (OPENS$K_ACC_APP), KWD (KEYWDSACCESS, 'APPEND'),
91 0155 1 |         WORD (OPENS$K_ACC_KEY), KWD (KEYWDSACCESS, 'KEYED'),
92 0156 1 |         0),
93 0157 1 |
94 0158 1 |     KEYWDSBLANK: VECTOR [5] PSECT ( FOR$CODE) INITIAL (
95 0159 1 |         WORD (OPENS$K_BLK_ZER), KWD (KEYWDSBLANK, 'ZERO'),
96 0160 1 |         WORD (OPENS$K_BLK_NUL), KWD (KEYWDSBLANK, 'NULL'),
97 0161 1 |         0),
98 0162 1 |
99 0163 1 |     KEYWDS$CARRIAGE: VECTOR [7] PSECT ( FOR$CODE) INITIAL (
100 0164 1 |         WORD (OPENS$K_CAR_FOR), KWD (KEYWDS$CARRIAGE, 'FORTRAN'),
101 0165 1 |         WORD (OPENS$K_CAR_LIS), KWD (KEYWDS$CARRIAGE, 'LIST'),
102 0166 1 |         WORD (OPENS$K_CAR_NON), KWD (KEYWDS$CARRIAGE, 'NONE'),
103 0167 1 |         0),
104 0168 1 |
105 0169 1 |     KEYWDS$DISPOSE: VECTOR [15] PSECT ( FOR$CODE) INITIAL (
106 0170 1 |         WORD (OPENS$K_DIS_SAV), KWD (KEYWDS$DISPOSE, 'SAVE'),
107 0171 1 |         WORD (OPENS$K_DIS_SAV), KWD (KEYWDS$DISPOSE, 'KEEP'),      ! synonym
108 0172 1 |         WORD (OPENS$K_DIS_DEL), KWD (KEYWDS$DISPOSE, 'DELETE'),
109 0173 1 |         WORD (OPENS$K_DIS_PRI), KWD (KEYWDS$DISPOSE, 'PRINT'),
110 0174 1 |         WORD (OPENS$K_DIS_SUB), KWD (KEYWDS$DISPOSE, 'SUBMIT'),
```

```

111 0175 1 WORD (OPENS$DIS_PRDE), KWD (KEYWDS$DISPOSE, 'PRINT/DELETE'),
112 0176 1 WORD (OPENS$DIS_SUDE), KWD (KEYWDS$DISPOSE, 'SUBMIT/DELETE'),
113 0177 1 0),
114 0178 1
115 0179 1 KEYWDS$FORM: VECTOR [5] PSECT (FOR$CODE) INITIAL (
116 0180 1 WORD (OPENS$FOR_FOR), KWD (KEYWDS$FORM, 'FORMATTED'),
117 0181 1 WORD (OPENS$FOR_UNF), KWD (KEYWDS$FORM, 'UNFORMATTED'),
118 0182 1 0),
119 0183 1
120 0184 1 KEYWDS$ORGANIZ: VECTOR [7] PSECT (FOR$CODE) INITIAL (
121 0185 1 WORD (OPENS$ORG_SEQ), KWD (KEYWDS$ORGANIZ, 'SEQUENTIAL'),
122 0186 1 WORD (OPENS$ORG_REL), KWD (KEYWDS$ORGANIZ, 'RELATIVE'),
123 0187 1 WORD (OPENS$ORG_IDX), KWD (KEYWDS$ORGANIZ, 'INDEXED'),
124 0188 1 0),
125 0189 1
126 0190 1 KEYWDS$RECORDTY: VECTOR [13] PSECT (FOR$CODE) INITIAL (
127 0191 1 WORD (OPENS$REC_FIX), KWD (KEYWDS$RECORDTY, 'FIXED'),
128 0192 1 WORD (OPENS$REC_VAR), KWD (KEYWDS$RECORDTY, 'VARIABLE'),
129 0193 1 WORD (OPENS$REC_SEGM), KWD (KEYWDS$RECORDTY, 'SEGMENTED'),
130 0194 1 WORD (OPENS$REC_STM), KWD (KEYWDS$RECORDTY, 'STREAM'),
131 0195 1 WORD (OPENS$REC_STMCR), KWD (KEYWDS$RECORDTY, 'STREAM_CR'),
132 0196 1 WORD (OPENS$REC_STMLF), KWD (KEYWDS$RECORDTY, 'STREAM_LF'),
133 0197 1 0),
134 0198 1
135 0199 1 KEYWDS$TYPE: VECTOR [9] PSECT (FOR$CODE) INITIAL (
136 0200 1 WORD (OPENS$TYP_OLD), KWD (KEYWDS$TYPE, 'OLD'),
137 0201 1 WORD (OPENS$TYP_NEW), KWD (KEYWDS$TYPE, 'NEW'),
138 0202 1 WORD (OPENS$TYP_SCR), KWD (KEYWDS$TYPE, 'SCRATCH'),
139 0203 1 WORD (OPENS$TYP_UNK), KWD (KEYWDS$TYPE, 'UNKNOWN'),
140 0204 1 0),
141 0205 1
142 0206 1 KEY_TABLE: VECTOR [OPENS$KEY_MAX + 1] PSECT (FOR$CODE) INITIAL (
143 0207 1 0,
144 0208 1 0,
145 0209 1 (KEYWDS$DISPOSE - KEY_TABLE),
146 0210 1 0,
147 0211 1 (KEYWDS$ACCESS - KEY_TABLE),
148 0212 1 (KEYWDS$FORM - KEY_TABLE),
149 0213 1 0,
150 0214 1 (KEYWDS$CARRIAGE - KEY_TABLE),
151 0215 1 0,
152 0216 1 0,
153 0217 1 0,
154 0218 1 0,
155 0219 1 0,
156 0220 1 0,
157 0221 1 0,
158 0222 1 (KEYWDS$TYPE - KEY_TABLE),
159 0223 1 0,
160 0224 1 0,
161 0225 1 0,
162 0226 1 (KEYWDS$ORGANIZ - KEY_TABLE),
163 0227 1 (KEYWDS$RECORDTY - KEY_TABLE),
164 0228 1 0,
165 0229 1 0,
166 0230 1 0,
167 0231 1 (KEYWDS$BLANK - KEY_TABLE));

```

```

0 = not used
1 = UNIT
2 = DISPOSE
3 = ERR
4 = ACCESS
5 = FORM
6 = RECORDSIZE
7 = CARRIAGECONTROL
8 = READONLY
9 = BUFFERCOUNT
10 = INITIALSIZE
11 = EXTENDSIZE
12 = NOSPANBLOCKS
13 = SHARED
14 = NAME
15 = TYPE
16 = MAXREC
17 = ASSOCIATEVARIABLE
18 = BLOCKSIZE
19 = ORGANIZATION
20 = RECORDTYPE
21 = USEROPEN
22 = IOSTAT
23 = KEY
24 = BLANK

```

FOR\$\$OPEN_KEYWD Translate OPEN keyword value strings
1-003

N 13
16-Sep-1984 00:40:46
14-Sep-1984 12:32:18

VAX-11 Bliss-32 V4.0-742
[FORRTL.SRC]FOROPNKEY.B32;1

Page 4
(2)

```
.. 168      0232  1
.. 169      0233  1
.. 170      0234  1  EXTERNAL REFERENCES:
.. 171      0235  1
.. 172      0236  1
.. 173      0237  1  EXTERNAL ROUTINE
.. 174      0238  1    FOR$$GET_VM:
.. 175      0239  1    FOR$$FREE_VM:
.. 176      0240  1
```

```
! Allocate virtual memory
. Free virtual memory
```

```
178 0241 1 GLOBAL ROUTINE FOR$$OPEN_KEYWD (  
179 0242 1     KEYWORD,  
180 0243 1     KEYSTRING) =  
181 0244 1  
182 0245 1  ++  
183 0246 1  FUNCTIONAL DESCRIPTION:  
184 0247 1  
185 0248 1     FOR$$OPEN_KEYWD is called from FOR$$OPECLD ARG inside  
186 0249 1     FOR$OPEN. It is called when an OPEN or CLOSE keyword, which  
187 0250 1     normally has a literal value computed by the compiler, is  
188 0251 1     given as a run-time character string.  
189 0252 1  
190 0253 1     This routine looks up the given string and returns the  
191 0254 1     equivalent small integer literal value. It automatically  
192 0255 1     forces upper case, and the comparison ignores trailing blanks.  
193 0256 1     Like in the FORTRAN compiler, leading and embedded blanks are  
194 0257 1     not allowed, and the strings must match exactly, i.e. no  
195 0258 1     abbreviations.  
196 0259 1  
197 0260 1  CALLING SEQUENCE:  
198 0261 1  
199 0262 1     value.wl.v = FOR$$OPEN_KEYWD (keyword.rl.v, keystring.rx.dx)  
200 0263 1  
201 0264 1  
202 0265 1  FORMAL PARAMETERS:  
203 0266 1  
204 0267 1     keyword           - The OPEN key number for the desired keyword.  
205 0268 1  
206 0269 1     keystring        - The address of a string descriptor containing  
207 0270 1     the string to be looked up.  
208 0271 1  
209 0272 1  IMPLICIT INPUTS:  
210 0273 1  
211 0274 1     NONE  
212 0275 1  
213 0276 1  IMPLICIT OUTPUTS:  
214 0277 1  
215 0278 1     NONE  
216 0279 1  
217 0280 1  ROUTINE VALUE:  
218 0281 1  
219 0282 1     -1 if "keyword" has no valid literal values.  
220 0283 1     0 if "keystring" is not valid for keyword "keyword".  
221 0284 1     The correct literal value otherwise.  
222 0285 1  
223 0286 1  SIDE EFFECTS:  
224 0287 1  
225 0288 1     NONE  
226 0289 1  
227 0290 1  --  
228 0291 1  
229 0292 2     BEGIN  
230 0293 2  
231 0294 2     MAP  
232 0295 2     KEYSTRING: REF BLOCK [,BYTE];  
233 0296 2  
234 0297 2     LOCAL
```

```
235 0298 2 KEY_TAB: REF BLOCK [,BYTE], ! Current table block address
236 0299 2 TAB_ADDR, ! Base address of keyword table entry
237 0300 2 STRING_ADDR, ! Address of allocated string
238 0301 2 STRING_SIZE; ! Size of allocated string
239 0302 2
240 0303 2 !+
241 0304 2 ! Copy KEYSTRING to a local string, forcing upper case.
242 0305 2 !-
243 0306 2
244 0307 2 STRING_SIZE = .KEYSTRING [DSC$W_LENGTH];
245 0308 2 STRING_ADDR = FOR$$GET_VM (.STRING_SIZE);
246 0309 2
247 0310 2 BEGIN
248 0311 2 LOCAL
249 0312 2 SOURCE, ! Source string address
250 0313 2 DEST; ! Destination string address
251 0314 2 SOURCE = .KEYSTRING [DSC$A_POINTER];
252 0315 2 DEST = .STRING_ADDR;
253 0316 2
254 0317 2 INCR I FROM 1 TO .STRING_SIZE DO
255 0318 2 BEGIN
256 0319 2 LOCAL
257 0320 2 C; ! Character read
258 0321 2 C = CH$RCHAR_A (SOURCE);
259 0322 2 IF (.C GEQU %C'a' AND .C LEQU %C'z')
260 0323 2 THEN
261 0324 2 CH$WCHAR_A (.C - (%C'a' - %C'A'), DEST)
262 0325 2 ELSE
263 0326 2 CH$WCHAR_A (.C, DEST);
264 0327 2 END;
265 0328 2 END;
266 0329 2
267 0330 2 !+
268 0331 2 ! See if KEYWORD has an entry in the table. If not, there
269 0332 2 ! are no literal values allowed for this keyword.
270 0333 2 !-
271 0334 2
272 0335 2 KEY_TAB = .KEY TABLE [.KEYWORD];
273 0336 2 IF .KEY_TAB EQ 0
274 0337 2 THEN
275 0338 2 BEGIN
276 0339 2 FOR$$FREE_VM (.STRING_SIZE, .STRING_ADDR);
277 0340 2 RETURN (-1); ! Indicates invalid argument to routine
278 0341 2 END;
279 0342 2
280 0343 2 !+
281 0344 2 ! Now get the true address of the keyword table entry
282 0345 2 !-
283 0346 2
284 0347 2 KEY_TAB = KEY TABLE + .KEY_TAB;
285 0348 2 TAB_ADDR = .KEY_TAB; ! Save base address for relative addressing
286 0349 2
287 0350 2 !+
288 0351 2 ! Scan the keyword table to see if the key string matches a
289 0352 2 ! valid keyword.
290 0353 2 !-
291 0354 2
```



```

292 0355 2 WHILE .KEY_TAB [KWDSW_KEY] NEQ 0 DO
293 0356 BEGIN
294 0357 IF CH$EQL (.STRING_SIZE, .STRING_ADDR, .KEY_TAB [KWDSW_COUNT],
295 0358 .TAB_ADDR + .KEY_TAB [KWDSA_KEYWD], %C' ')
296 0359 THEN
297 0360 BEGIN
298 0361 FOR$$FREE_VM (.STRING_SIZE, .STRING_ADDR);
299 0362 RETURN (.KEY_TAB [KWDSW_KEY]);
300 0363 END;
301 0364 KEY_TAB = .KEY_TAB + 8; ! Go to next keyword
302 0365 END;
303 0366
304 0367
305 0368
306 0369
307 0370
308 0371
309 0372
310 0373
311 0374 1

```

```

!+
!- If we got here, then we must not have found a matching key.
FOR$$FREE_VM (.STRING_SIZE, .STRING_ADDR);
RETURN (0); ! Indicated keyword value error
END;

```

										.TITLE	FOR\$\$OPEN_KEYWD Translate OPEN keyword value strings			
										.IDENT	\1-003\			
										.PSECT	_FOR\$CODE, NOWRT, SHR, PIC, 2			
4C	41	49	54	54	43	45	52	49	44	00000	P.AAA:	.ASCII	\DIRECT\	
				4E	45	55	51	45	53	00006	P.AAB:	.ASCII	\SEQUENTIAL\	
				44	4E	45	50	50	41	00010	P.AAC:	.ASCII	\APPEND\	
					44	45	59	45	4B	00016	P.AAD:	.ASCII	\KEYED\	
										0001B	.BLKB	1		
										0001	0001C	KEYWDS\$ACCESS:		
											.WORD	1		
										0006	0001E	.WORD	6	
										00000000*	00020	.LONG	<P.AAA-KEYWDS\$ACCESS>	
										0002	00024	.WORD	2	
										000A	00026	.WORD	10	
										00000000*	00028	.LONG	<P.AAB-KEYWDS\$ACCESS>	
										0003	0002C	.WORD	3	
										0006	0002E	.WORD	6	
										00000000*	00030	.LONG	<P.AAC-KEYWDS\$ACCESS>	
										0004	00034	.WORD	4	
										0005	00036	.WORD	5	
										00000000*	00038	.LONG	<P.AAD-KEYWDS\$ACCESS>	
										00000000	0003C	.LONG	0	
4F	52	45	5A	4C	4C	55	4E	00040	00044	P.AAE:	.ASCII	\ZERO\		
								00044	00048	P.AAF:	.ASCII	\NULL\		
										0001	00048	KEYWDS\$BLANK:		
											.WORD	1		
										0004	0004A	.WORD	4	
										00000000*	0004C	.LONG	<P.AAE-KEYWDS\$BLANK>	
										0002	00050	.WORD	2	
										0004	00052	.WORD	4	
										00000000*	00054	.LONG	<P.AAF-KEYWDS\$BLANK>	
										00000000	00058	.LONG	0	

4E	41	52	54	52	4F	46	0005C	P.AAG:	.ASCII	\FORTRAN\									
			54	53	49	4C	00063	P.AAH:	.ASCII	\LIST\									
			45	4E	4F	4E	00067	P.AAI:	.ASCII	\NONE\									
							0006B		.BLKB	1									
							0001	0006C	KEYWDS\$CARRIAGE:										
									.WORD	1									
							0007	0006E		.WORD	7								
							00000000*	00070		.LONG	<P.AAG-KEYWDS\$CARRIAGE>								
							0002	00074		.WORD	2								
							0004	00076		.WORD	4								
							00000000*	00078		.LONG	<P.AAH-KEYWDS\$CARRIAGE>								
							0003	0007C		.WORD	3								
							0004	0007E		.WORD	4								
							00000000*	00080		.LONG	<P.AAI-KEYWDS\$CARRIAGE>								
							00000000	00084		.LONG	0								
			45	56	41	53	00088	P.AAJ:	.ASCII	\SAVE\									
			50	45	45	4B	0008C	P.AAK:	.ASCII	\KEEP\									
			45	54	45	4C	45	44	00090	P.AAL:	.ASCII	\DELETE\							
			54	49	4E	49	52	50	00096	P.AAM:	.ASCII	\PRINT\							
			54	49	4D	42	55	53	0009B	P.AAN:	.ASCII	\SUBMIT\							
			45	54	45	4C	45	44	2F	54	49	4D	42	55	53	000A1	P.AAO:	.ASCII	\PRINT/DELETE\
45	54	45	4C	45	44	2F	54	49	4D	42	55	53	000AD	P.AAP:	.ASCII	\SUBMIT/DELETE\			
													000BA		.BLKB	2			
							0001	000BC	KEYWDS\$DISPOSE:										
									.WORD	1									
							0004	000BE		.WORD	4								
							00000000*	000C0		.LONG	<P.AAJ-KEYWDS\$DISPOSE>								
							0001	000C4		.WORD	1								
							0004	000C6		.WORD	4								
							00000000*	000C8		.LONG	<P.AAK-KEYWDS\$DISPOSE>								
							0002	000CC		.WORD	2								
							0006	000CE		.WORD	6								
							00000000*	000D0		.LONG	<P.AAL-KEYWDS\$DISPOSE>								
							0003	000D4		.WORD	3								
							0005	000D6		.WORD	5								
							00000000*	000D8		.LONG	<P.AAM-KEYWDS\$DISPOSE>								
							0004	000DC		.WORD	4								
							0006	000DE		.WORD	6								
							00000000*	000E0		.LONG	<P.AAN-KEYWDS\$DISPOSE>								
							0005	000E4		.WORD	5								
							000C	000E6		.WORD	12								
							00000000*	000E8		.LONG	<P.AAO-KEYWDS\$DISPOSE>								
							0006	000EC		.WORD	6								
							000D	000EE		.WORD	13								
							00000000*	000F0		.LONG	<P.AAP-KEYWDS\$DISPOSE>								
							00000000	000F4		.LONG	0								
			44	45	54	54	41	4D	52	4F	46	4E	55	000F8	P.AAQ:	.ASCII	\FORMATTED\		
														00101	P.AAR:	.ASCII	\UNFORMATTED\		
														0001	0010C	KEYWDS\$FORM:			
																.WORD	1		
														0009	0010E		.WORD	9	
														00000000*	00110		.LONG	<P.AAQ-KEYWDS\$FORM>	
														0002	0J114		.WORD	2	
														000B	00116		.WORD	11	
														00000000*	00118		.LONG	<P.AAR-KEYWDS\$FORM>	
														00000000	0011C		.LONG	0	
			4C	41	49	54	4E	45	55	51	45	53	00120	P.AAS:	.ASCII	\SEQUENTIAL\			

45	56	49	54	41	4C	45	52	0012A	P.AAT:	.ASCII	\RELATIVE\	:	
	44	45	58	45	44	4E	49	00132	P.AAU:	.ASCII	\INDEXED\	:	
								00139		.BLKB	3	:	
								0001	0013C	KEYWDS\$ORGANIZ:		:	
										.WORD	1	:	
								000A	0013E		.WORD	10	:
								00000000*	00140	.LONG	<P.AAS-KEYWDS\$ORGANIZ>	:	
								0002	00144	.WORD	2	:	
								0008	00146	.WORD	8	:	
								00000000*	00148	.LONG	<P.AAT-KEYWDS\$ORGANIZ>	:	
								0003	0014C	.WORD	3	:	
								0007	0014E	.WORD	7	:	
								00000000*	00150	.LONG	<P.AAU-KEYWDS\$ORGANIZ>	:	
								00000000	00154	.LONG	0	:	
				44	45	58	49	46	00158	P.AAV:	.ASCII	\FIXED\	:
	45	4C	42	41	49	52	41	56	0015D	P.AAW:	.ASCII	\VARIABLE\	:
44	45	54	4E	45	4D	47	45	53	00165	P.AAX:	.ASCII	\SEGMENTED\	:
				41	45	52	54	53	0016E	P.AAY:	.ASCII	\STREAM\	:
52	43	5F	4D	41	45	52	54	53	00174	P.AAZ:	.ASCII	\STREAM_CR\	:
46	4C	5F	4D	41	45	52	54	53	0017D	P.ABA:	.ASCII	\STREAM_LF\	:
									00186		.BLKB	2	:
									0001	00188	KEYWDS\$RECORDTY:		:
										.WORD	1	:	
									0005	0018A	.WORD	5	:
									00000000*	0018C	.LONG	<P.AAV-KEYWDS\$RECORDTY>	:
									0002	00190	.WORD	2	:
									0008	00192	.WORD	8	:
									00000000*	00194	.LONG	<P.AAW-KEYWDS\$RECORDTY>	:
									0003	00198	.WORD	3	:
									0009	0019A	.WORD	9	:
									00000000*	0019C	.LONG	<P.AAX-KEYWDS\$RECORDTY>	:
									0004	001A0	.WORD	4	:
									0006	001A2	.WORD	6	:
									00000000*	001A4	.LONG	<P.AAY-KEYWDS\$RECORDTY>	:
									0005	001A8	.WORD	5	:
									0009	001AA	.WORD	9	:
									00000000*	001AC	.LONG	<P.AAZ-KEYWDS\$RECORDTY>	:
									0006	001B0	.WORD	6	:
									0009	001B2	.WORD	9	:
									00000000*	001B4	.LONG	<P.ABA-KEYWDS\$RECORDTY>	:
									00000000	001B8	.LONG	0	:
				44	4C	4F			001BC	P.ABB:	.ASCII	\OLD\	:
				57	45	4E			001BF	P.ABC:	.ASCII	\NEW\	:
	48	43	54	41	52	43	53		001C2	P.ABD:	.ASCII	\SCRATCH\	:
	4E	57	4F	4E	4B	4E	55		001C9	P.ABE:	.ASCII	\UNKNOWN\	:
									0001	001D0	KEYWDS\$TYPE:		:
										.WORD	1	:	
									0003	001D2	.WORD	3	:
									00000000*	001D4	.LONG	<P.ABB-KEYWDS\$TYPE>	:
									0002	001D8	.WORD	2	:
									0003	001DA	.WORD	3	:
									00000000*	001DC	.LONG	<P.ABC-KEYWDS\$TYPE>	:
									0003	001E0	.WORD	3	:
									0007	001E2	.WORD	7	:
									00000000*	001E4	.LONG	<P.ABD-KEYWDS\$TYPE>	:
									0004	001E8	.WORD	4	:
									0007	001EA	.WORD	7	:

FOR\$\$OPEN_KEYWD Translate OPEN keyword value strings
1-003

H 14
16-Sep-1984 00:40:46 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:32:18 [FORRTL.SRC]FOROPNKEY.B32;1

Page 11
(3)

7E	0A	12	0007A	BNEQ	7\$:	
68	56	7D	0007C	MOVQ	STRING SIZE, -(SP)	:	0361
50	02	FB	0007F	CALLS	#2, FOR\$\$FREE_VM	:	
	64	3C	00082	MOVZWL	(KEY_TAB), R0	:	0362
		04	00085	RET		:	
54	08	C0	00086	7\$: ADDL2	#8, KEY_TAB	:	0364
	DF	11	00089	BRB	6\$:	0355
7E	56	7D	0008B	8\$: MOVQ	STRING SIZE, -(SP)	:	0371
68	02	FB	0008E	CALLS	#2, FOR\$\$FREE_VM	:	
	50	D4	00091	CLRL	R0	:	0372
		04	00093	RET		:	0374

; Routine Size: 148 bytes, Routine Base: _FOR\$CODE + 0260

: 313 0375 1 END !End of module
: 314 0376 0 ELUDOM

PSECT SUMMARY

Name Bytes Attributes
_FOR\$CODE 756 NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	2	0	581	00:01.1
-\$255\$DUA28:[FORRTL.OBJ]FORLIB.L32;1	711	31	4	52	00:00.6
-\$255\$DUA28:[FORRTL.OBJ]RTLLIB.L32;1	36	0	0	8	00:00.1

COMMAND QUALIFIERS

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

: Size: 148 code + 608 data bytes
: Run Time: 00:09.4
: Elapsed Time: 00:23.4
: Lines/CPU Min: 2400
: Lexemes/CPU-Min: 17265
: Memory Used: 98 pages
: Compilation Complete

