


```

FFFFFFFFF 000000 RRRRRRRR CCCCCCCC LL 000000 SSSSSSSS EEEEEEEEE
FFFFFFFFF 000000 RRRRRRRR CCCCCCCC LL 000000 SSSSSSSS EEEEEEEEE
FF 00 00 RR RR CC 00 00 SS SSSSSSSS EEEEEEEEE
FF 00 00 RR RR CC 00 00 SS SSSSSSSS EEEEEEEEE
FF 00 00 RR RR CC 00 00 SS SSSSSSSS EEEEEEEEE
FF 00 00 RR RR CC 00 00 SS SSSSSSSS EEEEEEEEE
FFFFFFFFF 00 00 RRRRRRRR CCCCCCCC LL 00 00 SSSSSSSS EEEEEEEEE
FFFFFFFFF 00 00 RRRRRRRR CCCCCCCC LL 00 00 SSSSSSSS EEEEEEEEE
FF 00 00 RR RR CC 00 00 SS SSSSSSSS EEEEEEEEE
FF 00 00 RR RR CC 00 00 SS SSSSSSSS EEEEEEEEE
FF 00 00 RR RR CC 00 00 SS SSSSSSSS EEEEEEEEE
FF 00 00 RR RR CC 00 00 SS SSSSSSSS EEEEEEEEE
FF 000000 RR RR CCCCCCCC LLLLLLLLLL 000000 SSSSSSSS EEEEEEEEE
FF 000000 RR RR CCCCCCCC LLLLLLLLLL 000000 SSSSSSSS EEEEEEEEE

```

```

LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```

```

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

```

```

1 0001 0 MODULE FOR$CLOSE (%TITLE 'FORTRAN CLOSE and default CLOSE'
2 0002 0 IDENT = '1-021' ! File: FORCLOSE.B32 Edit: SBL1021
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 **
30 0030 1 FACILITY: FORTRAN Support Library - user callable
31 0031 1
32 0032 1 ABSTRACT:
33 0033 1
34 0034 1 This module closes a file on a specified logical unit
35 0035 1 (LUN) and deallocates the I/O statement block (ISB) and
36 0036 1 RMS Record Access Block (RAB) control blocks which were
37 0037 1 allocated by OPEN or default OPEN for this LUN.
38 0038 1
39 0039 1 ENVIRONMENT: User access level; re-entrant, AST level or not.
40 0040 1
41 0041 1 AUTHOR: Thomas N. Hastings, CREATION DATE: 28-Apr-77; Version 01
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 Thomas N. Hastings, 28-Apr-77, Version 01
46 0046 1 [Previous edit history removed. SBL 30-Sep-1982]
47 0047 1 1-017 - Move the BUILTIN ACTUALCOUNT into the routine.
48 0048 1 The next BLISS compiler will require this. JBS 20-Aug-1980
49 0049 1 1-018 - Add DEF_DSC to calling sequence for FOR$$OPECLO_ARG. JAW
50 0050 1 02-Jul-1981
51 0051 1 1-019 - If there is a FAB still hanging around, deallocate it. DGP 18-Dec-1981
52 0052 1 1-020 - Reflect separation of FOR$ CCB structures from BASIC. SBL 30-Sep-1982
53 0053 1 1-021 - Remove deallocation of FAB. SBL 20-Jan-1983
54 0054 1 --
55 0055 1

```

```
57 0056 1 |
58 0057 1 | PROLOGUE FILE:
59 0058 1 |
60 0059 1 |
61 0060 1 | REQUIRE 'RTLIN:FORPROLOG';          FORTRAN definitions
62 0126 1 |
63 0127 1 |
64 0128 1 | TABLE OF CONTENTS:
65 0129 1 |
66 0130 1 |
67 0131 1 | FORWARD ROUTINE
68 0132 1 |   FOR$CLOSE;                        ! FORTRAN CLOSE statement
69 0133 1 |
70 0134 1 |
71 0135 1 | EQUATED SYMBOLS:
72 0136 1 |
73 0137 1 |   NONE
74 0138 1 |
75 0139 1 | OWN STORAGE:
76 0140 1 |
77 0141 1 |   NONE
78 0142 1 |
79 0143 1 | EXTERNAL REFERENCES:
80 0144 1 |
81 0145 1 |
82 0146 1 | EXTERNAL ROUTINE
83 0147 1 |   FOR$$OPECLO_ARG : NOVALUE,        ! Get OPEN/CLOSE arguments
84 0148 1 |   FOR$$SIGNAL_STO : NOVALUE,        ! Convert FORTRAN error#
85 0149 1 |                                       ! to 32 condition code and SIGNAL_STOP
86 0150 1 |   FOR$$ERR_OPECLO,                 ! OPEN/CLOSE error condition handler
87 0151 1 |                                       ! resignals or unwinds depending on whether user specified E
88 0152 1 |                                       ! (in OPEN module)
89 0153 1 |   FOR$$CB_PUSH : JSB_CB_PUSH NOVALUE, ! Push down active I/O and allocate
90 0154 1 |                                       ! LUB/ISB/RAB if not already for unit.
91 0155 1 |   FOR$$CB_POP : JSB_CB_RET NOVALUE, ! Pop LUB/ISB/RAB
92 0156 1 |                                       ! after popping back previous LUB/ISB/RAB, if any.
93 0157 1 |   FOR$$FREE_VM : NOVALUE,          ! Deallocate FAB
94 0158 1 |   FOR$$CLOSE_FILE : CALL_CCB;      ! Internal file close
95 0159 1 |
```

```

97 0160 1 GLOBAL ROUTINE FOR$CLOSE (          ! FORTRAN CLOSE statement
98 0161 1     KEYWD,                          ! keyword code - repeated arguments
99 0162 1     INFO)                          ! value of keyword (optional)
100 0163 1     =                              ! value is TRUE iff successful,
101 0164 1     ! FALSE if error and err= present.
102 0165 1
103 0166 1 ++
104 0167 1 ABSTRACT:
105 0168 1
106 0169 1     Interpret CLOSE statement keywords:
107 0170 1     attributes specified in the encoded keyword parameters.
108 0171 1     The keywords are UNIT=, DISPOSE=, and ERR=.
109 0172 1     DISPOSE= may have one of, 'SAVE', 'PRINT', or 'DELETE' values
110 0173 1     which overrides the DISPOSE= specification of the OPEN
111 0174 1     on that LUN. Call FOR$$CLOSE_FILE to RMS close the file.
112 0175 1     Flag the logical unit as closed.
113 0176 1     Deallocate LUB/ISB/RAB for this LUN
114 0177 1
115 0178 1 FORMAL PARAMETERS:
116 0179 1
117 0180 1 The following pair is repeated for each user specified keyword:
118 0181 1     KEYWD.rlu.v  Contains KEY<7:0>, ARGTYPE<15:8>, and
119 0182 1     INFO.rlu.v   possibly INFO<31:16>
120 0183 1     Optional informaion if need more
121 0184 1     than 16 bits.
122 0185 1
123 0186 1 IMPLICIT INPUTS:
124 0187 1
125 0188 1     FOR$$A_CUR_LUB  Current active LUB to be pushed down or 0
126 0189 1     if no LUB has an I/O statement in progress (usual).
127 0190 1     LUB$V_OPENED   1 if LUN is opened, 0 if already closed.
128 0191 1     LUB$W_IFI      RMS internal file id of file on this LUN.
129 0192 1     Needed as input to $CLOSE.
130 0193 1     LUB$V_SCRATCH  1 if this is a scratch file.
131 0194 1     LUB$V_READ_ONLY 1 if this file was specified READONLY in FORTRAN OPEN
132 0195 1
133 0196 1 IMPLICIT OUTPUTS:
134 0197 1
135 0198 1
136 0199 1 COMPLETIUN STATUS:
137 0200 1
138 0201 1     TRUE if success, FALSE if failure and ERR= keyword present
139 0202 1
140 0203 1 SIDE EFFECTS:
141 0204 1
142 0205 1     Deallocates LUB/ISB/RAB
143 0206 1     SIGNALs or SIGNAL_STOPs the following errors unless
144 0207 1     ERR= is present.
145 0208 1     SIGNAL_STOPs FOR$_CLOERR (28 = 'CLOSE ERROR')
146 0209 1     SIGNAL_STOPs FOR$_INCOPECLO (46 = 'INCONSISTENT OPEN/CLOSE
147 0210 1     STATEMENT SPECIFICATONS')
148 0211 1 --
149 0212 1
150 0213 2 BEGIN
151 0214 2
152 0215 2 GLOBAL REGISTER
153 0216 2     CCB = 11 : REF $FOR$CCB_DECL;

```

```
154 0217 2
155 0218 2
156 0219 2 BUILTIN
157 0220 2 ACTUALCOUNT;
158 0221 2
159 0222 2 MAP
160 0223 2 Use of formal argument list as a vector of blocks, each block
161 0224 2 is one longword.
162 0225 2
163 0226 2 KEYWD : BLOCKVECTOR [100, 1];
164 0227 2
165 0228 2 DCAL
166 0229 2 L_UNWIND_ACTION : VOLATILE, ! Unwind action code for handler
167 0230 2 CLOS : VOLATILE VECTOR [CLOS$K_KEY_MAX + 1], ! close parameter array
168 0231 2 NAM_DSC : DSC$DESCRIPTOR, ! string desc for ASCII
169 0232 2 DEF_DSC : DSC$DESCRIPTOR; ! string desc for ASCII
170 0233 2
171 0234 2 ENABLE ! Establish error handler to
172 0235 2 ! RESIGNAL or UNWIND if ERP= present
173 0236 2 FOR$$ERR_OPECLO (L_UNWIND_ACTION, CLOS); ! Pass unwind action and
174 0237 2
175 0238 2 ! pass CLOS array with ERR= flag in CLOS[OPEN$K_ERR]
176 0239 2
177 0240 2 Set cleanup action on UNWIND to no-operation (since LUB/ISB/RAB
178 0241 2 not pushed down yet)
179 0242 2
180 0243 2 L_UNWIND_ACTION = FOR$K_UNWINDNOP;
181 0244 2
182 0245 2 Scan user parameter list and copy in sorted order to CLOS
183 0246 2 array. SIGNAL_STOP FOR$ INVARGFOR (48='INVALID ARGUMENT TO FORTRAN I/O SYSTEM')
184 0247 2 after scanning all parameters to see if ERR= is present
185 0248 2 and setting CLOS[OPEN$K_ERR] accordingly to 1 or 0
186 0249 2
187 0250 2 FOR$$OPECLO_ARG (KEYWD, ACTUALCOUNT (), CLOS, CLOS$K_KEY_MAX, NAM_DSC, DEF_DSC, 0);
188 0251 2
189 0252 2 Allocate LUB/ISB/RAB for unit 0:99 if not already setup
190 0253 2 Push down active I/O if on another unit. If unit already has
191 0254 2 active I/O, SIGNAL_STOP FOR$ RECIO_OPE (40 = 'RECURSIVE I/O OPERATION')
192 0255 2 If logical unit outside legal range, SIGNAL_STOP FOR$ INVLOGUNI (32='INVALID LOGICAL UNIT NUMBER')
193 0256 2 Stores new LUB/ISB/RAB address in OTS Common OTSS$A_COR_LUB
194 0257 2 Finally indicate that UNWIND cleanup action is now to be to
195 0258 2 ret current LUB/ISB/RAB since it has now been sucessfully pushed.
196 0259 2 On return, CCB points to current control block.
197 0260 2
198 0261 2 FOR$$CB_PUSH (.CLOS [OPEN$K_UNIT], LUB$K_LUN_MIN);
199 0262 2 L_UNWIND_ACTION = FOR$K_UNWINDRET;
200 0263 2
201 0264 2 Check if file already CLOSED (or not OPEN)
202 0265 2
203 0266 2
204 0267 2 IF (.CCB [LUB$V_OPENED])
205 0268 2 THEN
206 0269 2 BEGIN
207 0270 2
208 0271 2 DISPOSE
209 0272 2
210 0273 2
```

```
211 0274 3 IF .CLOS [OPEN$K_DISPOSE] NEQ 0
212 0275 3 THEN
213 0276 4 BEGIN
214 0277 4
215 0278 4
216 0279 4 |* Turn off all dispose bits initially.
217 0280 4 |*
218 0281 4
219 0282 4 CCB [LUB$V_DELETE] = 0;
220 0283 4 CCB [LUB$V_PRINT] = 0;
221 0284 4 CCB [LUB$V_SUBMIT] = 0;
222 0285 4
223 0286 4 SELECT .CLOS [OPEN$K_DISPOSE] OF
224 0287 4 SET
225 0288 4
226 0289 4 [OPEN$K_DIS_SAV] :
227 0290 4 ! DISPOSE = 'SAVE'
228 0291 5 BEGIN
229 0292 5
230 0293 5 IF .CCB [LUB$V_SCRATCH] THEN FOR$$SIGNAL_STO (FOR$K_INCOPECLO);
231 0294 5
232 0295 4 END;
233 0296 4
234 0297 4 [OPEN$K_DIS_DEL, OPEN$K_DIS_PRDE, OPEN$K_DIS_SUDE] :
235 0298 4 ! DISPOSE = 'DELETE', 'PRINT/DELETE' or 'SUBMIT/DELETE'
236 0299 5 BEGIN
237 0300 5
238 0301 5 IF .CCB [LUB$V_READ_ONLY] THEN FOR$$SIGNAL_STO (FOR$K_INCOPECLO);
239 0302 5
240 0303 5 CCB [LUB$V_DELETE] = 1;
241 0304 4 END;
242 0305 4
243 0306 4 [OPEN$K_DIS_PRI, OPEN$K_DIS_PRDE] :
244 0307 4 ! DISPOSE = 'PRINT', 'PRINT/DELETE'
245 0308 5 BEGIN
246 0309 5
247 0310 5 IF .CCB [LUB$V_SCRATCH] THEN FOR$$SIGNAL_STO (FOR$K_INCOPECLO);
248 0311 5
249 0312 5 CCB [LUB$V_PRINT] = 1;
250 0313 4 END;
251 0314 4
252 0315 4 [OPEN$K_DIS_SUB, OPEN$K_DIS_SUDE] :
253 0316 4 ! DISPOSE = 'SUBMIT', 'SUBMIT/DELETE'
254 0317 4
255 0318 4 IF .CCB [LUB$V_SCRATCH]
256 0319 4 THEN
257 0320 4 FOR$$SIGNAL_STO (FOR$K_INCOPECLO)
258 0321 4 ELSE
259 0322 4 CCB [LUB$V_SUBMIT] = 1;
260 0323 4
261 0324 4 [OTHERWISE] :
262 0325 4 FOR$$SIGNAL_STO (FOR$K_INVARGFOR);
263 0326 4 YES;
264 0327 4
265 0328 3 END;
266 0329 3
267 0330 3 !*
```

```

268 0331 3 ! Call procedure FOR$$CLOSE_FILE to RMS close the file.
269 0332 3 !-
270 0333 3
271 0334 3     IF ( NOT FOR$$CLOSE_FILE () ) THEN FOR$$SIGNAL_STO (FOR$K_CLOERR);
272 0335 3
273 0336 3     END
274 0337 3 ELSE
275 0338 3 !+
276 0339 3 ! Set LUB$V_DEALLOC so FOR$$CB_POP will deallocate the rest of the LUN.
277 0340 3 !-
278 0341 3     BEGIN
279 0342 3     CCB [LUB$V_DEALLOC] = 1;
280 0343 3     END;
281 0344 3
282 0345 3 !+
283 0346 3 ! Pop back previous LUB or indicate that no I/O statement
284 0347 3 ! is currently active (OTSS$A_CUR_LUB = 0).
285 0348 3 ! In the latter case the deallocation will be performed right away.
286 0349 3 !-
287 0350 3     FOR$$CB_POP ();
288 0351 3
289 0352 3 !+
290 0353 3 ! Store success IOSTAT. If there was an error, the handler would
291 0354 3 ! do the store.
292 0355 3 !-
293 0356 3
294 0357 3 IF .CLOS [OPEN$K_IOSTAT] NEQ 0
295 0358 3 THEN
296 0359 3
297 0360 3     IF .CLOS [OPEN$K_IOSTAT_L]
298 0361 3     THEN
299 0362 3     .CLOS [OPEN$K_IOSTAT] = 0
300 0363 3 ELSE
301 0364 3     BEGIN
302 0365 3     LOCAL
303 0366 3     IOSTAT : REF BLOCK [, BYTE];
304 0367 3
305 0368 3     IOSTAT = .CLOS [OPEN$K_IOSTAT];
306 0369 3     IOSTAT [0, 0, 16, 0] = 0;
307 0370 3     ! Store one word
308 0371 3     END;
309 0372 3
310 0373 3 RETURN SSS_NORMAL;
311 0374 3 ! RETURN success from FOR$CLOSE
! END of FOR$CLOSE routine

```

```

.TITLE FOR$CLOSE FORTRAN CLOSE and default CLOSE
.IDENT \1-021\

.EXTRN FOR$$OPECLO_ARG
.EXTRN FOR$$SIGNAL_STO
.EXTRN FOR$$ERR_OPECLO
.EXTRN FOR$$CB_PUSH, FOR$$CB_POP
.EXTRN FOR$$FREE_VM, FOR$$CLOSE_FILE

.PSECT _FOR$CODE, NOWRT, SHR, PIC, 2

```


	03		53	D1	000B6	5\$:	CMPL	R3, #3	0306	
			05	13	000B9		BEQL	6\$		
	05		53	D1	000BB		CMPL	R3, #5		
			0F	12	000BE		BNEQ	8\$		
			54	D4	000C0	6\$:	CLRL	R4		
	05		62	E1	000C2		BBC	#5, (R2), 7\$	0310	
			2E	DD	000C6		PUSHL	#46		
	65		01	FB	000C8		CALLS	#1, FOR\$\$SIGNAL_STO		
	62	80	8F	88	000CB	7\$:	BISB2	#128, (R2)	0312	
	04		53	D1	000CF	8\$:	CMPL	R3, #4	0315	
			05	13	000D2		BEQL	9\$		
	06		53	D1	000D4		CMPL	R3, #6		
			11	12	000D7		BNEQ	11\$		
	07		54	D4	000D9	9\$:	CLRL	R4		
			05	E1	000DB		BBC	#5, (R2), 10\$	0318	
			2E	DD	000DF		PUSHL	#46	0320	
	65		01	FB	000E1		CALLS	#1, FOR\$\$SIGNAL_STO		
			04	11	000E4		BRB	11\$		
	FF	AB	20	88	000E6	10\$:	BISB2	#32, -1(CCB)	0322	
		05	54	E9	000EA	11\$:	BLBC	R4, 12\$	0324	
			30	DD	000ED		PUSHL	#48	0325	
	00000000G	65	01	FB	000EF		CALLS	#1, FOR\$\$SIGNAL_STO		
		00	00	FB	000F2	12\$:	CALLS	#0, FOR\$\$CLOSE_FILE	0334	
		0B	50	E8	000F9		BLBS	R0, 14\$		
			1C	DD	000FC		PUSHL	#28		
		65	01	FB	000FE		CALLS	#1, FOR\$\$SIGNAL_STO		
			04	11	00101		BRB	14\$	0267	
	FF	AB	10	88	00103	13\$:	BISB2	#16, -1(CCB)	0342	
		0000U000G	00	16	00107	14\$:	JSB	FOR\$\$CB_POP	0350	
			68	AE	D5	0010D	TSTL	CLOS+88	0357	
			0F	13	00110		BEQL	16\$		
		05	74	AE	E9	00112	BLBC	CLOS+100, 15\$	0360	
			68	BE	D4	00116	CLRL	@CLOS+88	0362	
			06	11	00119		BRB	16\$		
		50	68	AE	D0	0011B	15\$:	MOVL	CLOS+88, IOSTAT	0369
			60	B4	0011F		CLRW	(IOSTAT)	0370	
		50	01	D0	00121	16\$:	MOVL	#1, R0	0373	
			04	00124			RET		0374	
			0000	00125	17\$:	.WORD	Save nothing		0213	
		50	08	AC	D0	00127	MOVL	8(AP), R0		
		50	04	A0	D0	0012B	MOVL	4(R0), R0		
			90	A0	9F	0012F	PUSHAB	CLOS		
			FC	A0	9F	00132	PUSHAB	L_UNWIND_ACTION		
			02	DD	00135		PUSHL	#2		
			5E	DD	00137		PUSHL	SP		
	00000000G	7E	04	AC	7D	00139	MOVQ	4(AP), -(SP)		
		00	03	FB	0013D		CALLS	#3, FOR\$\$ERR_OPECLO		
			04	00144			RET			

; Routine Size: 325 bytes, Routine Base: _FOR\$CODE + 0000

```

: 312      0375  1
: 313      0376  1 END
: 314      0377  1
: 315      0378  0 ELUDOM

```

! END of FOR\$CLOSE module

PSECT SUMMARY

```
:  
: Name Bytes Attributes  
: _FOR$CODE 325 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 1 0 581 00:01.0  
: $255$DUA28:[FORRTL.OBJ]FORLIB.L32;1 711 198 27 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$:FORCLOSE/OBJ=OBJ$:FORCLOSE MSRC$:FORCLOSE/UPDATE=(ENH$:FORCLOSE)  
: Size: 325 code + 0 data bytes  
: Run Time: 00:08.1  
: Elapsed Time: 00:27.7  
: Lines/CPU Min: 2793  
: Lexemes/CPU-Min: 9354  
: Memory Used: 122 pages  
: Compilation Complete
```


