


```

FFFFFFFFF 000000 RRRRRRRR CCCCCCCC LL 000000 SSSSSSSS EEEEEEEEE FFFFFFFF
FFFFFFFFF 000000 RRRRRRRR CCCCCCCC LL 000000 SSSSSSSS EEEEEEEEE FFFFFFFF
FF 00 00 RR RR CC CC LL 00 00 SS SS EE EE FF FF
FF 00 00 RR RR CC CC LL 00 00 SS SS EE EE FF FF
FF 00 00 RR RR CC CC LL 00 00 SS SS EE EE FF FF
FF 00 00 RR RR CC CC LL 00 00 SS SS EE EE FF FF
FFFFFFFF 00 00 RRRRRRRR CCCCCCCC LL 00 00 SSSSSS EEEEEEE FFFFFFFF
FFFFFFFF 00 00 RRRRRRRR CCCCCCCC LL 00 00 SSSSSS EEEEEEE FFFFFFFF
FF 00 00 RR RR CC CC LL 00 00 SS SS EE EE FF FF
FF 00 00 RR RR CC CC LL 00 00 SS SS EE EE FF FF
FF 00 00 RR RR CC CC LL 00 00 SS SS EE EE FF FF
FF 000000 RR RR CCCCCCCC LLLLLLLLLL 000000 SSSSSSSS EEEEEEEEE FF
FF 000000 RR RR CCCCCCCC LLLLLLLLLL 000000 SSSSSSSS EEEEEEEEE FF

```

```

LL 111111 SSSSSSSS
LL 111111 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 II SS
LLLLLLLLLL 111111 SSSSSSSS
LLLLLLLLLL 111111 SSSSSSSS

```

```
1 0001 0 MODULE FOR$$CLOSE_FILE (%TITLE, 'CLOSE FILE'  
2 0002 0 IDENT = '1-002' ! File: FORCLOSEF.B32 Edit: SBL1002  
3 0003 0 ) =  
4 0004 1 BEGIN  
5 0005 1  
6 0006 1 *****  
7 0007 1 *  
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *  
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *  
10 0010 1 * ALL RIGHTS RESERVED. *  
11 0011 1 * *  
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *  
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *  
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *  
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *  
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *  
17 0017 1 * TRANSFERRED. *  
18 0018 1 * *  
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *  
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *  
21 0021 1 * CORPORATION. *  
22 0022 1 * *  
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *  
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *  
25 0025 1 * *  
26 0026 1 * *  
27 0027 1 *****  
28 0028 1  
29 0029 1 ++  
30 0030 1 FACILITY: FORTRAN Language Support  
31 0031 1  
32 0032 1 ABSTRACT:  
33 0033 1  
34 0034 1 This module issues an RMS close to a file on a LUN.  
35 0035 1  
36 0036 1 ENVIRONMENT: User access level; re-entrant, AST level or not.  
37 0037 1  
38 0038 1 AUTHOR: Steven B. Lionel, 30-Sep-1982  
39 0039 1  
40 0040 1 MODIFIED BY:  
41 0041 1  
42 0042 1 1-001 - Adapted from OT$$CLOSE_FILE, version 1-012. SBL 30-Sep-1982  
43 0043 1 1-002 - Use FAB and NAM in CCB. SBL 20-Jan-1983  
44 0044 1 --  
45 0045 1
```

```

: 47      0046 1  !
: 48      0047 1  ! PROLOGUE FILE:
: 49      0048 1  !
: 50      0049 1  !
: 51      0050 1  REQUIRE 'RTLIN:FORPROLOG';           ! FORTRAN definitions
: 52      0116 1  !
: 53      0117 1  !
: 54      0118 1  ! TABLE OF CONTENTS:
: 55      0119 1  !
: 56      0120 1  !
: 57      0121 1  FORWARD ROUTINE
: 58      0122 1  FOR$$CLOSE_FILE : CALL_CCB;           ! Internal file close
: 59      0123 1  !
: 60      0124 1  !
: 61      0125 1  ! EQUATED SYMBOLS:
: 62      0126 1  !
: 63      0127 1  !     NONE
: 64      0128 1  !
: 65      0129 1  ! OWN STORAGE:
: 66      0130 1  !
: 67      0131 1  !     NONE
: 68      0132 1  !
: 69      0133 1  ! EXTERNAL REFERENCES:
: 70      0134 1  !
: 71      0135 1  !     NONE
: 72      0136 1  !
: 73      0137 1  !

```

```
75      0138 1 GLOBAL ROUTINE FOR$$CLOSE_FILE          ! Internal file closer
76      0139 1      : CALL_CCB =
77      0140 1
78      0141 1 !++
79      0142 1 FUNCTIONAL DESCRIPTION:
80      0143 1
81      0144 1      Do an RMS CLOSE or DISCONNECT of a LUN. This includes handling
82      0145 1      any of the disposition flags in the LUB, whether set by OPEN or CLOSE.
83      0146 1
84      0147 1 FORMAL PARAMETERS:
85      0148 1
86      0149 1      NONE
87      0150 1
88      0151 1 IMPLICIT INPUTS:
89      0152 1
90      0153 1      Various fields from the LUB, pointed to by CCB.
91      0154 1
92      0155 1 IMPLICIT OUTPUTS
93      0156 1
94      0157 1      Various fields in the LUB and RAB.
95      0158 1
96      0159 1 ROUTINE VALUE:
97      0160 1
98      0161 1      The same as RMS CLOSE. The caller generally only tests the
99      0162 1      low-order bit of the completion code: if 1 the close succeeded,
100     0163 1      if 0 it failed.
101     0164 1
102     0165 1 SIDE EFFECTS:
103     0166 1
104     0167 1      CLOSEs the LUN, and marks it for deallocation.
105     0168 1
106     0169 1 !--
107     0170 1
108     0171 2 BEGIN
109     0172 2
110     0173 2 EXTERNAL REGISTER
111     0174 2      CCB : REF $FOR$CCB_DECL;
112     0175 2
113     0176 2 LOCAL
114     0177 2      CLOSE_RESULT;          ! RMS result of $CLOSE
115     0178 2
116     0179 2 BIND
117     0180 2      FAB = CCB: REF $FOR$FAB_CCB_STRUCT,
118     0181 2      NAM = CCB: REF $FOR$NAM_CCB_STRUCT;
119     0182 2
120     0183 2 !+
121     0184 2 !- If V_DELETE is on in the LUB, set the DLT bit in the FAB.
122     0185 2 !-
123     0186 2
124     0187 2 IF .CCB [LUB$V_DELETE] THEN FAB [FAB$V_DLT] = 1;
125     0188 2
126     0189 2 !+
127     0190 2 !- If V_PRINT is on in the LUB, set the SPL bit in the FAB.
128     0191 2 !-
129     0192 2
130     0193 2 IF .CCB [LUB$V_PRINT] THEN FAB [FAB$V_SPL] = 1;
131     0194 2
```

```
132      0195      2      !+
133      0196      2      !- If V_SUBMIT is on in the LUB, set the SCF bit in the FAB.
134      0197      2      !-
135      0198      2
136      0199      2      IF .CCB [LUB$V_SUBMIT] THEN FAB [FAB$V_SCF] = 1;
137      0200      2
138      0201      2      !+
139      0202      2      !- If the file was not successfully OPENed, zero the XAB chain pointer.
140      0203      2      !-
141      0204      2
142      0205      2      IF NOT .CCB [LUB$V_OPENED]
143      0206      2      THEN
144      0207      2          FAB [FAB$L_XAB] = 0;
145      0208      2
146      0209      2      !+
147      0210      2      !- Now do the RMS close if IFI is not zero.
148      0211      2      !-
149      0212      2
150      0213      2      IF .FAB [FAB$W_IFI] NEQ 0
151      0214      2      THEN
152      0215      2          BEGIN
153      0216      2
154      0217      2      !+
155      0218      2      !- Do a $CLOSE and put the error codes into the RAB so that
156      0219      2      !- our caller has a better idea of what the error was.
157      0220      2
158      0221      2
159      0222      2          CLOSE_RESULT = $CLOSE (FAB = FAB [0,0,0,0]);
160      0223      2          IF NOT .CLOSE_RESULT
161      0224      2          THEN
162      0225      2              BEGIN
163      0226      2                  CCB [RAB$L_STS] = .CLOSE_RESULT;
164      0227      2                  CCB [RAB$L_STV] = .FAB [FAB$L_STV];
165      0228      2                  END;
166      0229      2              END
167      0230      2          ELSE
168      0231      2              CLOSE_RESULT = 1;
169      0232      2
170      0233      2      !+
171      0234      2      !- Indicate that the LUN has been closed, so further I/O to it will
172      0235      2      !- fail.
173      0236      2      !-
174      0237      2
175      0238      2      CCB [LUB$V_OPENED] = 0;
176      0239      2
177      0240      2      !+
178      0241      2      !- Flag FOR$$CB_POP that it is to deallocate the virtual storage
179      0242      2      !- occupied by this LUN. This bit also prevents OPEN from opening
180      0243      2      !- this LUN. OPEN will only see it if FOR$$CB_POP does not deallocate
181      0244      2      !- the LUB, which will happen only if there is recursive I/O active on
182      0245      2      !- the LUN. When all of the recursive I/O has failed then the LUN
183      0246      2      !- can be opened again.
184      0247      2      !-
185      0248      2
186      0249      2      CCB [LUB$V_DEALLOC] = 1;
187      0250      2
188      0251      2      !+
```

```

: 189      0252 2      ! Return the RMS status resulting from the CLOSE.
: 190      0253 2      !-
: 191      0254 2
: 192      0255 2      RETURN (.CLOSE_RESULT);
: 193      0256 1      END;
    
```

```

                                .TITLE FOR$$CLOSE_FILE CLOSE FILE
                                .IDENT  \1-002\
                                .EXTRN  SYSS$CLOSE
                                .PSECT  _FOR$CODE,NOWRT, SHR, PIC,2

                                .ENTRY  FOR$$CLOSE_FILE, Save nothing
05      FC  AB      0000 00000      BBC      #6, -4(CCB), 1$      : 0138
        49  AB      80  8F  88 00002      BICB2     #128, 73(CCB)      : 0187
        FC      FC  AB  95 0000C 1$      TSTB      -4(CCB)      : 0193
                                BGEQ      2$
                                BISB2     #32, 73(CCB)
05      FF  AB      20  88 00011      BBC      #5, -1(CCB), 3$      : 0199
        49  AB      40  8F  88 0001A 2$      BISB2     #64, 73(CCB)
        03      FC  AB  E8 0001F 3$      BLBS      -4(CCB), 4$
        03      68  AB  D4 00023      CLRL      104(CCB)
        03      46  AB  B5 00026 4$      TSTW      70(CCB)
        03      18  13 00029      BEQL      5$
        03      44  AB  9F 0002B      PUSHAB    68(CCB)      : 0222
                                CALLS     #1, SYSS$CLOSE
                                BLBS      CLOSE_RESULT, 6$      : 0223
                                MOVL      CLOSE_RESULT, 8(CCB)      : 0226
                                MOVL      80(CCB), 12(CCB)      : 0227
                                BRB      6$      : 0213
                                MOVL      #1, CLOSE_RESULT      : 0231
                                BICB2     #1, -4(CCB)      : 0238
                                BISB2     #16, -1(CCB)      : 0249
                                RET      : 0256
                                04 0004E
                                01  01  D0 00043 5$
                                01  8A 00046 6$
                                10  88 0004A
                                04 0004E
                                00000000G  00
                                08  AB      50  50  D0 00038
                                OC  AB      50  AB  D0 0003C
                                03  11 00041
                                01  D0 00043 5$
                                FC  AB      01  8A 00046 6$
                                FF  AB      10  88 0004A
                                04 0004E
    
```

: Routine Size: 79 bytes, Routine Base: _FOR\$CODE + 0000

```

: 194      0257 1
: 195      0258 1 END
: 196      0259 1
: 197      0260 0 ELUDOM
    
```

! END of FOR\$\$CLOSE_FILE module

PSECT SUMMARY

Name	Bytes	Attributes
_FOR\$CODE	79	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	12	0	581	00:01.0
_\$255\$DUA28:[FORRTL.OBJ]FORLIB.L32;1	711	180	25	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\$:FORCLOSEF/OBJ=OBJ\$:FORCLOSEF MSRC\$:FORCLOSEF/UPDATE=(ENH\$:FORCLOSEF)

: Size: 79 code + 0 data bytes
: Run Time: 00:05.3
: Elapsed Time: 00:23.9
: Lines/CPU Min: 2921
: Lexemes/CPU-Min: 10752
: Memory Used: 88 pages
: Compilation Complete

