

[illegible]

• • • •

```

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                      II                      SS
LLLLLLLLLLLLLLLL      IIIIII                      SSSSSSSS
LLLLLLLLLLLLLLLL      IIIIII                      SSSSSSSS

```



```
0001 0 MODULE OTSS$CLOSE_FILE (
0002 0 IDENT = '1-012'
0003 0 ) =
0004 1 BEGIN
0005 1
0006 1 *****
0007 1 *
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 * ALL RIGHTS RESERVED.
0011 1 *
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 * TRANSFERRED.
0018 1 *
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 * CORPORATION.
0022 1 *
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *
0027 1 *****
0028 1
0029 1 ++
0030 1 FACILITY: Language Intependent Library - Close File
0031 1
0032 1 ABSTRACT:
0033 1
0034 1 This module issues an RMS close to a file on a LUN.
0035 1
0036 1 ENVIRONMENT: User access level; re-entrant, AST level or not.
0037 1
0038 1 AUTHOR: John Sauter, CREATION DATE: 23-JAN-1979
0039 1
0040 1 MODIFIED BY:
0041 1
0042 1 1-001 - Original from FORCLOSE. JBS 23-JAN-1979
0043 1 1-002 - Set LUB$V_DEALLOC so that the LUN's virtual memory will be
0044 1 released. JBS 24-JAN-1979
0045 1 1-003 - Call OTSS$PURGE IOBU before closing to purge I/O buffers.
0046 1 JBS 24-JAN-1979
0047 1 1-004 - Put two dollar signs on non-user entry points. JBS 26-JAN-1979
0048 1 1-005 - Change purge external name to OTSS$PUR IO CLO. JBS 07-MAR-1979
0049 1 1-006 - Call through LUB$A_CLOSE if non-zero. JBS 04-APR-1979
0050 1 1-007 - Implement LUB$V_SUBMIT. JBS (via SBL) 20-APR-1979
0051 1 1-008 - Don't do a $CLOSE if FAB$W_IFI is zero and LUB$V_USEROPEN
0052 1 is set. SPR 11-25235 SBL 31-July-1979
0053 1 1-009 - Don't call OTSS$PUR IO CLO. Our caller must now purge I/O
0054 1 buffers. JBS 20-AUG-1979
0055 1 1-010 - If multi-stream connect, do a $DISCONNECT instead of a $CLOSE.
0056 1 JBS 28-SEP-1979
0057 1 1-011 - If $CLOSE, put STS and STV fields in the RAB for better
```

OTSS\$CLOSE\_FILE  
1-012

J 9  
16-Sep-1984 01:24:56  
14-Sep-1984 12:39:40

VAX-11 Bliss-32 V4.0-742  
[LIBRTL.SRC]OTSCLOSEF.B32;1

Page 2  
(1)

```
: 58      0058 1 | error reporting. SBL 8-Oct-1979
: 59      0059 1 | 1-012 - Don't RMS close the file if Ifl is zero. Ignore USEROPEN. SBL 6-Dec-1979
: 60      0060 1 | --
: 61      0061 1 |
: 62      0062 1 | <BLF/PAGE>
```



```

64      0063 1 |
65      0064 1 | SWITCHES:
66      0065 1 |
67      0066 1 |
68      0067 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
69      0068 1 |
70      0069 1 |
71      0070 1 | LINKAGES:
72      0071 1 |
73      0072 1 |
74      0073 1 | REQUIRE 'RTLIN:OTSLNK';           ! define all linkages
75      0502 1 |
76      0503 1 |
77      0504 1 | TABLE OF CONTENTS:
78      0505 1 |
79      0506 1 |
80      0507 1 | FORWARD ROUTINE
81      0508 1 |     OTSS$CLOSE_FILE : CALL_CCB;   ! Internal file close
82      0509 1 |
83      0510 1 |
84      0511 1 | INCLUDE FILES:
85      0512 1 |
86      0513 1 |
87      0514 1 | REQUIRE 'RTLML:OTSLUB';           ! logical unit block definitions
88      0654 1 |
89      0655 1 | REQUIRE 'RTLIN:OTSMAC';           ! macros
90      0849 1 |
91      0850 1 | REQUIRE 'RTLIN:RTLPSECT';         ! Define DECLARE_PSECTS macro
92      0945 1 |
93      0946 1 | LIBRARY 'RTLSTARLE';             ! STARLET library for macros and symbols
94      0947 1 |
95      0948 1 |
96      0949 1 | MACROS:
97      0950 1 |
98      0951 1 |     None
99      0952 1 |
100     0953 1 | EQUATED SYMBOLS:
101     0954 1 |
102     0955 1 |     None
103     0956 1 |
104     0957 1 | PSECT DECLARATIONS:
105     0958 1 |
106     0959 1 | DECLARE_PSECTS (OTS);             ! declare PSECTs for OTSS$ facility
107     0960 1 |
108     0961 1 | OWN STORAGE:
109     0962 1 |
110     0963 1 |     None
111     0964 1 |
112     0965 1 | EXTERNAL REFERENCES:
113     0966 1 |
114     0967 1 |     NONE
115     0968 1 |
116     0969 1 |

```

```

: 118 0970 1 GLOBAL ROUTINE OTSS$CLOSE_FILE          ! Internal file closer
: 119 0971 1 : CALL_CCB =
: 120 0972 1
: 121 0973 1 ++
: 122 0974 1 FUNCTIONAL DESCRIPTION:
: 123 0975 1
: 124 0976 1 Do an RMS CLOSE or DISCONNECT of a LUN. This includes handling
: 125 0977 1 any of the disposition flags in the LUB, whether set by OPEN or CLOSE.
: 126 0978 1
: 127 0979 1 FORMAL PARAMETERS:
: 128 0980 1
: 129 0981 1 NONE
: 130 0982 1
: 131 0983 1 IMPLICIT INPUTS:
: 132 0984 1
: 133 0985 1 Various fields from the LUB, pointed to by CCB.
: 134 0986 1
: 135 0987 1 IMPLICIT OUTPUTS
: 136 0988 1
: 137 0989 1 Various fields in the LUB and RAB.
: 138 0990 1
: 139 0991 1 ROUTINE VALUE:
: 140 0992 1 COMPLETION CODES:
: 141 0993 1
: 142 0994 1 The same as RMS CLOSE. The caller generally only tests the
: 143 0995 1 low-order bit of the completion code: if 1 the close succeeded,
: 144 0996 1 if 0 it failed.
: 145 0997 1
: 146 0998 1 SIDE EFFECTS:
: 147 0999 1
: 148 1000 1 CLOSEs the LUN, and marks it for deallocation.
: 149 1001 1
: 150 1002 1 --
: 151 1003 1
: 152 1004 2 BEGIN
: 153 1005 2
: 154 1006 2 EXTERNAL REGISTER
: 155 1007 2 CCB : REF BLOCK [, BYTE];
: 156 1008 2
: 157 1009 2 LOCAL
: 158 1010 2 NAM_BLOCK : BLOCK [NAM$C_BLN, BYTE], ! local name block for delete
: 159 1011 2 FAB_BLOCK : BLOCK [FAB$C_BLN, BYTE], ! allocate LOCAL FAB
: 160 1012 2 FAB : REF BLOCK [FAB$C_BLN, BYTE], ! declare FAB base pointer
: 161 1013 2 CLOSE_RESULT; ! RMS result of $CLOSE
: 162 1014 2
: 163 1015 2 +
: 164 1016 2 Initialize FAB
: 165 1017 2 Copy FAB address into the LUB for proper error status reporting.
: 166 1018 2
: 167 1019 2 FAB = FAB_BLOCK;
: 168 1020 2 CH$FILL (0, FAB$C_BLN, .FAB);
: 169 1021 2 FAB [FAB$B_BID] = FAB$C_BID;
: 170 1022 2 FAB [FAB$B_BLN] = FAB$C_BLN;
: 171 1023 2 CCB [LUB$A_FAB] = .FAB;
: 172 1024 2 +
: 173 1025 2 Copy the directory ID into the NAM block and link into FAB.
: 174 1026 2 Set pointer to resultant name string (in case this is print

```



```

175 1027 2  ! so spooler will use in banner).
176 1028 2  !-
177 1029 2  CH$FILL (0, NAM$C_BLN, NAM_BLOCK);
178 1030 2  NAM_BLOCK [NAM$B_BID] = NAM$C_BID;
179 1031 2  NAM_BLOCK [NAM$B_BLN] = NAM$C_BLN;
180 1032 2  CH$MOVE (NAM$S_DID, CCB [LUB$B_DID], NAM_BLOCK [NAM$W_DID]);
181 1033 2  FAB [FAB$S_NAM] = NAM_BLOCK;
182 1034 2  FAB [FAB$V_NAM] = 1;
183 1035 2  NAM_BLOCK [NAM$S_RSA] = .CCB [LUB$A_RSN];
184 1036 2  NAM_BLOCK [NAM$B_RSL] = .CCB [LUB$B_RSL];
185 1037 2  !+
186 1038 2  !- If V_DELETE is on in the LUB, set the DLT bit in the FAB.
187 1039 2  !-
188 1040 2  IF .CCB [LUB$V_DELETE] THEN FAB [FAB$V_DLT] = 1;
189 1041 2  !+
190 1042 2  !- If V_PRINT is on in the LUB, set the SPL bit in the FAB.
191 1043 2  !-
192 1044 2  IF .CCB [LUB$V_PRINT] THEN FAB [FAB$V_SPL] = 1;
193 1045 2  !+
194 1046 2  !- If V_SUBMIT is on in the LUB, set the SCF bit in the FAB.
195 1047 2  !-
196 1048 2  IF .CCB [LUB$V_SUBMIT] THEN FAB [FAB$V_SCF] = 1;
197 1049 2  !+
198 1050 2  !- Copy remembered RMS internal file id from LUB to FAB.
199 1051 2  !-
200 1052 2  FAB [FAB$W_IFI] = .CCB [LUB$W_IFI];
201 1053 2  !+
202 1054 2  !- Do the close cleanup, if any is provided.
203 1055 2  !-
204 1056 2  IF (.CCB [LUB$A_CLOSE] NEQ 0) THEN CALL_CCB (.CCB [LUB$A_CLOSE]);
205 1057 2  !+
206 1058 2  !- Now do the RMS close if IFI is not zero.
207 1059 2  !-
208 1060 2  IF .FAB [FAB$W_IFI] NEQ 0
209 1061 2  THEN
210 1062 2  BEGIN
211 1063 2  !+
212 1064 2  !- If the file is just a stream from another file, do a $DISCONNECT.
213 1065 2  !- Otherwise, do a $CLOSE and put the error codes into the RAB so that
214 1066 2  !- our caller has a better idea of what the error was.
215 1067 2  !-
216 1068 2  IF (.CCB [LUB$V_M_STREAM])
217 1069 2  THEN
218 1070 2  CLOSE_RESULT = $DISCONNECT (RAB = .CCB)
219 1071 2  ELSE
220 1072 2  BEGIN
221 1073 2  CLOSE_RESULT = $CLOSE (FAB = .FAB);
222 1074 2  !+
223 1075 2  !-
224 1076 2  !-
225 1077 2  !-
226 1078 2  !-
227 1079 2  !-
228 1080 2  !-
229 1081 2  !-
230 1082 2  !-
231 1083 2  !-

```

```

232      1084  4      IF NOT .CLOSE_RESULT
233      1085  4      THEN
234      1086  5      BEGIN
235      1087  5          CCB [RAB$S_STS] = .CLOSE_RESULT;
236      1088  5          CCB [RAB$S_STV] = .FAB [FAB$S_STV];
237      1089  4      END;
238      1090  4      END
239      1091  4      ELSE
240      1092  3      END
241      1093  2      CLOSE_RESULT = 1;
242      1094  2
243      1095  2
244      1096  2      +
245      1097  2      - Indicate that the FAB error status is not meaningful anymore.
246      1098  2      -
247      1099  2      CCB [LUB$A_FAB] = 0;
248      1100  2      +
249      1101  2      - Indicate that the LUN has been closed, so further I/O to it will
250      1102  2      fail.
251      1103  2      -
252      1104  2      CCB [LUB$V_OPENED] = 0;
253      1105  2      +
254      1106  2      - Flag OTSS$POP CCB that it is to deallocate the virtual storage
255      1107  2      occupied by this LUN. This bit also prevents OPEN from opening
256      1108  2      this LUN. OPEN will only see it if OTSS$POP CCB does not deallocate
257      1109  2      the LUB, which will happen only if there is recursive I/O active on
258      1110  2      the LUN. When all of the recursive I/O has failed then the LUN
259      1111  2      can be opened again.
260      1112  2      -
261      1113  2      CCB [LUB$V_DEALLOC] = 1;
262      1114  2      +
263      1115  2      - Return the RMS status resulting from the CLOSE.
264      1116  2      -
265      1117  2      RETURN (.CLOSE_RESULT);
266      1118  1      END;

```

.TITLE OTSS\$CLOSE\_FILE  
.IDENT \1-012\

.EXTRN SYS\$DISCONNECT, SYS\$CLOSE

.PSECT \_OTSS\$CODE, NOWRT, SHR, PIC, 2

.ENTRY OTSS\$CLOSE\_FILE, Save R2, R3, R4, R5, R6

|       |                              |      |
|-------|------------------------------|------|
| MOVAB | -176(SP), SP                 | 0970 |
| MOVAB | FAB_BLOCK, FAB               | 1019 |
| MOVCS | #0, (SP), #0, #80, (FAB)     | 1020 |
| MOVW  | #20483, (FAB)                | 1021 |
| MOVL  | FAB, -24(CCB)                | 1023 |
| MOVCS | #0, (SP), #0, #96, NAM_BLOCK | 1029 |
| MOVW  | #24578, NAM_BLOCK            | 1030 |
| MOVCS | #6, -16(CCB), NAM_BLOCK+42   | 1032 |
| MOVAB | NAM_BLOCK, 40(FAB)           | 1033 |
| BISB2 | #1, -7(FAB)                  | 1034 |

|      |    |    |    |      |      |    |       |
|------|----|----|----|------|------|----|-------|
| 0050 | 8F | 00 | 5E | FF50 | CE   | 9E | 00000 |
|      |    |    | 56 |      | 6E   | 9E | 00002 |
|      |    |    | 6E |      | 00   | 2C | 0000A |
|      |    |    | 66 |      | 66   |    | 00011 |
|      |    |    | 66 | 5003 | 8F   | B0 | 00012 |
|      |    |    | E8 |      | 56   | D0 | 00017 |
| 0060 | 8F | 00 | 6E |      | 00   | 2C | 0001B |
|      |    |    |    | 50   | AE   |    | 00022 |
|      |    |    | 50 | AE   | 6002 | 8F | B0    |
|      |    | 7A | AE | AE   | 06   | 28 | 0002A |
|      |    |    | F0 | AB   | 06   | 28 | 0002A |
|      |    |    | 28 | A6   | AE   | 9E | 00030 |
|      |    |    | 07 | A6   | 01   | 88 | 00035 |



|           |    |    |    |    |    |       |       |                      |      |
|-----------|----|----|----|----|----|-------|-------|----------------------|------|
| 05        | 54 | AE | F8 | AB | D0 | 00039 | MOVL  | -8(CCB), NAM_BLOCK+4 | 1035 |
|           | 53 | AE | F7 | AB | 90 | 0003E | MOVB  | -9(CCB), NAM_BLOCK+3 | 1036 |
|           | FC | AB |    | 06 | E1 | 00043 | BBC   | #6, -4(CCB), -1\$    | 1041 |
|           | 05 | A6 | 80 | 8F | 88 | 00048 | BISB2 | #128, 5(FAB)         | 1047 |
|           |    |    | FC | AB | 95 | 0004D | TSTB  | -4(CCB)              | 1047 |
|           |    |    |    | 04 | 18 | 00050 | BGEQ  | 2\$                  | 1047 |
| 05        | 05 | A6 |    | 20 | 88 | 00052 | BISB2 | #32, 5(FAB)          | 1053 |
|           | FF | AB |    | 05 | E1 | 00056 | BBC   | #5, -1(CCB), 3\$     | 1053 |
|           | 05 | A6 | 40 | 8F | 88 | 00058 | BISB2 | #64, 5(FAB)          | 1058 |
|           | 02 | A6 | D0 | AB | B0 | 00060 | MOVW  | -48(CCB), 2(FAB)     | 1063 |
|           |    |    | A4 | AB | D5 | 00065 | TSTL  | -92(CCB)             | 1063 |
|           |    |    |    | 04 | 13 | 00068 | BEQL  | 4\$                  | 1063 |
|           | A4 | BB |    | 00 | FB | 0006A | CALLS | #0, a-92(CCB)        | 1069 |
|           |    |    | 02 | A6 | B5 | 0006E | TSTW  | 2(FAB)               | 1069 |
| 0B        | FF | AB |    | 27 | 13 | 00071 | BEQL  | 6\$                  | 1078 |
|           |    |    |    | 02 | E1 | 00073 | BBC   | #2, -1(CCB), 5\$     | 1080 |
| 00000000G | 00 |    |    | 5B | DD | 00078 | PUSHL | CCB                  | 1080 |
|           |    |    |    | 01 | FB | 0007A | CALLS | #1, SYSS\$DISCONNECT | 1083 |
|           |    |    |    | 1A | 11 | 00081 | BRB   | 7\$                  | 1083 |
| 00000000G | 00 |    |    | 56 | DD | 00083 | PUSHL | FAB                  | 1083 |
|           | 0E |    |    | 01 | FB | 00085 | CALLS | #1, SYSS\$CLOSE      | 1084 |
|           | 08 | AB |    | 50 | E8 | 0008C | BLBS  | CLOSE_RESULT, 7\$    | 1087 |
|           | 0C | AB | 0C | 50 | D0 | 0008F | MOVL  | CLOSE_RESULT, 8(CCB) | 1088 |
|           |    |    |    | A6 | D0 | 00093 | MOVL  | 12(FAB), 12(CCB)     | 1071 |
|           |    |    |    | 03 | 11 | 00098 | BRB   | 7\$                  | 1094 |
|           | 50 |    |    | 01 | D0 | 0009A | MOVL  | #1, CLOSE_RESULT     | 1099 |
|           |    |    | E8 | AB | D4 | 0009D | CLRL  | -24(CCB)             | 1104 |
|           | FC | AB |    | 01 | 8A | 000A0 | BICB2 | #1, -4(CCB)          | 1113 |
|           | FF | AB |    | 10 | 88 | 000A4 | BISB2 | #16, -1(CCB)         | 1118 |
|           |    |    |    | 04 | 00 | 000A8 | RET   |                      | 1118 |

; Routine Size: 169 bytes, Routine Base: \_OTSS\$CODE + C000

: 267 1119 1  
: 268 1120 1 END  
: 269 1121 1  
: 270 1122 0 ELUDOM

! END of OTSS\$CLOSE\_FILE module

# PSECT SUMMARY

| Name        | Bytes | Attributes                                              |
|-------------|-------|---------------------------------------------------------|
| _OTSS\$CODE | 169   | NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2) |

# Library Statistics

| File | Symbols |                | Pages Mapped | Processing Time |
|------|---------|----------------|--------------|-----------------|
|      | Total   | Loaded Percent |              |                 |

OTSS\$CLOSE\_FILE  
1-012

C 10  
16-Sep-1984 01:24:56  
14-Sep-1984 12:39:40

VAX-11 Bliss-32 V4.0-742  
[LIBRTL.SRC]OTSCLOSEF.B32;1

Page 8  
(3)

; \_\$255\$DUA28:[SYSLIB]STARLET.L32;1 9776 26 0 581 00:00.8

COMMAND QUALIFIERS

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

; Size: 169 code + 0 data bytes  
; Run Time: 00:06.7  
; Elapsed Time: 00:27.6  
; Lines/CPU Min: 10002  
; Lexemes/CPU-Min: 58038  
; Memory Used: 134 pages  
; Compilation Complete



0211 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

