



```

DDDDDDDD      EEEEEEEEEE      AAAAAA      CCCCCCCC      CCCCCCCC      SSSSSSSS
DDDDDDDD      EEEEEEEEEE      AAAAAA      CCCCCCCC      CCCCCCCC      SSSSSSSS
DD      DD      EE      AA      AA      CC      CC      CC      SS
DD      DD      EE      AA      AA      CC      CC      CC      SS
DD      DD      EE      AA      AA      CC      CC      CC      SS
DD      DD      EE      AA      AA      CC      CC      CC      SS
DD      DD      EE      AA      AA      CC      CC      CC      SS
DD      DD      EE      AA      AA      CC      CC      CC      SS
DD      DD      EE      AA      AA      CC      CC      CC      SS
DD      DD      EE      AA      AA      CC      CC      CC      SS
DD      DD      EE      AA      AA      CC      CC      CC      SS
DD      DD      EE      AA      AA      CC      CC      CC      SS
DDDDDDDD      EEEEEEEEEE      AAAAAA      CCCCCCCC      CCCCCCCC      SSSSSSSS
DDDDDDDD      EEEEEEEEEE      AAAAAA      CCCCCCCC      CCCCCCCC      SSSSSSSS
    
```

```

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
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
    
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

```

0001 0 MODULE DEACCS (
0002 0 LANGUAGE (BLISS32),
0003 0 IDENT = 'V04-000',
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1
0008 1 *****
0009 1 *
0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 * ALL RIGHTS RESERVED.
0013 1 *
0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 * TRANSFERRED.
0020 1 *
0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 * CORPORATION.
0024 1 *
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1 **
0032 1
0033 1 FACILITY: F11ACP Structure Level 1
0034 1
0035 1 ABSTRACT:
0036 1
0037 1 This routine implements the DEACCESS function.
0038 1
0039 1 ENVIRONMENT:
0040 1
0041 1 STARLET operating system, including privileged system services
0042 1 and internal exec routines.
0043 1
0044 1 --
0045 1
0046 1
0047 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 6-Jan-1977 23:29
0048 1
0049 1 MODIFIED BY:
0050 1
0051 1 V03-002 LMP0154 L. Mark Pilant, 14-Sep-1983 11:06
0052 1 Eliminate the write header protection check as it is
0053 1 either not done (for a create) or done elsewhere (for
0054 1 modify).
0055 1
0056 1 V03-001 ACG0343 Andrew C. Goldstein, 19-Jul-1983 16:53
0057 1 Inhibit revision count if NORECORD is specified

```

DEACCS  
V04-000

F 16  
16-Sep-1984 00:55:58  
14-Sep-1984 12:29:27

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[F11A.SRC]DEACCS.B32;1 Page (1) 2

```

: 58      0058 1 :
: 59      0059 1 :
: 60      0060 1 :
: 61      0061 1 :
: 62      0062 1 :
: 63      0063 1 :
: 64      0064 1 :
: 65      0065 1 :
: 66      0066 1 :
: 67      0067 1 :
: 68      0068 1 :
: 69      0069 1 :
: 70      0070 1 :
: 71      0071 1 :

V02-002 ACG0247      Andrew C. Goldstein,      23-Dec-1981  0:58
Update file revision count only if written

V02-001 ACG0245      Andrew C. Goldstein,      18-Dec-1981  18:27
Moving queueing of spool file to cleanup

V02-000 ACG0167      Andrew C. Goldstein,      7-May-1980   18:48
Previous revision history moved to F11A.REV

**

LIBRARY 'SYSS$LIBRARY:LIB.L32';
REQUIRE 'SRCS:FCPDEF.B32';
```

```

73 0386 1 GLOBAL ROUTINE DEACCESS =
74 0387 1
75 0388 1 |++
76 0389 1
77 0390 1 FUNCTIONAL DESCRIPTION:
78 0391 1
79 0392 1     This routine implements the DEACCESS function.
80 0393 1     If an attribute list is present, attributes are written.
81 0394 1
82 0395 1 CALLING SEQUENCE:
83 0396 1     DEACCESS ()
84 0397 1
85 0398 1 INPUT PARAMETERS:
86 0399 1     NONE
87 0400 1
88 0401 1 IMPLICIT INPUTS:
89 0402 1     IO_PACKET: I/O packet in process
90 0403 1     CURRENT_WINDOW: window of file
91 0404 1     PRIMARY_FCB: FCB of file
92 0405 1
93 0406 1 OUTPUT PARAMETERS:
94 0407 1     NONE
95 0408 1
96 0409 1 IMPLICIT OUTPUTS:
97 0410 1     NONE
98 0411 1
99 0412 1 ROUTINE VALUE:
100 0413 1     NONE
101 0414 1
102 0415 1 SIDE EFFECTS:
103 0416 1     file deaccessed
104 0417 1     FCB may be deleted
105 0418 1     header may be modified
106 0419 1
107 0420 1 |--
108 0421 1
109 0422 2 BEGIN
110 0423 2
111 0424 2 LOCAL
112 0425 2     K,                               ! local copy of truncate lock count
113 0426 2     ABD                          : REF BBLOCKVECTOR [ ,ABD$C_LENGTH],
114 0427 2                                     ! buffer descriptors
115 0428 2     FIB                             : REF BBLOCK,      ! FIB
116 0429 2     FCB                             : REF BBLOCK,      ! pointer to FCB
117 0430 2     HEADER                          : REF BBLOCK,      ! file header
118 0431 2     IDENT_AREA                       : REF BBLOCK;     ! header ident area
119 0432 2
120 0433 2 EXTERNAL
121 0434 2     CLEANUP_FLAGS                       : BITVECTOR,      ! cleanup action flags
122 0435 2     USER_STATUS                        : VECTOR,          ! I/O status going back to user
123 0436 2     IO_PACKET                          : REF BBLOCK,      ! I/O packet in process
124 0437 2     FILE_HEADER                        : REF BBLOCK,      ! global pointer to current file header
125 0438 2     CURRENT_WINDOW                     : REF BBLOCK,      ! window of file
126 0439 2     PRIMARY_FCB                        : REF BBLOCK,      ! FCB of file
127 0440 2     CURRENT_VCB                        : REF BBLOCK;     ! VCB of volume
128 0441 2
129 0442 2 EXTERNAL ROUTINE

```

```
130 0443 2 GET FIB, ; get FIB of request
131 0444 2 READ_HEADER, ; read file header
132 0445 2 MARK_DIRTY, ; mark buffer for write-back
133 0446 2 GET_TIME, ; get date and time string
134 0447 2 WRITE_ATTRIB, ; write attributes routine
135 0448 2 TRUNCATE, ; truncate file
136 0449 2 UPDATE_FCB, ; update contents of FCB
137 0450 2 CHECKSUM; ; compute file header checksum
138 0451 2
139 0452 2
140 0453 2 ! Set the cleanup flags to cause the deaccess to occur.
141 0454 2 ! Find the buffer descriptor and FIB.
142 0455 2 !
143 0456 2
144 0457 2 CLEANUP_FLAGS[CLF_ZCHANNEL] = 1;
145 0458 2 CLEANUP_FLAGS[CLF_DEACCESS] = 1;
146 0459 2 CLEANUP_FLAGS[CLF_DELWINDOW] = 1;
147 0460 2
148 0461 2 ! pointer to buffer descriptors
149 0462 2 ABD = .BBLOCK [.IO_PACKET[IRPSL_SVAPTE], AIB$L_DESCRIPTOR];
150 0463 2 FIB = GET_FIB (.ABD);
151 0464 2 FCB = .PRIMARY_FCB;
152 0465 2
153 0466 2 ! Make sure irrelevant parameters are not present.
154 0467 2 !
155 0468 2
156 0469 2 IF .FIB[FIB$V_EXTEND]
157 0470 2 THEN ERR_STATUS (SS$_BADPARAM);
158 0471 2
159 0472 2 ! If the file is accessed for write, or if the file is marked for delete
160 0473 2 ! and this is the last access, read the header.
161 0474 2 !
162 0475 2
163 0476 2 IF .CURRENT_WINDOW[WCB$V_WRITE]
164 0477 2 OR ((.FCB[FCB$V_MARKDEL] OR .CLEANUP_FLAGS[CLF_SPOOLFILE])
165 0478 2 AND .FCB[FCB$W_ACNT] EQL 1)
166 0479 2 THEN HEADER = READ_HEADER (0, .FCB);
167 0480 2
168 0481 2 ! If this the last deaccess from a file marked for delete, delete the file.
169 0482 2 ! If the file is a spool file, send it to the job controller.
170 0483 2 !
171 0484 2
172 0485 2 IF .FCB[FCB$W_ACNT] EQL 1
173 0486 2 THEN
174 0487 2 BEGIN
175 0488 2
176 0489 2 IF .FCB[FCB$V_MARKDEL]
177 0490 2 THEN CLEANUP_FLAGS[CLF_DELFILE] = 1;
178 0491 2
179 0492 2 IF .CLEANUP_FLAGS[CLF_SPOOLFILE]
180 0493 2 THEN CLEANUP_FLAGS[CLF_DOSPOOL] = 1;
181 0494 2 END;
182 0495 2
183 0496 2 ! Do deaccess processing for a write accessed file.
184 0497 2 !
185 0498 2
186 0499 2 IF .CURRENT_WINDOW[WCB$V_WRITE]
```

```
187 0500 2 THEN
188 0501 3 BEGIN
189 0502 3 MARK_DIRTY (.HEADER);
190 0503 3
191 0504 3
192 0505 3 ! Increment the revision count of the file. If a deaccess lock was
193 0506 3 requested on the file, set the lock bit. Then process the write
194 0507 3 attributes, if any. If attributes were written, then clear the
195 0508 3 lock bit.
196 0509 3
197 0510 3
198 0511 3 IF NOT .FIB[FIB$V NORECORD]
199 0512 4 AND (.CURRENT_WINDOW[WCBSL WRITES] NEQ 0
200 0513 4 OR .IO_PACKET[IRPSW BCNT] GTR ABD$C_ATTRIB
201 0514 4 OR .FIB[FIB$V_TRUNC])
202 0515 3 THEN
203 0516 4 BEGIN
204 0517 4 IDENT_AREA = .HEADER + .HEADER[FH1$B IDOFFSET]*2;
205 0518 4 IDENT_AREA[F11$W_REVISION] = .IDENT_AREA[F11$W_REVISION] + 1;
206 0519 4 GET_TIME (IDENT_AREA[F11$T_REVDATE]);
207 0520 3 END;
208 0521 3
209 0522 3 IF .CURRENT_WINDOW[WCBSV DLOCK]
210 0523 3 THEN HEADER[FH1$V_LOCKED] = 1;
211 0524 3
212 0525 3 IF .IO_PACKET[IRPSW_BCNT] GTR ABD$C_ATTRIB
213 0526 3 AND .USER_STATUS[0]
214 0527 3 THEN
215 0528 4 BEGIN
216 0529 4 WRITE ATTRIB (.HEADER, .ABD, 0);
217 0530 4 HEADER[FH1$V_LOCKED] = 0;
218 0531 3 END;
219 0532 3
220 0533 3 ! If a truncate is requested, do it (if the file was write accessed). Note
221 0534 3 that we must manually clear the truncate cleanup flag, since deaccess
222 0535 3 always exits through error exit.
223 0536 3
224 0537 3
225 0538 3 IF .FIB[FIB$V_TRUNC]
226 0539 3 THEN
227 0540 4 BEGIN
228 0541 4 IF .CURRENT_VCB[VCBSV NOALLOC]
229 0542 4 THEN ERR_EXIT (SS$WRITLCK);
230 0543 4
231 0544 4 K = .FCB[FCBSW TCNT];
232 0545 4 IF .CURRENT_WINDOW[WCBSV_NOTRUNC]
233 0546 4 THEN K = .K - 1;
234 0547 4 IF .K NEQ 0
235 0548 4 THEN ERR_EXIT (SS$ACCONFLICT);
236 0549 4
237 0550 4 CHECKSUM (.HEADER);
238 0551 4 TRUNCATE (.FIB, .HEADER, DEALLOC_BLOCKS);
239 0552 4 CLEANUP_FLAGS[CLF_CLEANTRUNC] = 0;
240 0553 4 CLEANUP_FLAGS[CLF_FIXFCB] = 0;
241 0554 4 KERNEL_CALL (UPDATE_FCB, .FILE_HEADER);
242 0555 3 END;
243 0556 3
```

```

: 244      0557 3   CHECKSUM (.HEADER);           ! checksum the file header
: 245      0558 2   END;
: 246      0559 2   !
: 247      0560 2   ! Return failure to let the error cleanup do the actual deaccessing.
: 248      0561 2   !
: 249      0562 2   !
: 250      0563 2   RETURN 0;
: 251      0564 2   !
: 252      0565 1   END;                       ! end of routine DEACCESS

```

```

.TITLE DEACCS
.IDENT \V04-000\

.EXTRN CLEANUP_FLAGS, USER_STATUS
.EXTRN IO_PACKET, FILE_HEADER
.EXTRN CURRENT_WINDOW, PRIMARY_FCB
.EXTRN CURRENT_VCB, GET_FIB
.EXTRN READ_HEADER, MARK_DIRTY
.EXTRN GET_TIME, WRITE_ATTRIB
.EXTRN TRUNCATE, UPDATE_FCB
.EXTRN CHECKSUM, SYSS$CMRNL

.PSECT $CODE$,NOWRT,2

.ENTRY DEACCESS, Save R2,R3,R4,R5,R6,R7,R8,R9 : 0386
MOVAB  USER_STATUS, R9
MOVAB  IO_PACKET, R8
MOVAB  CURRENT_WINDOW, R7
MOVAB  CLEANUP_FLAGS, R6
BISW2  #1027, CLEANUP_FLAGS+2 : 0459
MOVL   IO_PACKET, R0 : 0462
MOVL   @4(R0), ABD : 0463
PUSHL  ABD
CALLS  #1, GET_FIB
MOVL   R0, FIB
MOVL   PRIMARY_FCB, FCB : 0464
TSTB  22(FIB) : 0469
BGEQ  1$
BLBC  USER_STATUS, 1$ : 0470
MOVW  #20, USER_STATUS
MOVL  CURRENT_WINDOW, R0 : 0476
BBS  #1, 11(R0), 3$
BBS  #1, 34(FCB), 2$ : 0477
TSTB  CLEANUP_FLAGS
BGEQ  4$
CMPW  26(FCB), #1 : 0478
BNEQ  4$
PUSHL  FCB : 0479
CLRL  -(SP)
CALLS  #2, READ_HEADER
MOVL  R0, HEADER
CMPW  26(FCB), #1 : 0485
BNEQ  6$
BBC  #1, 34(FCB), 5$ : 0489
BISB2 #32, CLEANUP_FLAGS+2 : 0490
TSTB  CLEANUP_FLAGS : 0492

```

```

03FC 0000
59 0000G CF 9E 00002
58 0000G CF 9E 00007
57 0000G CF 9E 0000C
56 0000G CF 9E 00011
02 A6 0403 8F A8 00C16
50 68 D0 0001C
55 2C B0 D0 0001F
55 DD 00023
0000G CF 01 FB 00025
54 50 D0 0002A
52 0000G CF D0 0002D : 0464
16 A4 95 00032 : 0469
06 18 00035
03 69 E9 00037
69 14 B0 0003A
50 67 D0 0003D 1$:
OF 08 A0 01 E0 00040
04 22 A2 01 E0 00045
66 95 0004A
12 18 0004C
01 1A A2 B1 0004E 2$:
0C 12 00052
52 DD 00C54 3$:
7E D4 00C56
0000G CF 02 FB 00058
53 50 D0 0005D
01 1A A2 B1 00060 4$:
10 12 00064
04 22 A2 01 E1 00066
02 02 A6 20 88 0006B
66 95 0006F 5$:

```



			03	18	00071	BGEQ	6\$			
		66	04	88	00073	BISB2	#4, CLEANUP_FLAGS		0493	
		50	67	D0	00076	6\$:	MOVL	CURRENT_WINDOW, R0	0499	
03	OB	A0	01	E0	00079	BBS	#1, 11(R0), 7\$			
			00B6	31	0007E	BRW	16\$			
			53	DD	00081	7\$:	PUSHL	HEADER	0502	
27	0000G	CF	01	FB	00083	CALLS	#1, MARK_DIRTY			
		64	15	E0	00088	BBS	#21, (FIB), 9\$		0511	
		50	67	D0	0008C	MOVL	CURRENT_WINDOW, R0		0512	
			28	A0	D5	0008F	TSTL	40(R0)		
				0D	12	00092	BNEQ	8\$		
		50	68	D0	00094	MOVL	I0_PACKET, R0		0513	
		05	32	A0	B1	00097	CMPW	50(R0), #5		
				04	1A	0009B	BGTRU	8\$		
		12	17	A4	E9	0009D	BLBC	23(FIB), 9\$	0514	
		50	63	9A	000A1	8\$:	MOVZBL	(HEADER), R0	0517	
		50	6340	3E	000A4	MOVAV	(HEADER)[R0], IDENT_AREA			
			0A	A0	B6	000A8	INCW	10(IDENT_AREA)	0518	
			0C	A0	9F	000AB	PUSHAB	12(IDENT_AREA)	0519	
	0000G	CF	01	FB	000AE	CALLS	#1, GET_TIME			
		50	67	D0	000B3	9\$:	MOVL	CURRENT_WINDOW, R0	0522	
05	14	A0	01	E1	000B6	BBC	#1, 20(R0), 10\$			
	0C	A3	40	8F	88	000BB	BISB2	#64, 12(HEADER)	0523	
		50	68	D0	000C0	10\$:	MOVL	I0_PACKET, R0	0525	
		05	32	A0	B1	000C3	CMPW	50(R0), #5		
				11	1B	000C7	BLEQU	11\$		
		0E	69	E9	000C9	BLBC	USER_STATUS, 11\$		0526	
			7E	D4	000CC	CLRL	-(SPT)		0529	
			28	BB	000CE	PUSHR	#*M<R3, R5>			
	0000G	CF	03	FB	000D0	CALLS	#3, WRITE_ATTRIB			
		A3	40	8F	8A	000D5	BICB2	#64, 12(HEADER)	0530	
		52	17	A4	E9	000DA	11\$:	BLBC	23(FIB), 15\$	0538
		50	0000G	CF	D0	000DE	MOVL	CURRENT_VCB, R0	0541	
05	OB	A0	04	E1	000E3	BBC	#4, 11(R0), 12\$			
			025C	8F	BF	000E8	CHMU	#604	0542	
				04	000EC	RET				
		51	20	A2	3C	000ED	12\$:	MOVZWL	32(FCB), K	0544
		50	67	D0	000F1	MOVL	CURRENT_WINDOW, R0		0545	
02	15	A0	03	E1	000F4	BBC	#3, 21(R0), 13\$			
			51	D7	000F9	DECL	K		0546	
			51	D5	000FB	13\$:	TSTL	K	0547	
			05	13	000FD	BEQL	14\$			
			0800	8F	BF	000FF	CHMU	#2048	0548	
				04	00103	RET				
			53	DD	00104	14\$:	PUSHL	HEADER	0550	
	0000G	CF	01	FB	00106	CALLS	#1, CHECKSUM			
			01	DD	0010B	PUSHL	#1		0551	
			53	DD	0010D	PUSHL	HEADER			
			54	DD	0010F	PUSHL	FIB			
	0000G	CF	03	FB	00111	CALLS	#3, TRUNCATE			
		66	00080002	8F	CA	00116	BICL2	#524290, CLEANUP_FLAGS	0552	
			0000G	CF	DD	0011D	PUSHL	FILE_HEADER	0554	
				01	DD	00121	PUSHL	#1		
				5E	DD	00123	PUSHL	SP		
			0000G	CF	9F	00125	PUSHAB	UPDATE_FCB		
	00000000G	9F	04	FB	00129	CALLS	#4, @#SYSSCMKRNL			
			53	DD	00130	15\$:	PUSHL	HEADER	0557	

DEACCS  
V04-000

L 16  
16-Sep-1984 00:55:58  
14-Sep-1984 12:29:27

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[F11A.SRC]DEACCS.B32;1 Page 8  
(2)

0000G CF

01 FB 00132 CALLS #1, CHECKSUM  
50 D4 00137 16\$ CLRL RO  
04 00139 RET

: 0563  
: 0565

: Routine Size: 314 bytes, Routine Base: \$CODE\$ + 0000

: 253 0566 1  
: 254 0567 1 END  
: 255 0568 0 ELUDOM

PSECT SUMMARY

: Name Bytes Attributes  
: \$CODE\$ 314 NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

: File Total Symbols Loaded Percent Pages Mapped Processing Time  
: \_\$255\$DUA28:[SYSLIB]LIB.L32;1 18619 24 0 1000 00:01.8

COMMAND QUALIFIERS

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

: Size: 314 code + 0 data bytes  
: Run Time: 00:09.8  
: Elapsed Time: 00:27.3  
: Lines/CPU Min: 3495  
: Lexemes/CPU-Min: 14978  
: Memory Used: 144 pages  
: Compilation Complete

FCPOEF B32	ACPCNTR LIS	CHKSUM LIS	CHKPRO LIS	DEACCS LIS
BADSEN LIS	CLENUP LIS	CPYNAM LIS	CHKHDR LIS	COMMON LIS
CREHDR LIS	CREWIN LIS	ACCESS LIS	ALLOB LIS	CHKDMD LIS
CREATE LIS	CREFCB LIS	DELETE LIS		