

FFFFFFFFFF	111	111	111	XXX
FFFFFFFFFF	111	111	111	XXX
FFFFFFFFFF	111	111	111	XXX
FFF	111111	111111	111111	XXX
FFF	111111	111111	111111	XXX
FFF	111111	111111	111111	XXX
FFF	111	111	111	XXX
FFF	111	111	111	XXX
FFF	111	111	111	XXX
FFFFFFFFFF	111	111	111	XXX
FFFFFFFFFF	111	111	111	XXX
FFFFFFFFFF	111	111	111	XXX
FFF	111	111	111	XXX
FFF	111	111	111	XXX
FFF	111	111	111	XXX
FFF	111	111	111	XXX
FFF	111	111	111	XXX
FFF	111	111	111	XXX
FFF	111	111	111	XXX
FFF	111111111	111111111	111111111	XXX
FFF	111111111	111111111	111111111	XXX
FFF	111111111	111111111	111111111	XXX

MM	MM	AAAAAA	KK	KK	AAAAAA	CCCCCCCC	CCCCCCCC
MM	MM	AAAAAA	KK	KK	AA	CC	CC
MMMM	MMMM	AA	AA	KK	AA	AA	CC
MMMM	MMMM	AA	AA	KK	AA	AA	CC
MM MM	MM MM	AA	AA	KK	AA	AA	CC
MM MM	MM MM	AA	AA	KK	AA	AA	CC
MM MM	MM MM	AA	AA	KKKKKK	AA	AA	CC
MM MM	MM MM	AA	AA	KKKKKK	AA	AA	CC
MM MM	MM MM	AAAAAAA	KK	KK	AAAAAAA	CC	CC
MM MM	MM MM	AAAAAAA	KK	KK	AAAAAAA	CC	CC
MM MM	MM AA	AA	KK	KK	AA	AA	CC
MM MM	MM AA	AA	KK	KK	AA	AA	CC
MM MM	MM AA	AA	KK	KK	AA	AA	CC
MM MM	MM AA	AA	KK	KK	AA	AA	CC

....

LL		SSSSSSS
LL		SSSSSSS
LL		SS
LL		SS
LL		SS
LL		SSSSS
LL		SSSSS
LL		SS
LL		SS
LL		SS
LLLLLLLL		SSSSSSS
LLLLLLLL		SSSSSSS

```
1 0001 0 MODULE MAKACC (
2 0002 0   LANGUAGE (BLISS32),
3 0003 0   IDENT = 'V04-000'
4 0004 0   ) =
5 0005 1 BEGIN
6
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 *
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 ****
30 0030 1 *
31 0031 1 **+
32 0032 1 *
33 0033 1 FACILITY: F11ACP Structure Level 1
34 0034 1 *
35 0035 1 ABSTRACT:
36 0036 1 *
37 0037 1 This routine makes the necessary hookups in the I/O database to
38 0038 1 reflect a new file access.
39 0039 1 ENVIRONMENT:
40 0040 1 *
41 0041 1 STARLET operating system, including privileged system services
42 0042 1 and internal exec routines. This routine must be called
43 0043 1 in kernel mode.
44 0044 1 --
45 0045 1
46 0046 1
47 0047 1
48 0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 20-Dec-1976 17:28
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V03-004 CDS0003 Christian D. Saether 19-Apr-1984
53 0053 1 Bump REFCNT in Fcb. Do not bump other counts if
54 0054 1 this is NOACLOCK. Remove reference to old dirfcb index.
55 0055 1
56 0056 1 V03-003 CDS0002 Christian D. Saether 2-Mar-1984
57 0057 1 Set WRITE_TURN flag in WCB if index file, storage bitmap.
```

58 0058 1 or a directory is being write accessed.
59 0059 1
60 0060 1 V03-002 CDS0001 Christian D. Saether 30-Dec-1983
61 0061 1 Use L_NORM linkage and BIND_COMMON macro.
62 0062 1
63 0063 1 V03-001 LMP0059 L. Mark Pilant, 4-Jan-1983 12:28
64 0064 1 Don't insert the FCB into the queue as it is done when the
65 0065 1 FCB is created.
66 0066 1
67 0067 1 V02-002 LMP0003 L. Mark Pilant, 20-Nov-1981 9:30
68 0068 1 Modify so that all the segments to a window get inserted into
69 0069 1 the window queue.
70 0070 1
71 0071 1 V02-001 ACG0167 Andrew C. Goldstein, 16-Apr-1980 19:26
72 0072 1 Previous revision history moved to F11B.REV
73 0073 1 **
74 0074 1
75 0075 1
76 0076 1 LIBRARY 'SYSSLIBRARY:LIB.L32';
77 0077 1 REQUIRE 'SRC\$:FCPDEF.B32';

```
: 79      1068 1 GLOBAL ROUTINE MAKE_ACCESS (FCB, WINDOW, ABD) : L_NORM NOVALUE =
: 80
: 81      1069 1
: 82      1070 1 !++
: 83
: 84      1071 1 FUNCTIONAL DESCRIPTION:
: 85      1072 1
: 86      1073 1 This routine makes the necessary hookups in the I/O database to
: 87      1074 1 reflect a new file access.
: 88      1075 1
: 89      1076 1 CALLING SEQUENCE:
: 90      1077 1 MAKE_ACCESS (ARG1, ARG2, ARG3)
: 91
: 92      1078 1 INPUT PARAMETERS:
: 93      1079 1     ARG1: address of FCB to access
: 94      1080 1     ARG2: address of window to link up
: 95      1081 1     ARG3: address of buffer descriptors
: 96
: 97      1082 1 IMPLICIT INPUTS:
: 98      1083 1     CURRENT_VCB: VCB of volume in process
: 99
: 100     1084 1 OUTPUT PARAMETERS:
: 101     1085 1     NONE
: 102
: 103     1086 1 IMPLICIT OUTPUTS:
: 104     1087 1     NONE
: 105
: 106     1088 1 ROUTINE VALUE:
: 107     1089 1     NONE
: 108
: 109     1090 1 SIDE EFFECTS:
: 110     1091 1     VCB transaction count bumped, access counts in FCB adjusted,
: 111     1092 1     FCB and window hooked up.
: 112
: 113     1093 1
: 114     1094 1 BEGIN
: 115
: 116     1095 1 MAP
: 117     1096 2     FCB          : REF BBLOCK,      ! FCB arg
: 118     1097 2     WINDOW       : REF BBLOCK,      ! window arg
: 119     1098 2     ABD          : REF BBLOCKVECTOR [,ABD$C_LENGTH];
: 120     1099 2                           ! buffer descriptor arg
: 121
: 122     1100 1
: 123     1101 1 --+
: 124
: 125     1102 1
: 126     1103 2
: 127     1104 2
: 128     1105 2
: 129
: 130     1106 2 LOCAL    WINDOW_SEGMENT : REF BBLOCK;      ! address of the current window segment
: 131
: 132     1107 2
: 133     1108 2
: 134     1109 2
: 135     1110 2
: 136
: 137     1111 2
: 138     1112 2
: 139     1113 2
: 140     1114 2
: 141     1115 2
: 142     1116 2
: 143     1117 2
: 144     1118 2
: 145     1119 2
: 146     1120 2
: 147     1121 2
: 148     1122 2
: 149     1123 2
: 150     1124 2
```

```
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
1125 2 WINDOW_SEGMENT = .WINDOW;  
1126 2 DO  
1127 3 BEGIN  
1128 3 INSQUE (.WINDOW_SEGMENT, .FCB[FCBSL_WLBL]);  
1129 3 WINDOW_SEGMENT = .WINDOW_SEGMENT[WCBSL_LINK];  
1130 3 END  
1131 2 UNTIL .WINDOW_SEGMENT EQL 0;  
1132 2 FCB [FCBSW_REFCNT] = .FCB [FCBSW_REFCNT] + 1; ! bump reference count  
1133 2 IF NOT .WINDOW [WCBSV_NOACCCLOCK]  
1134 2 THEN  
1135 3 BEGIN  
1136 3 FCBSW_ACNT = .FCB[FCBSW_ACNT] + 1; ! bump access count  
1137 3 IF .WINDOW[WCBSV_NOREAD]  
1138 3 THEN FCBSV_EXCL = 1; ! set exclusive access  
1139 3 IF .WINDOW[WCBSV_NOWRITE]  
1140 3 THEN FCBSW_LCNT = .FCB[FCBSW_LCNT] + 1; ! no writers  
1141 3 IF .WINDOW[WCBSV_NOTRUNC]  
1142 3 THEN FCBSW_TCNT = .FCB[FCBSW_TCNT] + 1; ! no truncates  
1143 3 END;  
1144 2 : For a write access, bump the writer count. If this is the  
1145 2 : first write, and the file is the index file or the storage map, set  
1146 2 : the appropriate flag in the VCB.  
1147 2 :  
1148 2 IF .WINDOW[WCBSV_WRITE]  
1149 2 THEN  
1150 3 BEGIN  
1151 3 IF .FCB [FCBSB_FID_NMX] EQL 0  
1152 3 THEN  
1153 4 BEGIN  
1154 4 IF .FCB[FCBSW_FID_NUM] EQL 1  
1155 4 THEN  
1156 5 BEGIN  
1157 5 CURRENT_VCB[VCBSV_WRITE_IF] = 1;  
1158 5 WINDOW [WCBSV_WRITE_TURN] = 1;  
1159 4 END;  
1160 4  
1161 4 IF .FCB[FCBSW_FID_NUM] EQL 2  
1162 4 THEN  
1163 5 BEGIN  
1164 5 CURRENT_VCB[VCBSV_WRITE_SM] = 1;  
1165 5 WINDOW [WCBSV_WRITE_TURN] = 1;  
1166 4 END;  
1167 3 END;  
1168 3  
1169 3 IF .FCB[FCBSV_DIR]  
1170 3 THEN  
1171 4 BEGIN
```


07	15	50	08	AC	D0	0004B	3\$:	MOVL	WINDOW, R0	: 1148
		A0	03	E1	0004F		BBC	#3, 21(R0), 4\$		
		50	04	AC	D0	00054	MOVL	FCB, R0	1149	
			20	A0	B6	00058	INCW	32(R0)		
5C	08	50	08	AC	D0	0005B	4\$:	MOVL	WINDOW, R0	1158
		A0	01	E1	0005F		BBC	#1, 11(R0), 8\$		
		50	04	AC	D0	00064	MOVL	FCB, R0	1161	
			29	A0	95	00068	TSTB	41(R0)		
				30	12	0006B	BNEQ	6\$		
			01	24	A0	B1	CMPW	36(R0), #1	1164	
				10	12	00071	BNEQ	5\$		
	08	50	98	AA	D0	00073	MOVL	-104(BASE), R0	1167	
		A0	01	88	00077		BISB2	#1, 11(R0)		
		50	08	AC	D0	0007B	MOVL	WINDOW, R0	1168	
	15	A0	08	10	88	0007F	BISB2	#16, 21(R0)		
		50	04	AC	D0	00083	5\$::	MOVL	FCB, R0	1171
		02	24	A0	B1	00087	CMPW	36(R0), #2		
				10	12	0008B	BNEQ	6\$		
	08	50	98	AA	D0	0008D	MOVL	-104(BASE), R0	1174	
		A0	02	88	00091		BISB2	#2, 11(R0)		
		50	08	AC	D0	00095	MOVL	WINDOW, R0	1175	
	15	A0	08	10	88	00099	BISB2	#16, 21(R0)		
		50	04	AC	D0	0009D	6\$::	MOVL	FCB, R0	1179
		0B	22	A0	E9	000A1	BLBC	34(R0), 7\$		
			42	A0	B6	000A5	INCW	66(R0)	1182	
	15	50	08	AC	D0	000A8	MOVL	WINDOW, R0	1183	
		A0	10	88	000AC		BISB2	#16, 21(R0)		
		50	08	AC	D0	000B0	7\$::	MOVL	WINDOW, R0	1186
07	14	A0	02	E0	000B4		BBS	#2, 20(R0), 8\$		
		50	04	AC	D0	000B9	MOVL	FCB, R0	1188	
			1C	A0	B6	000BD	INCW	28(R0)		
			00000000G	9F	D6	000C0	8\$::	INCL	@#PMSSGL_OPEN	1196
			00000000G	9F	D6	000C6		INCL	@#PMSSGL_OPENS	1197
			50	98	AA	D0	000CC	MOVL	-104(BASE), R0	1198
				OC	A0	B6	000D0	INCW	12(R0)	
	02	50	0C	AC	D0	000D3	MOVL	ABD, R0	1200	
		A0	04	B0	000D7		MOVW	#4, 2(R0)		
		50	0C	BC	3C	000DB	MOVZWL	2ABD, R0	1201	
		50	0C	AC	C0	000DF	ADDL2	ABD, R0		
	01	A0	08	AC	D0	000E3	MOVL	WINDOW, 1(R0)		
	02	AA	01	88	000E8		BISB2	#1, 2(BASE)	1207	
				04	000EC		RET		1209	

: Routine Size: 237 bytes. Routine Base: \$CODE\$ + 0000

: 221 1210 1
: 222 1211 1 END
: 223 1212 0 ELUDOM

PSECT SUMMARY

MAKACC
V04-000

M 7
16-Sep-1984 00:42:55 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:30:34 DISK\$VMSMASTER:[F1IX.SRC]MAKACC.B32;1 Page 7
(2)

Name	Bytes	Attributes
\$CODE\$	237	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Symbols			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$_\$255\$DUA2B:[SYSLIB]LIB.L32;1	18619	42	0	1000	00:01.9

COMMAND QUALIFIERS

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

Size: 237 code + 0 data bytes
Run Time: 00:18.3
Elapsed Time: 00:38.3
Lines/CPU Min: 3984
Lexemes/CPU-Min: 49384
Memory Used: 239 pages
Compilation Complete

MAK
V04

0171 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LOCKDB
LIS

LOCKERS
LIS

MAKACC
LIS

MAKPTR
LIS

MATCHNAME
LIS

MPWIND
LIS

PARSNM
LIS

QUOTAUTIL
LIS

100ONE
LIS

LOCKON
LIS

MAPUBN
LIS

MODIFY
LIS

MOUNT
LIS

NXTHDR
LIS

MAKNMB
LIS

MAKSTR
LIS