


```

: 1 0001 0 MODULE MAKACC (
: 2 0002 0
:001 :CDS0004 0003 0 LANGUAGE (BLISS32),
: 3 0004 0 IDENT = 'V04-001'
: 4 -1 0005 1 BEGIN
: 5 0006 1
: 6 0007 1
: 7 0008 1
: 8 0009 1
: 9 0010 1
:10 0011 1 *
:11 0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
:12 0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
:13 0014 1 * ALL RIGHTS RESERVED.
:14 0015 1 *
:15 0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
:16 0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
:17 0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
:18 0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
:19 0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
:20 0021 1 * TRANSFERRED.
:21 0022 1 *
:22 0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
:23 0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
:24 0025 1 * CORPORATION.
:25 0026 1 *
:26 0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
:27 0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
:28 0029 1 *
:29 0030 1 *****
:30 0031 1 ++
:31 0032 1
:32 0033 1 FACILITY: F11ACP Structure Level 1
:33 0034 1
:34 0035 1 ABSTRACT:
:35 0036 1
:36 0037 1 This routine makes the necessary hookups in the I/O database to
:37 0038 1 reflect a new file access.
:38 0039 1 ENVIRONMENT:
:39 0040 1
:40 0041 1 STARLET operating system, including privileged system services
:41 0042 1 and internal exec routines. This routine must be called
:42 0043 1 in kernel mode.
:43 0044 1
:44 0045 1 --
:45 0046 1
:46 0047 1
:47 0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 20-Dec-1976 17:28
:48 0049 1
:49 0050 1 MODIFIED BY:
:50 0051 1
:51 0052 1 V04-001 CDS0004 Christian D. Saether 14-Nov-1984
:001 :CDS0004 0053 1 Remove test for directory here. It must check the
:002 :CDS0004 0054 1 file header instead in ACCESS.
:003 :CDS0004 0055 1
:004 :CDS0004 0056 1 V03-004 CDS0003 Christian D. Saether 19-Apr-1984
: 52 0057 1 Bump REFCNT in Fcb. Do not bump other counts if
: 53

```

```
.. 54 0058 1 | this is NOACCLOCK. Remove reference to old dirfcb index.  
.. 55 0059 1 |  
.. 56 0060 1 | V03-003 CDS0002 Christian D. Saether 2-Mar-1984  
.. 57 0061 1 | Set WRITE_TURN flag in WCB if index file, storage bitmap,  
.. 58 0062 1 | or a directory is being write accessed.  
.. 59 0063 1 |  
.. 60 0064 1 | V03-002 CDS0001 Christian D. Saether 30-Dec-1983  
.. 61 0065 1 | Use L_NORM linkage and BIND_COMMON macro.  
.. 62 0066 1 |  
.. 63 0067 1 | V03-001 LMP0059 L. Mark Pilant, 4-Jan-1983 12:28  
.. 64 0068 1 | Don't insert the FCB into the queue as it is done when the  
.. 65 0069 1 | FCB is created.  
.. 66 0070 1 |  
.. 67 0071 1 | V02-002 LMP0003 L. Mark Pilant, 20-Nov-1981 9:30  
.. 68 0072 1 | Modify so that all the segments to a window get inserted into  
.. 69 0073 1 | the window queue.  
.. 70 0074 1 |  
.. 71 0075 1 | V02-001 ACG0167 Andrew C. Goldstein, 16-Apr-1980 19:26  
.. 72 0076 1 | Previous revision history moved to F11B.REV  
.. 73 0077 1 | **  
.. 74 0078 1 |  
.. 75 0079 1 |  
.. 76 0080 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';  
.. 77 0081 1 | REQUIRE 'SRCS:FCPDEF.B32';
```

```

79 1072 1 GLOBAL ROUTINE MAKE_ACCESS (FCB, WINDOW, ABD) : L_NORM NOVALUE =
80 1073 1
81 1074 1 ++
82 1075 1
83 1076 1 FUNCTIONAL DESCRIPTION:
84 1077 1
85 1078 1 This routine makes the necessary hookups in the I/O database to
86 1079 1 reflect a new file access.
87 1080 1
88 1081 1 CALLING SEQUENCE:
89 1082 1 MAKE_ACCESS (ARG1, ARG2, ARG3)
90 1083 1
91 1084 1 INPUT PARAMETERS:
92 1085 1 ARG1: address of FCB to access
93 1086 1 ARG2: address of window to link up
94 1087 1 ARG3: address of buffer descriptors
95 1088 1
96 1089 1 IMPLICIT INPUTS:
97 1090 1 CURRENT_VCB: VCB of volume in process
98 1091 1
99 1092 1 OUTPUT PARAMETERS:
100 1093 1 NONE
101 1094 1
102 1095 1 IMPLICIT OUTPUTS:
103 1096 1 NONE
104 1097 1
105 1098 1 ROUTINE VALUE:
106 1099 1 NONE
107 1100 1
108 1101 1 SIDE EFFECTS:
109 1102 1 VCB transaction count bumped, access counts in FCB adjusted,
110 1103 1 FCB and window hooked up.
111 1104 1
112 1105 1 --
113 1106 1
114 1107 2 BEGIN
115 1108 2
116 1109 2 MAP
117 1110 2 FCB : REF BBLOCK, ! FCB arg
118 1111 2 WINDOW : REF BBLOCK, ! window arg
119 1112 2 ABD : REF BBLOCKVECTOR [ ,ABD$C_LENGTH];
120 1113 2 ! buffer descriptor arg
121 1114 2
122 1115 2 LOCAL
123 1116 2 WINDOW_SEGMENT : REF BBLOCK; ! address of the current window segment
124 1117 2
125 1118 2 BIND_COMMON;
126 1119 2
127 1120 2 EXTERNAL
128 1121 2 PM$SGL_OPEN : ADDRESSING_MODE (ABSOLUTE),
129 1122 2 ! system count of currently open files
130 1123 2 PM$SGL_OPENS : ADDRESSING_MODE (ABSOLUTE);
131 1124 2 ! system count of files opened
132 1125 2
133 1126 2 ! Now hook the window onto the FCB and adjust the access counts
134 1127 2 ! according to the access control bits in the window.
135 1128 2

```

```

136 1129 WINDOW_SEGMENT = .WINDOW;
137 1130 DO
138 1131     BEGIN
139 1132     INSQUE (.WINDOW_SEGMENT, .FCB[FCBSL_WLBL]);
140 1133     WINDOW_SEGMENT = .WINDOW_SEGMENT[WCBSL_LINK];
141 1134     END
142 1135 UNTIL .WINDOW_SEGMENT EQL 0;
143 1136
144 1137 FCB [FCBSW_REFCNT] = .FCB [FCBSW_REFCNT] + 1;    ! bump reference count
145 1138
146 1139 IF NOT .WINDOW [WCBSV_NOACCLOCK]
147 1140 THEN
148 1141     BEGIN
149 1142     FCB[FCBSW_ACNT] = .FCB[FCBSW_ACNT] + 1;    ! bump access count
150 1143
151 1144     IF .WINDOW[WCBSV_NOREAD]
152 1145     THEN FCB[FCBSV_EXCL] = 1;    ! set exclusive access
153 1146
154 1147     IF .WINDOW[WCBSV_NOWRITE]
155 1148     THEN FCB[FCBSW_LCNT] = .FCB[FCBSW_LCNT] + 1; ! no writers
156 1149
157 1150     IF .WINDOW[WCBSV_NOTRUNC]
158 1151     THEN FCB[FCBSW_TCNT] = .FCB[FCBSW_TCNT] + 1; ! no truncates
159 1152
160 1153     END;
161 1154
162 1155 ! For a write access, bump the writer count. If this is the
163 1156 ! first write, and the file is the index file or the storage map, set
164 1157 ! the appropriate flag in the VCB.
165 1158
166 1159 IF .WINDOW[WCBSV_WRITE]
167 1160 THEN
168 1161     BEGIN
169 1162     IF .FCB [FCBSB_FID_NMX] EQL 0
170 1163     THEN
171 1164         BEGIN
172 1165         IF .FCB[FCBSW_FID_NUM] EQL 1
173 1166         THEN
174 1167             BEGIN
175 1168             CURRENT_VCB[VCBSV_WRITE_IF] = 1;
176 1169             WINDOW [WCBSV_WRITE_TURN] = 1;
177 1170             END;
178 1171
179 1172         IF .FCB[FCBSW_FID_NUM] EQL 2
180 1173         THEN
181 1174             BEGIN
182 1175             CURRENT_VCB[VCBSV_WRITE_SM] = 1;
183 1176             WINDOW [WCBSV_WRITE_TURN] = 1;
184 1177             END;
185 1178
186 1179         END;
187 1180     END;
188 1181
189 1182 IF NOT .WINDOW [WCBSV_NOACCLOCK]
190 1183 THEN
191 1184     FCB[FCBSW_WCNT] = .FCB[FCBSW_WCNT] + 1;
192 1185
193 1186
194 1187
195 1188
196 1189
197-7 1190
198 1191
199 1192

```

```

: 200      1186      3
: 201      1187      3      END;
: 202      1188      3
: 203      1189      3      ! Count the access in the volume transaction count and return
: 204      1190      3      ! the window address for the user's CCB.
: 205      1191      3
: 206      1192      3
: 207      1193      3      PMSSGL_OPEN = .PMSSGL_OPEN + 1;      ! bump open file count
: 208      1194      3      PMSSGL_OPENS = .PMSSGL_OPENS + 1;      ! bump count of opens
: 209      1195      3      CURRENT_VCB[VCBSW_TRANS] = .CURRENT_VCB[VCBSW_TRANS] + 1;
: 210      1196      3
: 211      1197      3      ABD[ABDSC_WINDOW, ABDSW_COUNT] = 4;      ! enable write-back
: 212      1198      3      .ABD[ABDSC_WINDOW, ABDSW_TEXT] + ABD[ABDSC_WINDOW, ABDSW_TEXT] + 1 = .WINDOW;
: 213      1199      3      ! put window address in buffer text
: 214      1200      3
: 215      1201      3      ! Mark the access complete in the cleanup action flags.
: 216      1202      3
: 217      1203      3
: 218      1204      3      CLEANUP_FLAGS[CLF_DEACCESS] = 1;
: 219      1205      3
: 220      1206      3      1 END;

```

! end of routine MAKE_ACCESS

```

                                .TITLE  MAKACC
                                .IDENT  \V04-001\
                                .EXTRN  PMSSGL_OPEN, PMSSGL_OPENS
                                .PSECT  $CODE$,NOWRT,2

                                .ENTRY  MAKE_ACCESS, Save nothing
                                MOVL    WINDOW, WINDOW_SEGMENT      : 1072
                                MOVL    FCB, R0                      : 1130
                                INSQUE  (WINDOW_SEGMENT), @20(R0)   : 1133
                                MOVL    32(WINDOW_SEGMENT), WINDOW_SEGMENT
                                BNEQ    1$                          : 1134
                                MOVL    FCB, R0                      : 1136
                                INCW    24(R0)                      : 1138
                                MOVL    WINDOW, R0                  : 1140
                                BBS     #2, 20(R0), 4$              :
                                MOVL    FCB, R0                      : 1144
                                INCW    26(R0)                      :
                                MOVL    WINDOW, R0                  : 1146
                                BBC     #2, 21(R0), 2$              :
                                MOVL    FCB, R0                      : 1147
                                BISB2  #8, 34(R0)                   :
                                MOVL    WINDOW, R0                  : 1149
                                BLBC   20(R0), 3$                   :
                                MOVL    FCB, R0                      : 1150
                                INCW    30(R0)                      :
                                MOVL    WINDOW, R0                  : 1152
                                BBC     #3, 21(R0), 4$              :
                                MOVL    FCB, R0                      : 1153
                                INCW    32(R0)                      :
                                MOVL    WINDOW, R0                  : 1162
                                BBC     #1, 11(R0), 7$              :
                                MOVL    FCB, R0                      : 1165

```

		29	A0	95	00068	TSTB	41(R0)	
		30	12	0006B	BNEQ	6\$		
	01	24	A0	B1	0006D	CMPW	36(R0), #1	1168
		10	12	00071	BNEQ	5\$		
	50	98	AA	D0	00073	MOVL	-104(BASE), R0	1171
0B	A0		01	88	00077	BISB2	#1, 11(R0)	
	50	08	AC	D0	0007B	MOVL	WINDOW, R0	1172
15	A0		10	88	0007F	BISB2	#16, 21(R0)	
	50	04	AC	D0	00083	MOVL	FCB, R0	1175
	02	24	A0	B1	00087	CMPW	36(R0), #2	
		10	12	0008B	BNEQ	6\$		
	50	98	AA	D0	0008D	MOVL	-104(BASE), R0	1178
0B	A0		02	88	00091	BISB2	#2, 11(R0)	
	50	08	AC	D0	00095	MOVL	WINDOW, R0	1179
15	A0		10	88	00099	BISB2	#16, 21(R0)	
	50	08	AC	D0	0009D	MOVL	WINDOW, R0	1183
07	14		02	E0	000A1	BBS	#2, 20(R0), 7\$	
	50	04	AC	D0	000A6	MOVL	FCB, R0	1185
		1C	A0	B6	000AA	INCW	28(R0)	
		00000000G	9F	D6	000AD	INCL	@#PMSS\$GL_OPEN	1193
		00000000G	9F	D6	000B3	INCL	@#PMSS\$GL_OPENS	1194
	50	98	AA	D0	000B9	MOVL	-104(BASE), R0	1195
		0C	A0	B6	000BD	INCW	12(R0)	
	50	0C	AC	D0	000C0	MOVL	ABD, R0	1197
02	A0		04	B0	000C4	MOVW	#4, 2(R0)	
	50	0C	BC	3C	000C8	MOVZWL	@ABD, R0	1198
	50	0C	AC	C0	000CC	ADDL2	ABD, R0	
01	A0		08	AC	D0	000D0	MOVL	WINDOW, 1(R0)
	02	AA	01	88	000D5	BISB2	#1, 2(BASE)	1204
			04	000D9	RET			1206

: Routine Size: 218 bytes, Routine Base: \$CODE\$ + 0000

```
: 221      1207  1
: 222      1208  1 END
: 223      1209  0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	218	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

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

MAKACC
V04-001

D 12
8-Jan-1985 18:12:07
2-Oct-1984 12:43:34

VAX-11 Bliss-32 V4.0-742
[F11X.BUGSRC]MAKACC.B32;1

Page 7
(2)

: _\$255\$DUA18:[SYSLIB]LIB.L32;1 18619 40 0 1000 00:02.0

COMMAND QUALIFIERS

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

: Size: 218 code + 0 data bytes
: Run Time: 00:19.0
: Elapsed Time: 00:35.0
: Lines/CPU Min: 3809
: Lexemes/CPU-Min: 46856
: Memory Used: 232 pages
: Compilation Complete

