



```

DDDDDDDD      IIIIII  RRRRRRRR      AAAAAA      CCCCCCCC      CCCCCCCC
DDDDDDDD      IIIIII  RRRRRRRR      AAAAAA      CCCCCCCC      CCCCCCCC
DD      DD      II      RR      RR      AA      AA      CC      CC      CC      CC
DD      DD      II      RR      RR      AA      AA      CC      CC      CC      CC
DD      DD      II      RR      RR      AA      AA      CC      CC      CC      CC
DD      DD      II      RR      RR      AA      AA      CC      CC      CC      CC
DD      DD      II      RRRRRRRR      AA      AA      CC      CC      CC      CC
DD      DD      II      RRRRRRRR      AA      AA      CC      CC      CC      CC
DD      DD      II      RR      RR      AAAAAAAAAA      CC      CC      CC      CC
DD      DD      II      RR      RR      AAAAAAAAAA      CC      CC      CC      CC
DD      DD      II      RR      RR      AA      AA      CC      CC      CC      CC
DD      DD      II      RR      RR      AA      AA      CC      CC      CC      CC
DDDDDDDD      IIIIII  RR      RR      AA      AA      CCCCCCCC      CCCCCCCC
DDDDDDDD      IIIIII  RR      RR      AA      AA      CCCCCCCC      CCCCCCCC

```

```

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 DIRACC (
0002 0
0003 0     LANGUAGE (BLISS32),
0004 0     IDENT = 'V04-000'
0005 1 BEGIN
0006 1
0007 1
0008 1
0009 1
0010 1 *
0011 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0012 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0013 1 *   ALL RIGHTS RESERVED.
0014 1 *
0015 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0016 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0017 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0018 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0019 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0020 1 *   TRANSFERRED.
0021 1 *
0022 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0023 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0024 1 *   CORPORATION.
0025 1 *
0026 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0027 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0028 1 *
0029 1 *
0030 1
0031 1 **
0032 1
0033 1 FACILITY: F11ACP Structure Level 2
0034 1
0035 1 ABSTRACT:
0036 1
0037 1     This routine "accesses" a directory and readies it for processing.
0038 1 ENVIRONMENT:
0039 1
0040 1     STARLET operating system, including privileged system services
0041 1     and internal exec routines.
0042 1
0043 1 --
0044 1
0045 1
0046 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 2-Jan-1977 22:53
0047 1
0048 1 MODIFIED BY:
0049 1
0050 1     V03-015 CDS0009      Christian D. Saether    29-Aug-1984
0051 1     Deal with potential multiple headers caused by
0052 1     ACL's.
0053 1
0054 1     V03-014 CDS0008      Christian D. Saether    31-July-1984
0055 1     Changes to accomodate block-cache based directory
0056 1     index caching.
0057 1

```

58	0058	1	V03-013	LMP0275	L. Mark Pilant,	12-Jul-1984	14:56
59	0059	1			Initialize the ACL info in the ORB to be a null descriptor		
60	0060	1			list rather than an empty queue. This avoids the overhead		
61	0061	1			of locking and unlocking the ACL mutex, only to find out		
62	0062	1			that the ACL was empty.		
63	0063	1					
64	0064	1	V03-012	CDS0007	Christian D. Saether	22-Apr-1984	
65	0065	1			Modify access arbitration.		
66	0066	1					
67	0067	1	V03-011	ACG0412	Andrew C. Goldstein,	22-Mar-1984	18:23
68	0068	1			Implement agent access mode support; add access mode		
69	0069	1			check protection call		
70	0070	1					
71	0071	1	V03-010	CDS0006	Christian D. Saether	29-Dec-1983	
72	0072	1			Use L_NORM linkage and BIND_COMMON macro.		
73	0073	1					
74	0074	1	V03-009	CDS0005	Christian D. Saether	3-Oct-1983	
75	0075	1			Save/restore CURR_LCKINDX instead of PRIM_LCKINDX.		
76	0076	1					
77	0077	1	V03-008	CDS0004	Christian D. Saether	14-Sep-1983	
78	0078	1			Modify interface to SERIAL_FILE.		
79	0079	1					
80	0080	1	V03-007	ACG0354	Andrew C. Goldstein,	13-Sep-1983	16:16
81	0081	1			Add alternate access validation mask		
82	0082	1					
83	0083	1	V03-006	CDS0003	Christian D. Saether	4-May-1983	
84	0084	1			Add directory file processing synchronization		
85	0085	1			using SERIAL_FILE.		
86	0086	1			Move directory protection check and access arbitration		
87	0087	1			checks to path that sets up DIR_FCB.		
88	0088	1					
89	0089	1	V03-005	CDS0002	Christian D. Saether	9-Apr-1983	
90	0090	1			Reflect changes to ACCESS_LOCK interface.		
91	0091	1					
92	0092	1	V03-004	CDS0001	Christian D. Saether	20-Jan-1982	
93	0093	1			Use ACCESS_LOCK routine for access arbitration.		
94	0094	1					
95	0095	1	V03-003	ACG0308	Andrew C. Goldstein,	14-Jan-1983	15:07
96	0096	1			Fix consistency problems with FCB linkages		
97	0097	1					
98	0098	1	V03-002	LMP0059	L. Mark Pilant,	22-Dec-1982	8:22
99	0099	1			Eliminate the call to ACL_BUILDACL as the ACL is now built by		
100	0100	1			the FCB creation/entering routines.		
101	0101	1					
102	0102	1	V03-001	LMP0036	L. Mark Pilant,	6-Aug-1982	10:30
103	0103	1			Add support for ACL's.		
104	0104	1					
105	0105	1	V02-005	ACG0229	Andrew C. Goldstein,	23-Dec-1981	20:55
106	0106	1			Count directory LRU hits and misses		
107	0107	1					
108	0108	1	V02-004	ACG33975	Andrew C. Goldstein,	13-Mar-1981	15:46
109	0109	1			Fix losing directory FCB's when LRU is disabled		
110	0110	1					
111	0111	1	V02-003	ACG0167	Andrew C. Goldstein,	16-Apr-1980	19:25
112	0112	1			Previous revision history moved to f11B.REV		
113	0113	1					
114	0114	1					

DIRACC  
V04-000

I 11  
16-Sep-1984 00:19:02  
5-Sep-1984 22:06:33

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[F11X.SRC]DIRACC.B32;1 Page 3  
(1)

```

: 115      0115 1
: 116      0116 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 117      0117 1 REQUIRE 'SRC$:FCPDEF.B32';
: 118      1108 1

```

DI  
VC

.....

```

120 1109 1 GLOBAL ROUTINE DIR_ACCESS (FIB, WRITE) : L_NORM NOVALUE =
121 1110 1
122 1111 1 !**
123 1112 1
124 1113 1 FUNCTIONAL DESCRIPTION:
125 1114 1
126 1115 1 This routine "accesses" a directory and readies it for processing.
127 1116 1
128 1117 1 CALLING SEQUENCE:
129 1118 1 DIR_ACCESS (ARG1, ARG2)
130 1119 1
131 1120 1 INPUT PARAMETERS:
132 1121 1 ARG1: address of FIB
133 1122 1 ARG2: 0 if read only
134 1123 1 1 if read/write
135 1124 1
136 1125 1 IMPLICIT INPUTS:
137 1126 1 NONE
138 1127 1
139 1128 1 OUTPUT PARAMETERS:
140 1129 1 NONE
141 1130 1
142 1131 1 IMPLICIT OUTPUTS:
143 1132 1 DIR_FCB: address of FCB for directory
144 1133 1
145 1134 1 ROUTINE VALUE:
146 1135 1 NONE
147 1136 1
148 1137 1 SIDE EFFECTS:
149 1138 1 FCB may be created
150 1139 1 directory LRU may be altered
151 1140 1 directory file header may be read
152 1141 1
153 1142 1 --
154 1143 1
155 1144 2 BEGIN
156 1145 2
157 1146 2 MAP
158 1147 2 FIB : REF BBLOCK; ! FIB argument
159 1148 2
160 1149 2 LOCAL
161 1150 2 CURR_LKMODE, ! current access lock mode
162 1151 2 SAV_LCKINDX,
163 1152 2 STATUS, ! protection check status value
164 1153 2 FCB : REF BBLOCK, ! FCB address
165 1154 2 HEADER : REF BBLOCK; ! directory header address
166 1155 2
167 1156 2 BIND_COMMON;
168 1157 2
169 1158 2 EXTERNAL ROUTINE
170 1159 2 ARBITRATE_ACCESS : L JSB_2ARGS, ! access arbitration
171 1160 2 BUILD_EXT_FCBS : L_NORM NOVALUE, ! build extension headers
172 1161 2 CONV_ACCLOCK : L_NORM, ! convert access lock
173 1162 2 CREATE_FCB : L_NORM, ! create fcb
174 1163 2 MAKE_DIRINDX : L_NORM, ! validate/create directory index
175 1164 2 REBLD_PRIM_FCB : L_NORM NOVALUE, ! refresh fcb from header
176 1165 2 SERIAL_FILE : L_NORM, ! interlock directory processing.

```

```
177 1166 2 SWITCH_VOLUME : L_NORM, : switch to correct volume
178 1167 2 SEARCH_FCB : L_NORM, : search FCB list
179 1168 2 READ_HEADER : L_NORM, : read file header
180 1169 2 CHECK_PROTECT : L_NORM; : check file protection
181 1170
182 1171
183 1172 : Switch context to the volume of the specified RVN.
184 1173 :
185 1174
186 1175 SWITCH_VOLUME (.FIB[FIB$W_DID_RVN]);
187 1176
188 1177 : First find the FCB of the directory, if any exists.
189 1178 :
190 1179
191 1180 IF (FCB = .DIR_FCB) EQL 0
192 1181 THEN
193 1182 BEGIN
194 1183 LOCAL
195 1184 FCBVALID : INITIAL (0);
196 1185
197 1186 : Synchronize further file processing on this directory.
198 1187 :
199 1188
200 1189 SAV_LCKINDX = .CURR_LCKINDX;
201 1190
202 1191 DIR_LCKINDX = SERIAL_FILE (FIB [FIB$W_DID]);
203 1192
204 1193 FCB = SEARCH_FCB (FIB[FIB$W_DID]);
205 1194
206 1195 IF .FCB NEQ 0
207 1196 THEN
208 1197 FCBVALID = MAKE_DIRINDX (.FCB);
209 1198
210 1199 IF NOT .FCBVALID
211 1200 THEN
212 1201 BEGIN
213 1202
214 1203 : This flag tells READ_HEADER to not update the FILE_HEADER cell.
215 1204 :
216 1205
217 1206 STSFLGS [STS_LEAVE_FILEHDR] = 1;
218 1207
219 1208 HEADER = READ_HEADER (FIB [FIB$W_DID], .FCB);
220 1209
221 1210 IF .FCB EQL 0
222 1211 THEN
223 1212 BEGIN
224 1213 FCB = CREATE_FCB (.HEADER);
225 1214 MAKE_DIRINDX (.FCB);
226 1215 END
227 1216 ELSE
228 1217 REBLD_PRIM_FCB (.FCB, .HEADER);
229 1218
230 1219 IF .HEADER [FH2$W_EX_FIDNUM] NEQ 0
231 1220 OR .HEADER [FR2$B_EX_FIDNMX] NEQ 0
232 1221 THEN
233 1222 BUILD_EXT_FCBS (.HEADER, .FCB);
```

```

234      1223 4
235      1224 4 ! Check the file attributes to make sure that
236      1225 4 ! this is a real directory file.
237      1226 4
238      1227 4
239      1228 4     IF NOT .HEADER[FH2$V DIRECTORY]
240      1229 4         OR NOT .HEADER[FR2$V CONTIG]
241      1230 4         OR .BBLOCK [HEADER[FR2$W RECATTR], FAT$B_RTYPE] NEQ FAT$C_VARIABLE
242      1231 4         OR .BBLOCK [HEADER[FH2$W RECATTR], FAT$B_RATTRIB] NEQ FAT$M_NOSPAN
243      1232 4     THEN ERR_EXIT (SS$_BADIRECTORY);
244      1233 4
245      1234 4     FCB [FCB$W_DIRSEQ] = .FCB [FCB$W_DIRSEQ] + 1;
246      1235 4
247      1236 4     END;
248      1237 4
249      1238 4     CURR_LCKINDX = .SAV_LCKINDX;
250      1239 4
251      1240 4     IF .FCB [FCB$W_FID_SEQ] NEQ .FIB [FIB$W_DID_SEQ]
252      1241 4     THEN
253      1242 4         ERR_EXIT (SS$_FILESEQCHK);
254      1243 4
255      1244 4     DIR_FCB = .FCB;                                ! store global FCB address
256      1245 4
257      1246 4 ! Check directory file protection. For alternate access validation,
258      1247 4 ! delete access implies write to the directory.
259      1248 4
260      1249 4
261      1250 4     STATUS = CHECK_PROTECT (.WRITE, .HEADER, .FCB, 0,
262      1251 4         (IF .BBLOCK [FIB[FIB$V_ALT_ACCESS], ARMS$V_DELETE]
263      1252 4         THEN ARMS$M_WRITE ELSE 0),
264      1253 4         .FIB[FIB$V_ALT_REQ]);
265      1254 4
266      1255 4     IF .STATUS EQL SS$_NOTALLPRIV
267      1256 4     THEN FIB[FIB$V_ALT_GRANTED] = 0;
268      1257 4
269      1258 4 ! Arbitrate the access interlocks.
270      1259 4
271      1260 4
272      1261 4     CURR_LKMODE = .FCB [FCB$B_ACCLKMODE];
273      1262 4
274      1263 4     IF NOT ARBITRATE_ACCESS (IF .WRITE THEN FIB$M_WRITE ELSE 0, .FCB)
275      1264 4     THEN ERR_EXIT (SS$_ACCONFLICT);
276      1265 4
277      1266 4     CONV_ACCLOCK (.CURR_LKMODE, .FCB);
278      1267 4
279      1268 4     END;
280      1269 4
281      1270 4     RETURN 1;
282      1271 4
283      1272 4     ! end of routine DIR_ACCESS

```

```

.TITLE DIRACC
.IDENT \V04-000\

.EXTRN ARBITRATE_ACCESS
.EXTRN BUILD_EXT_FCBS, CONV_ACCLOCK

```



					.EXTRN	CREATE_FCB, MAKE_DIRINDX		
					.EXTRN	REBLD PRIM_FCB, SERIAL_FILE		
					.EXTRN	SWITCH_VOLUME, SEARCH_FCB		
					.EXTRN	READ_HEADER, CHECK_PROTECT		
					.PSECT	\$CODE\$,NOWRT,2		
					.ENTRY	DIR_ACCESS, Save R2,R3,R4		1109
					MOVL	FIB, R0		1175
					MOVZWL	14(R0), -(SP)		
					CALLS	#1, SWITCH_VOLUME		
					MOVL	208(BASE), -FCB		1180
					BEQL	1\$		
					RET			
					CLRL	FCBVALID		1182
					MOVL	20(BASE), SAV_LCKINDX		1189
					ADDL3	#10, FIB, -(SP)		1191
7E	04				CALLS	#1, SERIAL_FILE		
	0000G				MOVL	R0, 212(BASE)		
	00D4				ADDL3	#10, FIB, -(SP)		1193
7E	04				CALLS	#1, SEARCH_FCB		
	0000G				MOVL	R0, FCB		
					BEQL	2\$		1195
					PUSHL	FCB		1197
					CALLS	#1, MAKE_DIRINDX		
					MOVL	R0, FCBVALID		
					BLBS	FCBVALID, 9\$		1199
					BISB2	#8, -90(BASE)		1206
					PUSHL	FCB		1208
					ADDL3	#10, FIB, -(SP)		
					CALLS	#2, READ_HEADER		
					MOVL	R0, HEADER		
					TSTL	FCB		1210
					BNEQ	3\$		
					PUSHL	HEADER		1213
					CALLS	#1, CREATE_FCB		
					MOVL	R0, FCB		
					PUSHL	FCB		1214
					CALLS	#1, MAKE_DIRINDX		
					BRB	4\$		1210
					PUSHL	HEADER		1217
					PUSHL	FCB		
					CALLS	#2, REBLD PRIM_FCB		
					TSTW	14(HEADERT)		1219
					BNEQ	5\$		
					TSTB	19(HEADER)		1220
					BEQL	6\$		
					PUSHR	#*M<R2,R3>		1222
					CALLS	#2, BUILD_EXT_FCBS		
					BBC	#5, 53(HEADERT), 7\$		1228
					TSTB	52(HEADER)		1229
					BGEQ	7\$		
					CMPB	20(HEADER), #2		1230
					BNEQ	7\$		
					CMPB	21(HEADER), #8		1231
					BEQL	8\$		
					CHMU	#2088		1232



Library Statistics

File	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		
:_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	42 0	1000	00:01.8

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS:DIRACC/OBJ=OBJ\$:DIRACC MSRCS:DIRACC/UPDATE=(ENHS:DIRACC)

: Size: 284 code + 0 data bytes  
: Run Time: 00:18.8  
: Elapsed Time: 00:35.7  
: Lines/CPU Min: 4080  
: Lexemes/CPU-Min: 47273  
: Memory Used: 242 pages  
: Compilation Complete

