



```

SSSSSSSS NN NN DDDDDDDD BBBB8888 AAAAAA DDDDDDDD
SSSSSSSS NN NN DDDDDDDD BBBB8888 AAAAAA DDDDDDDD
SS NN NN DD DD BB BB AA AA DD DD
SS NN NN DD DD BB BB AA AA DD DD
SS NNNN NN DD DD BB BB AA AA DD DD
SS NNNN NN DD DD BB BB AA AA DD DD
SSSSSS NN NN DD DD BBBB8888 AA AA DD DD
SSSSSS NN NN DD DD BBBB8888 AA AA DD DD
SS NN NNNN DD DD BB BB AAAAAAAAAA DD DD
SS NN NNNN DD DD BB BB AAAAAAAAAA DD DD
SS NN NN DD DD BB BB AA AA DD DD
SSSSSSSS NN NN DDDDDDDD BBBB8888 AA AA DDDDDDDD
SSSSSSSS NN NN DDDDDDDD BBBB8888 AA AA DDDDDDDD

```

```

LL LL 111111 SSSSSSSS
LL LL 111111 SSSSSSSS
LL LL 11 SS
LL LL 11 SS
LL LL 11 SS
LL LL 11 SS
LL LL 11 SSSSSS
LL LL 11 SSSSSS
LL LL 11 SS
LL LL 11 SS
LL LL 11 SS
LLLLLLLLLLLL 111111 SSSSSSSS
LLLLLLLLLLLL 111111 SSSSSSSS

```



```

1 0001 0 MODULE SNDBAD (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000',
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
12 0012 1 * ALL RIGHTS RESERVED. *
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 2
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine sends a message to the bad block analysis program to
38 0038 1 deal with a file that is marked bad.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1
42 0042 1 STARLET operating system, including privileged system services
43 0043 1 and internal exec routines.
44 0044 1
45 0045 1 --
46 0046 1
47 0047 1
48 0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 26-May-1978 14:50
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V03-007 CDS0006 Christian D. Saether 2-July-1984
53 0053 1 Need to have enhanced privileges for $ASSIGN also.
54 0054 1
55 0055 1 V03-006 CDS0005 Christian D. Saether 20-Jun-1984
56 0056 1 Raise/lower process bioct and astcnt around
57 0057 1 $gio to mailbox so that it does not fail for
    
```

```
58 0058 1 Lack of those quotas.
59 0059 1
60 0060 1 V03-005 CDS0004 Christian D. Saether 30-Dec-1983
61 0061 1 Use L_NORM linkage and BIND_COMMON macro.
62 0062 1
63 0063 1 V03-004 CDS0003 Christian D. Saether 5-Oct-1983
64 0064 1 Fix bug restoring privs to PCB.
65 0065 1
66 0066 1 V03-003 CDS0002 Christian D. Saether 13-Jan-1983
67 0067 1 Separately save and restore PHD privs.
68 0068 1
69 0069 1 V03-002 CDS0001 Christian D. Saether 28-Dec-1982
70 0070 1 Give the process DETACH and SETPRV for the CREPRC of
71 0071 1 the bad block scanner, and BYPASS to assign channel
72 0072 1 to bad block scanner mailbox.
73 0073 1 Also use PIC_DESC instead of DESCRIPTOR.
74 0074 1
75 0075 1 V03-001 LMP0037 L. Mark Pilant, 28-Jun-1982 15:10
76 0076 1 Remove the addressing mode module switch.
77 0077 1
78 0078 1 V02-003 LMP0013 L. Mark Pilant, 15-Mar-1981 16:20
79 0079 1 Remove unused and unneeded storage (to fix Linker truncation
80 0080 1 errors).
81 0081 1
82 0082 1 V02-002 ACG0230 Andrew C. Goldstein, 24-Dec-1981 0:17
83 0083 1 Go to general mode addressing for externals
84 0084 1
85 0085 1 V02-001 ACG0167 Andrew C. Goldstein, 16-Apr-1980 19:28
86 0086 1 Previous revision history moved to F11B.REV
87 0087 1 **
88 0088 1
89 0089 1
90 0090 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
91 0091 1 REQUIRE 'SRC$:FCPDEF.B32';
```

```

93 1082 1 GLOBAL ROUTINE SEND_BADSCAN (FID) : L_NORM NOVALUE =
94 1083 1
95 1084 1 +-+
96 1085 1
97 1086 1 FUNCTIONAL DESCRIPTION:
98 1087 1
99 1088 1     This routine sends a message to the bad block analysis program to
100 1089 1     deal with a file that is marked bad.
101 1090 1
102 1091 1
103 1092 1 CALLING SEQUENCE:
104 1093 1     SEND_BADSCAN (ARG1)
105 1094 1
106 1095 1 INPUT PARAMETERS:
107 1096 1     ARG1: address of file ID of file
108 1097 1
109 1098 1 IMPLICIT INPUTS:
110 1099 1     CURRENT_UCB: UCB of device containing file
111 1100 1
112 1101 1 OUTPUT PARAMETERS:
113 1102 1     NONE
114 1103 1
115 1104 1 IMPLICIT OUTPUTS:
116 1105 1     NONE
117 1106 1
118 1107 1 ROUTINE VALUE:
119 1108 1     NONE
120 1109 1
121 1110 1 SIDE EFFECTS:
122 1111 1     bad block scan process started
123 1112 1
124 1113 1 --
125 1114 1
126 1115 2 BEGIN
127 1116 2
128 1117 2 MAP
129 1118 2     FID          : REF BBLOCK;    ! file ID argument
130 1119 2
131 1120 2 LOCAL
132 1121 2     PTR          : REF BBLOCK,
133 1122 2     SAVE_PRIV   : VECTOR [4],
134 1123 2     DESC0       : VECTOR [2],    ! descriptor
135 1124 2     MBX_CHANNEL : WORD;        ! channel number for mailbox
136 1125 2
137 1126 2 BIND_COMMON:
138 1127 2
139 1128 2 EXTERNAL
140 1129 2     CTL$GL_PCB   : ADDRESSING_MODE (GENERAL),
141 1130 2     CTL$GL_PHD   : ADDRESSING_MODE (GENERAL);
142 1131 2
143 1132 2 ! Assign a channel to the bad block scanner mailbox. Note that we simply
144 1133 2 ! give up on errors - the file will be left marked for delete and bad and
145 1134 2 ! can be picked up and retried later.
146 1135 2
147 1136 2
148 1137 2 PIC_DESC ('ACPSBADBLOCK_MBX', DESC0);
149 1138 2

```

```

150 1139 2 ! We don't need to raise bio, ast counts for the assign, only need detach
151 1140 2 ! for the creprc, but just do it all once for all the services that follow.
152 1141
153 1142
154 1143 PTR = .CTL$GL PCB;
155 1144 PTR [PCBSW_BIOCNT] = .PTR [PCBSW_BIOCNT] + 1;
156 1145 PTR [PCBSW_ASTCNT] = .PTR [PCBSW_ASTCNT] + 1;
157 1146 SAVE_PRIV [0] = .(PTR [PCBSQ_PRIV]);
158 1147 SAVE_PRIV [1] = .(PTR [PCBSQ_PRIV]+4);
159 1148 BBLOCK [ PTR [PCBSQ_PRIV], PRVSV_DETACH] = 1;
160 1149 BBLOCK [ PTR [PCBSQ_PRIV], PRVSV_SETPRV] = 1;
161 1150 BBLOCK [ PTR [PCBSQ_PRIV], PRVSV_BYPASS] = 1;
162 1151
163 1152 PTR = .CTL$GL PHD;
164 1153 SAVE_PRIV [2] = .(PTR [PHDSQ_PRIVMSK]);
165 1154 SAVE_PRIV [3] = .(PTR [PHDSQ_PRIVMSK]+4);
166 1155 BBLOCK [ PTR [PHDSQ_PRIVMSK], PRVSV_DETACH] = 1;
167 1156 BBLOCK [ PTR [PHDSQ_PRIVMSK], PRVSV_SETPRV] = 1;
168 1157 BBLOCK [ PTR [PHDSQ_PRIVMSK], PRVSV_BYPASS] = 1;
169 1158
170 P 1159 IF NOT $ASSIGN (CHAN = MBX_CHANNEL,
171 1160 DEVNAM = D$SCO)
172 1161 THEN
173 1162 BEGIN
174 1163 (PTR [PHDSQ_PRIVMSK]) = .SAVE_PRIV [2];
175 1164 (PTR [PHDSQ_PRIVMSK]+4) = .SAVE_PRIV [3];
176 1165
177 1166 PTR = .CTL$GL PCB;
178 1167 PTR [PCBSW_BIOCNT] = .PTR [PCBSW_BIOCNT] - 1;
179 1168 PTR [PCBSW_ASTCNT] = .PTR [PCBSW_ASTCNT] - 1;
180 1169 (PTR [PCBSQ_PRIV]) = .SAVE_PRIV [0];
181 1170 (PTR [PCBSQ_PRIV]+4) = .SAVE_PRIV [1];
182 1171
183 1172 RETURN
184 1173 END;
185 1174
186 1175 ! Send the message. Then attempt to create the bad block scan process. If one
187 1176 ! is already active, the create will fail due to duplicate process names,
188 1177 ! and the message will simply be queued.
189 1178
190 1179
191 P 1180 IF $QIO (CHAN = .MBX_CHANNEL,
192 P 1181 FUNC = IOS_WRITEBLK OR IOSM_NOW,
193 P 1182 EFN = MBX_EFN,
194 P 1183 P1 =
195 P 1184 BEGIN
196 P 1185 ! Construct the message in the message buffer.
197 P 1186 !
198 P 1187
199 P 1188 LOCAL
200 P 1189 MESSAGE : BBLOCK [BB$C_LENGTH]; ! message buffer
201 P 1190
202 P 1191 CH$FILL (0, BB$C_LENGTH, MESSAGE);
203 P 1192 MESSAGE[BB$B_MSGTYPE] = MSG$_SCANBAD;
204 P 1193 MESSAGE[BB$W_SEQUENCE] = 0;
205 P 1194 MESSAGE[BB$L_UCB] = .CURRENT_UCB;
206 P 1195 CH$MOVE (FID$C_LENGTH, .FID, MESSAGE[BB$W_FID]);

```

```

: 207 P 1196 2 MESSAGE
: 208 P 1197 2 END
: 209 P 1198 2 P2 = BBS$C_LENGTH
: 210 1199 )
: 211 1200
: 212 1201 THEN
: 213 1202 BEGIN
: 214 1203
: 215 1204 LOCAL
: 216 1205 DESC1 : VECTOR [2], ! descriptor
: 217 1206 DESC2 : VECTOR [2]; ! descriptor
: 218 1207
: 219 1208 PIC_DESC ('SYS$SYSTEM:BADBLOCK.EXE', DESC0);
: 220 1209 PIC_DESC ('TTA1:', DESC1);
: 221 1210 PIC_DESC ('BADBLOCK_SCAN', DESC2);
: 222 1211
: 223 P 1212 $CREPRC (
: 224 P 1213 IMAGE = DESC0,
: 225 P 1214 INPUT = DESC1,
: 226 P 1215 OUTPUT = DESC1,
: 227 P 1216 ERROR = DESC1,
: 228 P 1217 PRIVADR = UPLIT (-1, -1),
: 229 P 1218 PRCNAM = DESC2,
: 230 P 1219 BASPRI = 4,
: 231 P 1220 UIC = 1^16 + 3
: 232 1221 );
: 233 1222
: 234 1223 END;
: 235 1224
: 236 1225 $DASSGN (CHAN = .MBX_CHANNEL);
: 237 1226
: 238 1227 (PTR [PHD$Q_PRIVMSK]) = .SAVE_PRIV [2];
: 239 1228 (PTR [PHD$Q_PRIVMSK]+4) = .SAVE_PRIV [3];
: 240 1229
: 241 1230 PTR = .CTL$GL PCB;
: 242 1231 PTR [PCBSW_BIOCNT] = .PTR [PCBSW_BIOCNT] - 1;
: 243 1232 PTR [PCBSW_ASTCNT] = .PTR [PCBSW_ASTCNT] - 1;
: 244 1233 (PTR [PCBSQ_PRIV]) = .SAVE_PRIV [0];
: 245 1234 (PTR [PCBSQ_PRIV]+4) = .SAVE_PRIV [1];
: 246 1235
: 247 1236 1 END; ! end of routine SEND_BADSCAN

```

														.TITLE	SNDBAD		
														.IDENT	\V04-000\		
														.PSECT	\$CODE\$,NOWRT,2		
42	4D	5F	4B	43	4F	4C	42	44	41	42	24	50	43	41	00000	P.AAA: .ASCII	\ACPS\$BADBLOCK_MBX\
														58	0000F		
42	44	41	42	3A	4D	45	54	53	59	53	24	53	59	53	00010	P.AAB: .ASCII	\SYS\$SYSTEM:BADBLOCK.EXE\<0>
														00	0001F		
														00	00028	P.AAC: .ASCII	\TTA1:\<0><0>
00	00	4E	41	43	53	5F	4B	43	4F	4C	42	44	41	42	00030	P.AAD: .ASCII	\BADBLOCK_SCAN\<0><0><0>
														0C	0003F		
														00040	P.AAE: .LONG	-1, -1	





		44	AE	9F	000C9		PUSHAB	DESCO		
			7E	D4	000CC		CLRL	-(SP)		
00000000G	00		0D	FB	000CE		CALLS	#13, SYSS\$CREPRC		
	7E		6E	3C	000D5	2\$:	MOVZWL	MBX_CHANNEL, -(SP)		1225
00000000G	00		01	FB	000D8		CALLS	#1, -SYSS\$DASSGN		
	66	28	AE	7D	000DF	3\$:	MOVQ	SAVE_PRIV+8, (PTR)		1227
	56		68	D0	000E3		MOVL	CTL\$GL_PCB, PTR		1230
			3A	A6	B7	000E6	DECW	58(PTR)		1231
			38	A6	B7	000E9	DECW	56(PTR)		1232
0084	C6		20	AE	7D	000EC	MOVQ	SAVE_PRIV, 132(PTR)		1233
					04	000F2	RET			1236

: Routine Size: 243 bytes, Routine Base: \$CODE\$ + 0048

```

: 248          1237 1
: 249          1238 1 END
: 250          1239 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	315	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_S255\$DUA28:[SYSLIB]LIB.L32;1	18619	38 0	1000	00:01.9

COMMAND QUALIFIERS

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

```

: Size:          243 code + 72 data bytes
: Run Time:      00:20.2
: Elapsed Time: 00:38.3
: Lines/CPU Min: 3674
: Lexemes/CPU-Min: 51081
: Memory Used: 248 pages
: Compilation Complete

```

