


```

DDDDDDDD      EEEEEEEEEE      LL      BBBB8888      AAAAAA      DDDDDDDD
DDDDDDDD      EEEEEEEEEE      LL      BBBB8888      AAAAAA      DDDDDDDD
DD      DD      EE      LL      BB      BB      AA      AA      DD      DD
DD      DD      EE      LL      BB      BB      AA      AA      DD      DD
DD      DD      EE      LL      BB      BB      AA      AA      DD      DD
DD      DD      EE      LL      BB      BB      AA      AA      DD      DD
DD      DD      EEEEEEEE      LL      BBBB8888      AA      AA      DD      DD
DD      DD      EEEEEEEE      LL      BBBB8888      AA      AA      DD      DD
DD      DD      EE      LL      BB      BB      AAAAAAAAAA      DD      DD
DD      DD      EE      LL      BB      BB      AAAAAAAAAA      DD      DD
DD      DD      EE      LL      BB      BB      AA      AA      DD      DD
DD      DD      EE      LL      BB      BB      AA      AA      DD      DD
DD      DD      EE      LL      BB      BB      AA      AA      DD      DD
DD      DD      EE      LL      BB      BB      AA      AA      DD      DD
DDDDDDDD      EEEEEEEEEE      LLLLLLLLLL      BBBB8888      AA      AA      DDDDDDDD
DDDDDDDD      EEEEEEEEEE      LLLLLLLLLL      BBBB8888      AA      AA      DDDDDDDD

```

```

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 0001 0 MODULE DELBAD (
: 2 0002 0
:001 3 CDS0005 0003 0 LANGUAGE (BLISS32),
: 4 -1 0004 0 IDENT = 'V04-001'
: 5 0005 1 BEGIN
: 6 0006 1
: 7 0007 1
: 8 0008 1
: 9 0009 1
:10 0010 1 *****
:11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
:12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
:13 0013 1 * ALL RIGHTS RESERVED. *
:14 0014 1 *
:15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
:16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
:17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
:18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
:19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
:20 0020 1 * TRANSFERRED. *
:21 0021 1 *
:22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
:23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
:24 0024 1 * CORPORATION. *
:25 0025 1 *
:26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
:27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
: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 removes the indicated blocks from the given file header
:38 0038 1 and appends them to the bad block file.
: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: 29-May-1978 22:43
:49 0049 1
:50 0050 1 MODIFIED BY:
:51 0051 1
:001 52 CDS0005 0052 1 V04-001 CDS0005 Christian D. Saether 7-Nov-1984
:002 53 CDS0005 0053 1 Reread primary header to update hiblock and highwater
:003 54 CDS0005 0054 1 mark. Also update revision date.
:004 55 CDS0005 0055 1
: 52 0056 1 V03-006 CDS0004 Christian D. Saether 14-Aug-1984
: 53 0057 1 Remove obsolete reference to update_filesize routine.

```

```
.. 54 0058 1 |
.. 55 0059 1 |
.. 56 0060 1 |
.. 57 0061 1 |
.. 58 0062 1 |
.. 59 0063 1 |
.. 60 0064 1 |
.. 61 0065 1 |
.. 62 0066 1 |
.. 63 0067 1 |
.. 64 0068 1 |
.. 65 0069 1 |
.. 66 0070 1 |
.. 67 0071 1 |
.. 68 0072 1 |
.. 69 0073 1 |
.. 70 0074 1 |
.. 71 0075 1 |
.. 72 0076 1 |
.. 73 0077 1 |
.. 74 0078 1 |
.. 75 0079 1 |
.. 76 0080 1 |
.. 77 0081 1 |
.. 78 0082 1 |
.. 79 0083 1 |
.. 80 0084 1 |
V03-005 CDS0003 Christian D. Saether 31-July-1984
Remove local definition of get_map_pointer linkage.
V03-004 CDS0002 Christian D. Saether 2-May-1984
Perform deallocation to bad block file in secondary
context. Add appropriate serialization.
V03-003 CDS0001 Christian D. Saether 29-Dec-1983
Use L_NORM linkage and BIND_COMMON macro.
V03-002 ACG0367 Andrew C. Goldstein, 26-Oct-1983 19:49
Update BADBLK.SYS file highwater mark
V03-001 LMP0037 L. Mark Pilant, 28-Jun-1982 15:10
Remove the addressing mode module switch.
V02-003 ACG0230 Andrew C. Goldstein, 24-Dec-1981 0:16
Go to longword external addressing
V02-002 ACG0167 Andrew C. Goldstein, 16-Apr-1980 19:25
Previous revision history moved to F11B.REV
**
LIBRARY 'SYSS$LIBRARY:LIB.L32';
REQUIRE 'SRCS:FCPDEF.B32';
```

```

: 82      1075 1 GLOBAL ROUTINE DEALLOCATE_BAD (FIB, FILE_HDR, POINTER, LAST_COUNT) : L_NORM NOVALUE =
: 83      1076 1
: 84      1077 1 !++
: 85      1078 1
: 86      1079 1 FUNCTIONAL DESCRIPTION:
: 87      1080 1
: 88      1081 1     This routine removes the indicated blocks from the given file header
: 89      1082 1     and appends them to the bad block file.
: 90      1083 1
: 91      1084 1
: 92      1085 1 CALLING SEQUENCE:
: 93      1086 1     DEALLOCATE_BAD (ARG1, ARG2, ARG3, ARG4)
: 94      1087 1
: 95      1088 1 INPUT PARAMETERS:
: 96      1089 1     ARG1: address of user FIB
: 97      1090 1     ARG2: address of file header
: 98      1091 1     ARG3: address of map pointer at which to start
: 99      1092 1     ARG4: new value for last pointer block count
100      1093 1
101      1094 1 IMPLICIT INPUTS:
102      1095 1     NONE
103      1096 1
104      1097 1 OUTPUT PARAMETERS:
105      1098 1     NONE
106      1099 1
107      1100 1 IMPLICIT OUTPUTS:
108      1101 1     NONE
109      1102 1
110      1103 1 ROUTINE VALUE:
111      1104 1     NONE
112      1105 1
113      1106 1 SIDE EFFECTS:
114      1107 1     file header updated, bad block log file updated, bad block file extended
115      1108 1
116      1109 1 --
117      1110 1
118      1111 2 BEGIN
119      1112 2
120      1113 2 MAP
121      1114 2     FIB          : REF BBLOCK,  ! user FIB argument
122      1115 2     FILE_HDR     : REF BBLOCK;  ! address of file header
123      1116 2
124      1117 2 LINKAGE
125      1118 2     L_MAKE_POINTER = CALL :
126      1119 2     GLOBAL (BUILD_POINTER = 9);
127      1120 2
128      1121 2 GLOBAL REGISTER
129      1122 2     COUNT          = 6,          ! count of blocks returned
130      1123 2     LBN            = 7,          ! LBN of map entry
131      1124 2     MAP_POINTER    = 8 : REF BBLOCK, ! pointer to scan map
132      1125 2     BUILD_POINTER  = 9 : REF BBLOCK; ! pointer to build new map entry
133      1126 2
134      1127 2 LOCAL
:001 !CDS0005 1128 2     HEADER         : REF BBLOCK;  ! local address of file header
: 137-2      1129 2
: 138      1130 2 BIND
: 139      1131 2     BB_FID = UPLIT WORD (BADBLK_FID, BADBLK_FID, 0);

```

```

: 140 1132
: 141 1133
: 142 1134
: 143 1135
:001 :CDS0005 1136
:002 :CDS0005 1137
: 144 1138
: 145 1139
: 146 1140
: 147 1141
: 148 1142
: 149 1143
: 150 1144
: 151 1145
: 152 1146
: 154-1 1147
: 155 1148
: 156 1149
: 157 1150
: 158 1151
: 159 1152
: 160 1153
: 161 1154
: 162 1155
: 163 1156
: 164 1157
: 165 1158
: 166 1159
: 167 1160
: 168 1161
: 169 1162
: 170 1163
: 171 1164
: 172 1165
: 173 1166
: 174 1167
: 175 1168
: 176 1169
: 177 1170
: 178 1171
: 179 1172
: 180 1173
: 181 1174
: 182 1175
: 183 1176
: 184 1177
: 185 1178
:001 :CDS0005 1179
:002 :CDS0005 1180
:003 :CDS0005 1181
:004 :CDS0005 1182
:005 :CDS0005 1183
:006 :CDS0005 1184
:007 :CDS0005 1185
:008 :CDS0005 1186
:009 :CDS0005 1187
:010 :CDS0005 1188

BIND_COMMON;

EXTERNAL ROUTINE
  SET_REVISION      : L_NORM NOVALUE ADDRESSING_MODE (GENERAL),
                    :               set revision date
  SAVE_CONTEXT      : L_NORM,        : save primary context
  RESTORE_CONTEXT   : L_NORM,        : restore primary context
  SERIAL_FILE       : L_NORM,        : file serialization lock.
  RELEASE_SERIAL_LOCK : L_NORM,      : relinquish file serialization
  WRITE_DIRTY       : L_NORM,        : write modified buffers
  GET_MAP_POINTER   : L_MAP_POINTER, : get value of next map entry
  MAKE_POINTER      : L_MAKE_POINTER, : build new map entry
  NEXT_HEADER       : L_NORM,        : read next extension header
  MARK_DIRTY        : L_NORM,        : mark buffer for rewrite
  CHECKSUM          : L_NORM,        : compute file header checksum
  READ_HEADER       : L_NORM,        : read file header
  EXTEND_HEADER     : L_NORM,        : create extension header
  SCAN_BADLOG       : L_NORM,        : scan pending bad block log file

: Get into secondary context.
SAVE_CONTEXT ();

: Construct pointers into the file header and get the current contents of the
: last map pointer.
HEADER = .FILE_HDR;
MAP_POINTER = .POINTER;

GET_MAP_POINTER ();

: Now append the blocks to the bad block file.
LBN = .LBN + .LAST_COUNT;           ! compute LBN of bad cluster
COUNT = .COUNT - .LAST_COUNT;

: Serialize on the bad block file.
PRIM_LCKINDX = SERIAL_FILE (BB_FID);
HEADER = READ_HEADER (BB_FID, 0);

WHILE 1 DO
  BEGIN
    IF .HEADER [FH2$W_EX_FIDNUM] EQL 0
      AND .HEADER [FH2$B_EX_FIDNUM] EQL 0
    THEN
      EXITLOOP;

    HEADER = NEXT_HEADER (.HEADER, 0);
  END;

```

```
:011 :CDS0005 1189
:192-6 1190 MARK_DIRTY (.HEADER);
:193 1191 BUILD_POINTER = .HEADER + (.HEADER[FH2$B_MPOFFSET] + .HEADER[FH2$B_MAP_INUSE]) * 2;
:194 1192
:195 1193 IF NOT MAKE_POINTER (.COUNT, .LBN, .HEADER)
:196 1194 THEN
:197 1195 BEGIN
:198 1196 HEADER = EXTEND_HEADER (UPLIT_BYTE (REP FIB$C_LENGTH OF (0)), .HEADER, 0);
:199 1197 BUILD_POINTER = .HEADER + .HEADER[FH2$B_MPOFFSET] * 2;
:200 1198 MAKE_POINTER (.COUNT, .LBN, .HEADER);
:201 1199 END;
:202 1200
:001 :CDS0005 1201 ! If we have gotten into extension headers, we need to re-read the primary
:002 :CDS0005 1202 header to update fields there. Checksum the current header if that is
:003 :CDS0005 1203 the case.
:004 :CDS0005 1204
:005 :CDS0005 1205
:006 :CDS0005 1206 IF .HEADER [FH2$W_SEG_NUM] NEQ 0
:007 :CDS0005 1207 THEN
:008 :CDS0005 1208 BEGIN
:009 :CDS0005 1209 CHECKSUM (.HEADER);
:010 :CDS0005 1210
:011 :CDS0005 1211 HEADER = READ_HEADER (BB_FID, 0);
:012 :CDS0005 1212
:013 :CDS0005 1213 MARK_DIRTY (.HEADER);
:014 :CDS0005 1214 END;
:015 :CDS0005 1215
:016 :CDS0005 1216 SET_REVISION (.HEADER, 1);
:017 :CDS0005 1217
:203 1218 BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_HIBLK] =
:204 1219 ROT (ROT (.BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_HIBLK], 16) + .COUNT, 16);
:205 1220
:206 1221 ! If this file header supports it, stuff the high water field to
:207 1222 be the allocated size.
:208 1223
:209 1224
:210 1225 IF .HEADER [FH2$B_IDOFFSET] GEQU ($BYTEOFFSET (FH2$L_HIGHWATER)+4)/2
:211 1226 THEN
:212 1227 HEADER [FH2$L_HIGHWATER] = ROT (.BBLOCK [HEADER[FH2$W_RECATTR], FAT$L_HIBLK], 16) + 1;
:213 1228
:214 1229 CHECKSUM (.HEADER);
:215 1230
:216 1231 ! Write the modified header(s), release the serialization lock, and return to
:217 1232 primary context.
:218 1233
:219 1234
:220 1235 WRITE_DIRTY (.LB_BASIS [.PRIM_LCKINDX]);
:221 1236
:222 1237 RELEASE_SERIAL_LOCK (.PRIM_LCKINDX);
:223 1238
:224 1239 RESTORE_CONTEXT ();
:225 1240
:226 1241 ! Finally, remove the bad block cluster from the volume pending bad block log
:227 1242 file, if it was there.
:228 1243
:229 1244
:230 1245 SCAN_BADLOG (0, 0, .LBN, REMOVE_BADBLOCK, .COUNT);
```

: 231 1246 2
: 232 1247 1 END;

! end of routine DEALLOCATE_BAD

					.TITLE	DELBAD		
					.IDENT	\V04-001\		
					.PSECT	\$CODES,NOWRT,2		
0000	0003	0003	00000	P.AAA:	.WORD	3,3,0	:	
		00#	00006	P.AAB:	.BYTE	0[64]	:	
				BB_FID=		P.AAA		
				.EXTRN	SET_REVISION, SAVE_CONTEXT			
				.EXTRN	RESTORE_CONTEXT			
				.EXTRN	SERIAL_FILE, RELEASE_SERIAL_LOCK			
				.EXTRN	WRITE_DIRTY, GET_MAP_POINTER			
				.EXTRN	MAKE_POINTER, NEXT_HEADER			
				.EXTRN	MARK_DIRTY, CHECKSUM			
				.EXTRN	READ_HEADER, EXTEND_HEADER			
				.EXTRN	SCAN_BADLOG			
				.ENTRY	DEALLOCATE_BAD, Save R2,R3,R6,R7,R8,R9		:	1075
0000G	53	B5	AF 9E 00002		MOVAB	BB_FID, R3	:	
	CF		00 FB 00006		CALLS	#0, SAVE_CONTEXT	:	1156
	52	08	AC D0 00001		MOVL	FILE_HDR, HEADER	:	1162
	58	0C	AC D0 00001		MOVL	POINTER, MAP_POINTER	:	1163
			0000G 30 00013		BSBW	GET_MAP_POINTER	:	1165
	57	10	AC C0 00016		ADDL2	LAST_COUNT, LBN	:	1170
	56	10	AC C2 0001A		SUBL2	LAST_COUNT, COUNT	:	1171
			53 DD 0001E		PUSHL	R3	:	1176
0000G	CF		01 FB 00020		CALLS	#1, SERIAL_FILE	:	
	18	AA	50 D0 00025		MOVL	R0, 24(BASE)	:	
			7E D4 00029		CLRL	-(SP)	:	1178
			53 DD 0002B		PUSHL	R3	:	
0000G	CF		02 FB 0002D		CALLS	#2, READ_HEADER	:	
	52		50 D0 00032	1\$:	MOVL	R0, HEADER	:	
		0E	A2 B5 00035		TSTW	14(HEADER)	:	1182
			05 12 00038		BNEQ	2\$:	
		13	A2 95 0003A		TSTB	19(HEADER)	:	1183
			0B 13 0003D		BEQL	3\$:	
			7E D4 0003F	2\$:	CLRL	-(SP)	:	1187
			52 DD 00041		PUSHL	HEADER	:	
0000G	CF		02 FB 00043		CALLS	#2, NEXT_HEADER	:	
			E8 11 00048		BRB	1\$:	
			52 DD 0004A	3\$:	PUSHL	HEADER	:	1190
0000G	CF		01 FB 0004C		CALLS	#1, MARK_DIRTY	:	
	50	01	A2 9A 00051		MOVZBL	1(HEADER), R0	:	1191
	51	3A	A2 9A 00055		MOVZBL	58(HEADER), R1	:	
	50		51 C0 00059		ADDL2	R1, R0	:	
	59		6240 3E 0005C		MOVAW	(HEADER)[R0], BUILD_POINTER	:	
			52 DD 00060		PUSHL	HEADER	:	1193
	7E		56 7D 00062		MOVQ	COUNT, -(SP)	:	
0000G	CF		03 FB 00065		CALLS	#3, MAKE_POINTER	:	
	21		50 E8 0006A		BLBS	R0, 4\$:	
			7E D4 0006D		CLRL	-(SP)	:	1196
			52 DD 0006F		PUSHL	HEADER	:	

			06	A3	9F	00071	PUSHAB	P AAB		
	0000G	CF		03	FB	00074	CALLS	#3, EXTEND_HEADER		
		52		50	DD	00079	MOVL	R0, HEADER		
		50	01	A2	9A	0007C	MOVZBL	1(HEADER), R0		1197
		59		6240	3E	00080	MOVAV	(HEADER)[R0], BUILD_POINTER		
				52	DD	00084	PUSHL	HEADER		1198
		7E		56	7D	00086	MOVQ	COUNT, -(SP)		
	0000G	CF		03	FB	00089	CALLS	#3, MAKE_POINTER		
			04	A2	B5	0008E	TSTW	4(HEADERT		1206
				1A	13	00091	BEQL	5\$		
				52	DD	00093	PUSHL	HEADER		1209
	0000G	CF		01	FB	00095	CALLS	#1, CHECKSUM		
				7E	D4	0009A	CLRL	-(SP)		1211
				53	DD	0009C	PUSHL	R3		
	0000G	CF		02	FB	0009E	CALLS	#2, READ_HEADER		
		52		50	DD	000A3	MOVL	R0, HEADER		
				52	DD	000A6	PUSHL	HEADER		1213
	0000G	CF		01	FB	000A8	CALLS	#1, MARK_DIRTY		
				01	DD	000AD	PUSHL	#1		1216
				52	DD	000AF	PUSHL	HEADER		
	00000000G	00		02	FB	000B1	CALLS	#2, SET_REVISION		
	50	18		10	9C	000B8	ROTL	#16, 24(HEADER), R0		1219
				56	C0	000BD	ADDL2	COUNT, R0		
18	A2			10	9C	000C0	ROTL	#16, R0, 24(HEADER)		
				62	91	000C5	CMPB	(HEADER), #40		1225
				0A	1F	000C8	BLSSU	6\$		
	50	18		10	9C	000CA	ROTL	#16, 24(HEADER), R0		1227
		4C	A2	01	A0	9E	MOVAB	1(R0), 76(HEADER)		
				52	DD	000D4	PUSHL	HEADER		1229
	0000G	CF		01	FB	000D6	CALLS	#1, CHECKSUM		
		50	18	AA	D0	000DB	MOVL	24(BASE), R0		1235
			0080	CA40	DD	000DF	PUSHL	128(BASE)[R0]		
	0000G	CF		01	FB	000E4	CALLS	#1, WRITE_DIRTY		
			18	AA	DD	000E9	PUSHL	24(BASE)		1237
	0000G	CF		01	FB	000EC	CALLS	#1, RELEASE_SERIAL_LOCK		
	0000G	CF		00	FB	000F1	CALLS	#0, RESTORE_CONTEXT		1239
				56	DD	000F6	PUSHL	COUNT		1245
				7E	D4	000F8	CLRL	-(SP)		
				57	DD	000FA	PUSHL	LBN		
				7E	7C	000FC	CLRQ	-(SP)		
	0000G	CF		05	FB	000FE	CALLS	#5, SCAN_BADLOG		
				04	00	00103	RET			1247

; Routine Size: 260 bytes, Routine Base: \$CODE\$ + 0046

```

: 233      1248 1
: 234      1249 1 END
: 235      1250 0 ELUDOM

```

PSECT SUMMARY

```
: Name Bytes Attributes
: $CODE$ 330 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
```

Library Statistics

```
: File Total Symbols Loaded Percent Pages Mapped Processing Time
: _$255$DUA18:[SYSLIB]LIB.L32;1 18619 27 0 1000 00:02.0
```

COMMAND QUALIFIERS

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

```
: Size: 260 code + 70 data bytes
: Run Time: 00:19.7
: Elapsed Time: 00:35.8
: Lines/CPU Min: 3803
: Lexemes/CPU-Min: 44272
: Memory Used: 232 pages
: Compilation Complete
```

0443 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

0443 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

DELBAD
LIS

QUOTAUTIL
LIS

CREHDR
LIS

RDBLOK
LIS

MAKACC
LIS

DEACCS
LIS

EXTHDR
LIS

CREATE
LIS