


```

1 0001 0 MODULE SELVOL (
2 0002 0     LANGUAGE (BLISS32),
3 0003 0     IDENT = 'V04-001'
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 selects a suitable volume for the creation of a file
38 0038 1     or the continuation of a file on some other volume.
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: 21-Nov-1978 16:59
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1     V04-001 ACG0464 Andrew C. Goldstein, 7-Sep-1984 17:22
53 0053 1     Rework to function in a cluster and process based environment
54 0054 1
55 0055 1     V03-002 ACG0407 Andrew C. Goldstein, 19-Mar-1984 14:53
56 0056 1     Dispose of GETACC routine
57 0057 1

```

SELVOL
V04-001

K 1
16-Sep-1984 01:09:23
14-Sep-1984 12:30:46

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11X.SRC]SELVOL.B32;5 Page (1)

```
58 0058 1 | V03-001 CDS0001 Christian D. Saether 2-Jan-1984
59 0059 1 | Use L_NORM linkage and BIND_COMMON macro.
60 0060 1 |
61 0061 1 | B0104 ACG0082 Andrew C. Goldstein, 8-Nov-1979 22:25
62 0062 1 | Skip over write locked volumes
63 0063 1 |
64 0064 1 | B0103 ACG0071 Andrew C. Goldstein, 12-Oct-1979 10:58
65 0065 1 | Range check placement RVN in volume selection
66 0066 1 |
67 0067 1 | B0102 ACG0039 Andrew C. Goldstein, 16-May-1979 13:02
68 0068 1 | Do correct error exit on contig allocation failure
69 0069 1 |
70 0070 1 | B0101 ACG0008 Andrew C. Goldstein, 26-Dec-1978 18:32
71 0071 1 | Add placement control support
72 0072 1 |
73 0073 1 | **
74 0074 1 |
75 0075 1 |
76 0076 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';
77 0077 1 | REQUIRE 'SRC$:FCPDEF.B32';
```

```

79 1068 1 GLOBAL ROUTINE SELECT_VOLUME (FIB, BLOCKS_NEEDED) : L_NORM NOVALUE =
80 1069 1
81 1070 1 ++
82 1071 1
83 1072 1 FUNCTIONAL DESCRIPTION:
84 1073 1
85 1074 1 This routine scans the RVT for the volume with the most free space,
86 1075 1 or, if a contiguous allocation is asked for, the volume with the
87 1076 1 most free space and sufficient contiguous space.
88 1077 1
89 1078 1
90 1079 1 CALLING SEQUENCE:
91 1080 1 SELECT_VOLUME (ARG1, ARG2)
92 1081 1
93 1082 1 INPUT PARAMETERS:
94 1083 1 ARG1: address of user FIB
95 1084 1 ARG2: number of blocks to be allocated
96 1085 1
97 1086 1 IMPLICIT INPUTS:
98 1087 1 LOC_RVN: placement RVN or 0
99 1088 1 CURRENT_VCB: VCB of current volume
100 1089 1
101 1090 1 OUTPUT PARAMETERS:
102 1091 1 NONE
103 1092 1
104 1093 1 IMPLICIT OUTPUTS:
105 1094 1 CURRENT_UCB, CURRENT_VCB, CURRENT_RVN: set to volume switched to
106 1095 1 UNREC_COUNT, UNREC_BLOCKS: count and LBN of blocks preallocated, if any
107 1096 1
108 1097 1 ROUTINE VALUE:
109 1098 1 NONE
110 1099 1
111 1100 1 SIDE EFFECTS:
112 1101 1 context switched to new volume, blocks may be allocated
113 1102 1
114 1103 1 --
115 1104 1
116 1105 2 BEGIN
117 1106 2
118 1107 2 MAP
119 1108 2 FIB : REF BBLOCK; ! user FIB arg
120 1109 2
121 1110 2 LOCAL
122 1111 2 STATUS, ! error status to return
123 1112 2 BEST_SIZE, ! largest volume of current scan
124 1113 2 BEST_RVN, ! RVN of above volume
125 1114 2 TRIED_IT : BITVECTOR [256], ! vector of volumes tried so far
126 1115 2 RVT : REF BBLOCK, ! address of relative volume table
127 1116 2 UCB : REF BBLOCK, ! UCB under consideration
128 1117 2 VCB : REF BBLOCK; ! VCB under consideration
129 1118 2
130 1119 2 BIND_COMMON;
131 1120 2
132 1121 2 EXTERNAL ROUTINE
133 1122 2 ALLOCATION_LOCK : L_NORM, ! acquire volume lock
134 1123 2 SWITCH_VOLUME : L_NORM, ! switch context to new volume
135 1124 2 ALLOC_BLOCKS : L_NORM; ! allocate blocks from storage map

```

```
136 1125 2
137 1126 2
138 1127 2 ! We scan the volumes of the volume set in reverse size order. If a non-
139 1128 2 ! contiguous allocation is being done, we simply return with the volume with
140 1129 2 ! the most free space. If a contiguous request is made, try to do the allocation
141 1130 2 ! on each volume until it succeeds. The first pass (J = 0) is used to
142 1131 2 ! process RVN placement, if given.
143 1132 2
144 1133 2
145 1134 2 ALLOCATION_LOCK ();
146 1135 2 RVT = .CURRENT_VCB[VCB$$_RVT];
147 1136 2 IF .RVT EQL .CURRENT_UCB THEN RETURN; ! noop if not a volume set
148 1137 2
149 1138 2 IF .LOC_RVN GTRU .RVT[RVT$$_NVOLS] ! discard garbage RVN's
150 1139 2 THEN LOC_RVN = 0;
151 1140 2
152 1141 2 CH$FILL (0, 256/8, TRIED_IT);
153 1142 2
154 1143 2 INCR J FROM (.LOC_RVN EQL 0) TO .RVT[RVT$$_NVOLS]
155 1144 2 DO
156 1145 2 BEGIN
157 1146 2 BEST_SIZE = 0;
158 1147 2 BEST_RVN = 0;
159 1148 2
160 1149 2 ! The inner loop scans the RVT for the volume (mounted) with the most free
161 1150 2 ! which we haven't tried yet. We take out the allocation lock on each
162 1151 2 ! volume before looking at it (by calling SWITCH_VOLUME) to get an up to
163 1152 2 ! date copy of the volume's free space.
164 1153 2
165 1154 2
166 1155 4 INCR K FROM (IF .J EQL 0 THEN .LOC_RVN ELSE 1)
167 1156 4 TO (IF .J EQL 0 THEN .LOC_RVN ELSE .RVT[RVT$$_NVOLS])
168 1157 3 DO
169 1158 4 BEGIN
170 1159 4 UCB = .VECTOR [RVT[RVT$$_UCBLST], .K-1];
171 1160 4 IF .UCB NEQ 0
172 1161 4 THEN
173 1162 5 BEGIN
174 1163 5 VCB = .UCB[UCB$$_VCB];
175 1164 5 SWITCH_VOLUME (.R);
176 1165 5 IF .VCB[VCB$$_FREE] GTRU .BEST_SIZE
177 1166 5 AND NOT .TRIED_IT[.K]
178 1167 5 THEN
179 1168 6 BEGIN
180 1169 6 BEST_SIZE = .VCB[VCB$$_FREE];
181 1170 6 BEST_RVN = .K;
182 1171 5 END;
183 1172 4 END;
184 1173 3 END;
185 1174 2
186 1175 2 ! Having picked a volume, check it for usefulness. A size of zero means the
187 1176 2 ! whole volume set is full. If we are trying for contiguous space, check if
188 1177 2 ! there is at least that much space and try the allocation.
189 1178 2
190 1179 2
191 1180 3 TRIED_IT[.BEST_RVN] = 1;
192 1181 4 IF (
```

```

193 1182 4 IF .FIB[FIB$V_ALCON]
194 1183 4 THEN .BEST_SIZE LSSU .BLOCKS_NEEDED
195 1184 4 ELSE .BEST_SIZE EQL 0
196 1185 4 )
197 1186 3 THEN
198 1187 4 BEGIN
199 1188 4 IF .J NEQ 0
200 1189 4 THEN EXITLOOP;
201 1190 4 END
202 1191 4
203 1192 3 ELSE
204 1193 4 BEGIN
205 1194 4 SWITCH VOLUME (.BEST_RVN);
206 1195 4 UNREC_RVN = .BEST_RVN;
207 1196 5 IF (
208 1197 5 IF .BLOCKS_NEEDED NEQ 0
209 1198 5 THEN ALLOC_BLOCKS (.FIB, .BLOCKS_NEEDED, UNREC_LBN, UNREC_COUNT)
210 1199 5 ELSE 1
211 1200 5 )
212 1201 4 THEN RETURN;
213 1202 3 END;
214 1203 3
215 1204 3 LOC_RVN = 0; ! discard placement after first try
216 1205 3 LOC_LBN = 0;
217 1206 2 END; ! end of outer retry loop
218 1207 2
219 1208 2 ! We exit or fall out of the loop if we have tried all volumes in the set
220 1209 2 ! that seemed worth trying, and couldn't get anything.
221 1210 2
222 1211 2
223 1212 2 ERR_EXIT (SS$_DEVICEFULL);
224 1213 2
225 1 14 1 END; ! end of routine SELECT_VOLUME

```

.TITLE	SELVOL			
.IDENT	\V04-001\			
.EXTRN	ALLOCATION_LOCK			
.EXTRN	SWITCH_VOLUME, ALLOC_BLOCKS			
.PSECT	\$CODE\$,NOWRT,2			
.ENTRY	SELECT_VOLUME, Save R2,R3,R4,R5,R6,R7,R8,-			: 1068
	R9,R11			
	#36, SP			
SUBL2	29(BASE), R11			: 1117
MOVAB	#0, ALLOCATION_LOCK			: 1134
CALLS	-10(BASE), R0			: 1135
MOVL	32(:0), RVT			
MOVL	RVT, -108(BASE)			: 1136
CMPL	1\$			
BNEQ				
RET				
CMPZV	#0, #8, 11(RVT), (R11)			: 1138
BGEQU	2\$			
CLRL	(R11)			: 1139
MOVC5	#0, (SP), #0, #32, TRIED_IT			: 1141

				OBFC 0000
	SE		24	C2 00002
	5B	1C	AA	9E 00005
	0000G		00	FB 00009
	50	98	AA	D0 0000E
	56	20	A0	D0 00012
	94		56	D1 00016
			01	12 0001A
				04 0001C
6B		0B	A6	08
				00 ED 0001D 1\$:
				02 1E 00023
				6B D4 00025
20			00	2C 00027 2\$:

SELVOL
V04-001

C 2
16-Sep-1984 01:09:23
14-Sep-1984 12:30:46

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11X.SRC]SELVOL.B32;3 Page 7
(2)

SH
VO

0850 8F BF 000D5 17\$: CHMU #2128
04 000D9 18\$: RET

: 1212
: 1214

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

: 226 1215 1
: 227 1216 1 END
: 228 1217 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		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	24 0	1000	00:01.9

COMMAND QUALIFIERS

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

: Size: 218 code + 0 data bytes
: Run Time: 00:18.1
: Elapsed Time: 00:35.6
: Lines/CPU Min: 4032
: Lexemes/CPU-Min: 47463
: Memory Used: 223 pages
: Compilation Complete

SCHFIB LIS

SND5MB LIS

SHFDIR LIS

SNDER LIS

TRUNC LIS

FAL

FAL MAP

SELVOL LIS

DAPDEF MDL

SMALOC LIS

SNOBAD LIS

SWTUL LIS

WTURN LIS