

SSSSSSSS	YY	YY	MM	MM	88888888	IIIIII	000000	NN	NN	TTTTTTTTTT	
SSSSSSSS	YY	YY	MM	MM	88888888	IIIIII	000000	NN	NN	TTTTTTTTTT	
SS	YY	YY	MMM	MMM	BB	BB	00	00	NN	NN	TT
SS	YY	YY	MMM	MMM	BB	BB	00	00	NN	NN	TT
SS	YY	YY	MM	MM	BB	BB	00	00	NNNN	NN	TT
SSSSSS	YY	YY	MM	MM	BB	BB	00	00	NNNN	NN	TT
SSSSSS	YY	YY	MM	MM	BB	BB	00	00	NN	NN	TT
SS	YY	YY	MM	MM	BB	BB	00	00	NN	NN	TT
SS	YY	YY	MM	MM	BB	BB	00	00	NN	NN	TT
SS	YY	YY	MM	MM	BB	BB	00	00	NN	NN	TT
SSSSSSSS	YY	YY	MM	MM	88888888	IIIIII	000000	NN	NN	TT
SSSSSSSS	YY	YY	MM	MM	88888888	IIIIII	000000	NN	NN	TT

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 SYMBIONT (%TITLE 'Symbiont communication'
2 0002 0 IDENT = 'V04-000'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 !*****
7 0007 1 !*
8 0008 1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 !*  ALL RIGHTS RESERVED.
11 0011 1 !*
12 0012 1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 !*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 !*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 !*  TRANSFERRED.
18 0018 1 !*
19 0019 1 !*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 !*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 !*  CORPORATION.
22 0022 1 !*
23 0023 1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 !*
26 0026 1 !*
27 0027 1 !*****
28 0028 1
29 0029 1 ++
30 0030 1 FACILITY:
31 0031 1 Job controller.
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1 This module contains the routines that communicate with symbionts.
35 0035 1
36 0036 1 ENVIRONMENT:
37 0037 1 VAX/VMS user and kernel mode.
38 0038 1 --
39 0039 1
40 0040 1 AUTHOR: M. Jack, CREATION DATE: 16-Feb-1982
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 V03-016 JAK0232 J A Krycka 31-Aug-1984
45 0045 1 Ensure that the display of the error message associated with
46 0046 1 a symbiont deletion error message is not inhibited.
47 0047 1
48 0048 1 V03-015 JAK0228 J A Krycka 30-Aug-1984
49 0049 1 Temporarily disable the pausing of a output queue upon
50 0050 1 processing an operator request message.
51 0051 1
52 0052 1 V03-014 JAK0220 J A Krycka 18-Jul-1984
53 0053 1 Support SJCS_PAGINATE at the queue level in addition to the
54 0054 1 job and file-levels.
55 0055 1
56 0056 1 V03-013 JAK0219 J A Krycka 17-Jul-1984
57 0057 1 Track changes in JOBCTLDEF.REQ.

```

```
58 0058 1  
59 0059 1 V03-012 JAK0206 J A Krycka 06-May-1984  
60 0060 1 Conditionally request image dump for symbiont process.  
61 0061 1  
62 0062 1 V03-011 GRR0011 Gregory R. Robert 19-Apr-1984  
63 0063 1 Enable image dump for symbiont process.  
64 0064 1  
65 0065 1 V03-010 JAK0200 J A Krycka 15-Mar-1984  
66 0066 1 Add IOSM_NORSWAIT function modifier to mailbox write.  
67 0067 1  
68 0068 1 V03-009 GRR0008 Gregory R. Robert 26-Sep-1983  
69 0069 1 Remove GRR0005 (LIB is already refereced in JOBCTLDEF).  
70 0070 1  
71 0071 1 V03-008 GRR0005 Gregory R. Robert 26-Sep-1983  
72 0072 1 Fetch symbiont definitions directly from LIB.  
73 0073 1  
74 0074 1 V03-007 MLJ0118 Martin L. Jack, 23-Aug-1983  
75 0075 1 Change field names, track symbiont changes.  
76 0076 1  
77 0077 1 V03-006 MLJ0115 Martin L. Jack, 30-Jul-1983  
78 0078 1 Changes for job controller baselevel.  
79 0079 1  
80 0080 1 V03-005 MLJ0114 Martin L. Jack, 23-Jun-1983  
81 0081 1 Changes for job controller baselevel.  
82 0082 1  
83 0083 1 V03-004 MLJ0113 Martin L. Jack, 26-May-1983  
84 0084 1 Changes for job controller baselevel.  
85 0085 1  
86 0086 1 V03-003 MLJ0112 Martin L. Jack, 29-Apr-1983  
87 0087 1 Changes for job controller and print symbiont baselevel.  
88 0088 1  
89 0089 1 V03-002 MLJ0110 Martin L. Jack, 18-Apr-1983  
90 0090 1 Correct failure to set stopped state in STOP_SYMBIONT_STREAM.  
91 0091 1  
92 0092 1 V03-001 MLJ0109 Martin L. Jack, 14-Apr-1983  
93 0093 1 Changes for job controller baselevel.  
94 0094 1  
95 0095 1 !**
```

```

: 97 0096 1 REQUIRE 'SRCS:JOBCTLDEF'; . Job controller definitions
: 98 1137 1
: 99 1138 1
100 1139 1 FORWARD ROUTINE
101 1140 1 OPERATOR_REQUEST_ACTION,
102 1141 1 OPERATOR_REQUEST: NOVALUE,
103 1142 1 SEND_SYMBIONT_MESSAGE: NOVALUE,
104 1143 1 START_SYMBIONT_TASK: NOVALUE,
105 1144 1 STOP_SYMBIONT_TASK: NOVALUE,
106 1145 1 PAUSE_SYMBIONT_TASK: NOVALUE,
107 1146 1 RESUME_SYMBIONT_TASK: NOVALUE,
108 1147 1 START_SYMBIONT_STREAM,
109 1148 1 STOP_SYMBIONT_STREAM: NOVALUE,
110 1149 1 RESET_SYMBIONT_STREAM: NOVALUE,
111 1150 1 PROCESS_SYMBIONT_MESSAGE: NOVALUE,
112 1151 1 SYMBIONT_SERVICE: NOVALUE,
113 1152 1 SYMBIONT_DELETION: NOVALUE,
114 1153 1 DELETE_SYMBIONTS: NOVALUE,
115 1154 1 SYMBIONT_COMPLETED_BLOCKS;
116 1155 1
117 1156 1
118 1157 1 EXTERNAL ROUTINE
119 1158 1 ALLOCATE_MEMORY,
120 1159 1 COMPLETE_JOB: NOVALUE,
121 1160 1 DEALLOCATE_MEMORY: NOVALUE,
122 1161 1 DEALLOCATE_VARIABLE_DATA: NOVALUE,
123 1162 1 ENQUEUE_JOB: L_OUTPUT_2 NOVALUE,
124 1163 1 ENTER_PROCESS_DATA: NOVALUE,
125 1164 1 FETCH_VARIABLE_DATA: NOVALUE,
126 1165 1 FETCH_VARIABLE_ITEM,
127 1166 1 FETCH_VARIABLE_ITEM_LIST,
128 1167 1 FIND_PENDING_JOBS: NOVALUE,
129 1168 1 LOCK_QUEUE_FILE: NOVALUE,
130 1169 1 READ_RECORD,
131 1170 1 RELEASE_RECORD: NOVALUE,
132 1171 1 REWRITE_RECORD: NOVALUE,
133 1172 1 SCAN_INCOMPLETE_SERVICES: NOVALUE,
134 1173 1 STORE_VARIABLE_DATA,
135 1174 1 UNLOCK_QUEUE_FILE: NOVALUE,
136 1175 1 UPDATE_GETQUIT_DATA: NOVALUE,
137 1176 1
138 1177 1
139 1178 1 EXTERNAL
140 1179 1 JOBCTLMBX_DESC,
141 1180 1 NLAO_DESC,
142 1181 1 OPAO_DESC,
143 1182 1
144 1183 1
145 1184 1 ! Symbiont control table.
146 1185 1 !
147 1186 1 MACRO
148 1187 1 SCT_L_FLINK= 0,0,32,0 % ! Link to next SCT
149 1188 1 SCT_V_DELETING= 4,0,1,0 % ! Symbiont is deleting itself
150 1189 1 SCT_B_MAXSTREAMS= 5,0,8,0 % ! Maximum active streams
151 1190 1 SCT_W_MAILBOX= 6,0,16,0 % ! Unit number of mailbox
152 1191 1 SCT_L_PID= 8,0,32,0 % ! PID of symbiont process
153 1192 1 SCT_L_BITMAP= 12,0,32,0 % ! Stream index allocation bitmap

```

:	154	1193	1	SCT_L_RESETTING=	16,0,32,0 %	:	Stream resetting bitmap
:	155	1194	1	SCT_T_PROCESSOR=	20,0,0,0 %	:	Image filename (ASCII)
:	156	1195	1	SCT_L_QUEUES=	60,0,0,0 %	:	Base of 32 SMQ pointers
:	157	1196	1				
:	158	1197	1				
:	159	1198	1	LITERAL			
:	160	1199	1	SCT_K_MAXSTREAMS=	32;	:	Maximum active streams
:	161	1200	1				
:	162	1201	1				
:	163	1202	1	BUILTIN			
:	164	1203	1	FFC,			
:	165	1204	1	MOV3,			
:	166	1205	1	TESTBITSC;			

```
168 1206 1 ROUTINE OPERATOR_REQUEST_ACTION(MSG_DESC)=
169 1207 1
170 1208 1 !++
171 1209 1
172 1210 1 FUNCTIONAL DESCRIPTION:
173 1211 1 This is an action routine for the $PUTMSG that issues an operator
174 1212 1 request to the printer operator. It writes the record to the operator
175 1213 1 via OPCOM or via broadcast.
176 1214 1
177 1215 1 INPUT PARAMETERS:
178 1216 1 MSG_DESC - Descriptor for message.
179 1217 1
180 1218 1 IMPLICIT INPUTS:
181 1219 1 NONE
182 1220 1
183 1221 1 OUTPUT PARAMETERS:
184 1222 1 NONE
185 1223 1
186 1224 1 IMPLICIT OUTPUTS:
187 1225 1 NONE
188 1226 1
189 1227 1 ROUTINE VALUE:
190 1228 1 FALSE, to signal $PUTMSG not to write the message.
191 1229 1
192 1230 1 SIDE EFFECTS:
193 1231 1 NONE
194 1232 1
195 1233 1 --
196 1234 1
197 1235 2 BEGIN
198 1236 2 MAP
199 1237 2 MSG_DESC: REF BBLOCK; ! Descriptor for message text
200 1238 2 LOCAL
201 1239 2 LENGTH: WORD, ! Length of message, minimized
202 1240 2 OPC_BUFFER: BBLOCK[$BYTEOFFSET(OPC$L_MS_TEXT) + 512],
203 1241 2 ! Buffer for OPCOM message
204 1242 2 OPC_DESC: VECTOR[2], ! Descriptor for message buffer
205 1243 2 STATUS; ! Status return
206 1244 2
207 1245 2
208 1246 2 ! Set up the OPCOM message buffer.
209 1247 2
210 1248 2 OPC_BUFFER[OPC$B_MS_TYPE] = OPC$ RQ RQST;
211 1249 2 OPC_BUFFER[OPC$B_MS_TARGET] = OPC$M_NM_PRINT;
212 1250 2 OPC_BUFFER[OPC$W_MS_STATUS] = 0;
213 1251 2 OPC_BUFFER[OPC$L_MS_RQSTID] = 0;
214 1252 2 LENGTH = .MSG_DESC[DSC$W_LENGTH];
215 1253 2 IF .LENGTH GTRU 512 THEN LENGTH = 512;
216 1254 2 CH$MOVE(.LENGTH, .MSG_DESC[DSC$A_POINTER], OPC_BUFFER[OPC$L_MS_TEXT]);
217 1255 2 OPC_DESC[0] = $BYTEOFFSET(OPC$L_MS_TEXT) + .LENGTH;
218 1256 2 OPC_DESC[1] = OPC_BUFFER;
219 1257 2
220 1258 2
221 1259 2 ! Try to send the message by OPCOM. If this fails, send a broadcast to the
222 1260 2 ! system console.
223 1261 2
224 1262 2 STATUS = $SNDOPR(MSGBUF=OPC_DESC);
```

```

: 225      1263 2 IF NOT .STATUS OR .STATUS EQL OPC$_NOPERATOR
: 226      1264 2 THEN
: 227      1265 2     $BRKTHRU(
: 228      1266 2         MSGBUF=.MSG_DESC,
: 229      1267 2         SENDTO=OPAO_DESC,
: 230      1268 2         SNTYP=BRK$_DEVICE,
: 231      1269 2         TIMEOUT=10);
: 232      1270 2
: 233      1271 2
: 234      1272 2 ! Return FALSE, to signal $PUTMSG not to write the message.
: 235      1273 2 !
: 236      1274 2 FALSE
: 237      1275 1 END;

```

```

.TITLE SYMBIONT Symbiont communication
.IDENT \V04-000\
.PSECT COMMON,NOEXE, OVR,2

```

```

00000 DIAG_STORAGE BASE:
      .BLKB 0
00000 DIAG_TRACE:
      .BLKB 96
00060 DIAG_COUNT:
      .BLKB 96
000C0 DIAG_FLAGS:
      .BLKB 4
000C4 WORK_AREA:
      .BLKB 44
000F0 SNDJBC_COUNT:
      .BLKB 132
00174 GETQUI_COUNT:
      .BLKB 40
0019C SNDACC_COUNT:
      .BLKB 28
001B8 SNDSMB_COUNT:
      .BLKB 72
00200 DIAG_STORAGE END:
      .BLKB 0
00200 FLAGS: .BLKB 4
00204 IMAGE_DUMP STSFLG:
      .BLKB 4
00208 THIS_SYSID:
      .BLKB 6
0020E .BLKB 2
00210 CUR_TIME:
      .BLKB 8
00218 HOURLY_TIME:
      .BLKB 8
00220 HOURLY_PARAMS:
      .BLKB 20
00234 SYMBIONT_COUNT:
      .BLKB 4
00238 QUEUE_REFERENCE_COUNT:
      .BLKB 4
0023C MBX_MESSAGE_COUNT:

```


00240 MBX: .BLKB 4
00244 MBX_END: .BLKB 4
00248 MEMORY_FREE_QUEUES: .BLKB 40
00270 NONAST_WORK_QUEUE: .BLKB 8
00278 BCB_FREE_LIST: .BLKB 4
0027C BCB_ACTIVE_LIST: .BLKB 4
00280 GQL_FREE_LIST: .BLKB 4
00284 GQL_ACTIVE_LIST: .BLKB 4
00288 OPEN_GETQUI_LIST: .BLKB 4
0028C PROCESS_DATA_LIST: .BLKB 4
00290 SYMBIONT_CONTROL: .BLKB 4
00294 SPARE_AREA: .BLKB 12
002A0 REMOTE_REQUEST_LKSB: .BLKB 8
002A8 QUEUE_FILE_LKSB: .BLKB 8
002B0 QUEUE_LOCK_LKSB: .BLKB 8
002B8 RSP: .BLKB 8
002C0 JBC_PRIORITY: .BLKB 4
002C4 JBC_PRIVILEGES: .BLKB 8
002CC JBC_QUOTAS: .BLKB 66
0030E .BLKB 2
00310 JBC_UIC: .BLKB 4
00314 QUEUE_FAB: .BLKB 80
00364 QUEUE_RAB: .BLKB 68
003A8 QUEUE_NAM: .BLKB 96
00408 QUEUE_XAB: .BLKB 88
00460 QUEUE_RSA: .BLKB 255
0055F .BLKB 1
00560 QUEUE_ALQ: .BLKB 4
00564 QUEUE_MBF: .BLKB 1
00565 .BLKB 3
00568 ACCOUNTING_FABS: .BLKB 8
00570 ACCOUNTING_RABS: .BLKB 8

```

00578 ACCOUNT_FAB_A:      .BLKB  8
                          .BLRB  80
005C8 ACCOUNT_RAB_A:      .BLRB  68
                          .BLRB  96
0060C ACCOUNT_NAM_A:      .BLRB  96
                          .BLRB  255
0076B                      .BLKB  1
0076C ACCOUNT_FAB_B:      .BLRB  80
                          .BLRB  68
007BC ACCOUNT_RAB_B:      .BLRB  96
                          .BLRB  255
00800 ACCOUNT_NAM_B:      .BLRB  96
                          .BLRB  255
00860 ACCOUNT_RSA_B:      .BLKB  1
0095F                      .BLKB  80
00960 DIAG_FAB:           .BLKB  68
                          .BLKB  4
009B0 DIAG_RAB:           .BLKB  8
009F4 MBX_CHAN:           .BLKB  8
009F8 MBX_IOSB:           .BLKB 1024
00A00 MBX_BUFFER:         .BLKB  0
00E00 VALUE_STORAGE_BASE: .BLKB  0
00E00 ITEM_PRESENT:       .BLKB  32
00E20 VALUE_GETQUI_BASE:  .BLKB  0
00E20 VALUE_ACCOUNTING_MESSAGE: .BLKB  8
00E26 VALUE_ACCOUNTING_TYPES: .BLKB  4
00E2A VALUE_AFTER_TIME:   .BLRB  8
00E32 VALUE_ALIGNMENT_PAGES: .BLKB  1
00E33 VALUE_BASE_PRIORITY: .BLKB  1
00E34 VALUE_BATCH_INPUT:  .BLRB  6
00E3A VALUE_BATCH_OUTPUT: .BLRB 10
00E44 VALUE_BUFFER_COUNT: .BLKB  1
00E45 VALUE_CHARACTERISTIC_NAME: .BLRB  6
00E4B VALUE_CHARACTERISTIC_NUMBER: .BLKB  1
00E4C VALUE_CHARACTERISTICS: .BLKB 16

```

.....

00E5C VALUE_CHECKPOINT_DATA:
 .BLK8 8
00E62 VALUE_CLI:
 .BLK8 6
00E68 VALUE_CPU_DEFAULT:
 .BLK8 4
00E6C VALUE_CPU_LIMIT:
 .BLK8 4
00E70 VALUE_DESTINATION_QUEUE:
 .BLK8 8
00E78 VALUE_DEVICE_NAME:
 .BLK8 6
00E7E VALUE_ENTRY_NUMBER:
 .BLK8 4
00E82 VALUE_ENTRY_NUMBER_OUTPUT:
 .BLK8 10
00E8C VALUE_EXTEND_QUANTITY:
 .BLK8 2
00E8E VALUE_FILE_COPIES:
 .BLK8 1
00E8F VALUE_FILE_IDENTIFICATION:
 .BLK8 36
00EB3 VALUE_FILE_SETUP_MODULES:
 .BLK8 8
00EB9 VALUE_FILE_SPECIFICATION:
 .BLK8 6
00EBF VALUE_FIRST_PAGE:
 .BLK8 4
00EC3 VALUE_FORM_DESCRIPTION:
 .BLK8 6
00EC9 VALUE_FORM_LENGTH:
 .BLK8 1
00ECA VALUE_FORM_MARGIN_BOTTOM:
 .BLK8 1
00ECB VALUE_FORM_MARGIN_LEFT:
 .BLK8 2
00ECD VALUE_FORM_MARGIN_RIGHT:
 .BLK8 2
00ECF VALUE_FORM_MARGIN_TOP:
 .BLK8 1
00ED0 VALUE_FORM_NAME:
 .BLK8 6
00ED6 VALUE_FORM_NUMBER:
 .BLK8 4
00EDA VALUE_FORM:
 .BLK8 8
00EE2 VALUE_FORM_SETUP_MODULES:
 .BLK8 8
00EE8 VALUE_FORM_STOCK:
 .BLK8 6
00EEE VALUE_FORM_WIDTH:
 .BLK8 2
00EFO VALUE_GENERIC_TARGET:
 .BLK8 996
012D4 VALUE_JOB_COPIES:
 .BLK8 1
012D5 VALUE_JOB_LIMIT:

012D6 VALUE_JOB_NAME: .BLKB 1
012DC VALUE_JOB_RESET_MODULES: .BLKB 6
012E2 VALUE_JOB_SIZE_MAXIMUM: .BLKB 6
012E6 VALUE_JOB_SIZE_MINIMUM: .BLKB 4
012EA VALUE_JOB_STATUS_OUTPUT: .BLKB 4
012F4 VALUE_LAST_PAGE: .BLKB 4
012F8 VALUE_LIBRARY_SPECIFICATION: .BLKB 6
012FE VALUE_LOG_QUEUE: .BLKB 8
01306 VALUE_LOG_SPECIFICATION: .BLKB 6
0130C VALUE_NOTE: .BLKB 6
01312 VALUE_OPERATOR_REQUEST: .BLKB 6
01318 VALUE_OWNER_UIC: .BLKB 4
0131C VALUE_PAGE_SETUP_MODULES: .BLKB 8
01322 VALUE_PARAMETER_1: .BLKB 6
01328 VALUE_PARAMETER_2: .BLKB 6
0132E VALUE_PARAMETER_3: .BLKB 6
01334 VALUE_PARAMETER_4: .BLKB 6
0133A VALUE_PARAMETER_5: .BLKB 6
01340 VALUE_PARAMETER_6: .BLKB 6
01346 VALUE_PARAMETER_7: .BLKB 6
0134C VALUE_PARAMETER_8: .BLKB 6
01352 VALUE_PRIORITY: .BLKB 1
01353 VALUE_PROCESSOR: .BLKB 6
01359 VALUE_PROTECTION: .BLKB 4
0135D VALUE_QUEUE: .BLKB 6
01363 VALUE_QUEUE_FILE_SPECIFICATION: .BLKB 8
01369 VALUE_RELATIVE_PAGE: .BLKB 4
0136D VALUE_RESERVED_INPUT_1: .BLKB 1

```

0136E VALUE_RESERVED_INPUT_2:
      .BLKB 2
01370 VALUE_RESERVED_INPUT_3:
      .BLKB 4
01374 VALUE_RESERVED_INPUT_4:
      .BLKB 6
0137A VALUE_RESERVED_OUTPUT_1:
      .BLKB 10
01384 VALUE_RESERVED_OUTPUT_2:
      .BLKB 10
0138E VALUE_SEARCH_STRING:
      .BLKB 6
01394 VALUE_SCSNODE_NAME:
      .BLKB 6
0139A VALUE_WSDEFAULT:
      .BLKB 2
0139C VALUE_WSEXTENT:
      .BLKB 2
0139E VALUE_WSQUOTA:
      .BLKB 2
013A0 VALUE_STORAGE_END:
      .BLKB 0

```

```

JBC$_CLOSEOUT= 266328
JBC$_NOCMKRNL= 272388
JBC$_NOOPER= 272532
JBC$_NOSYSNAM= 272404
JBC$_OPENIN= 266392
JBC$_OPENOUT= 266400
JBC$_READERR= 266416
JBC$_WRITEERR= 266448
      .EXTRN ALLOCATE_MEMORY
      .EXTRN COMPLETE_JOB, DEALLOCATE_MEMORY
      .EXTRN DEALLOCATE_VARIABLE_DATA
      .EXTRN ENQUEUE_JOB, ENTER_PROCESS_DATA
      .EXTRN FETCH_VARIABLE_DATA
      .EXTRN FETCH_VARIABLE_ITEM
      .EXTRN FETCH_VARIABLE_ITEM_LIST
      .EXTRN FIND_PENDING_JOBS
      .EXTRN LOCK_QUEUE_FILE
      .EXTRN READ_RECORD, RELEASE_RECORD
      .EXTRN REWRITE_RECORD, SCAN_INCOMPLETE_SERVICES
      .EXTRN STORE_VARIABLE_DATA
      .EXTRN UNLOCK_QUEUE_FILE
      .EXTRN UPDATE_GETQUI_DATA
      .EXTRN JOBCTLMBX_DESC, NLAO_DESC
      .EXTRN OPAO_DESC, SYS$SNDOPR
      .EXTRN SYS$BRKTHRU
      .PSECT CODE, NOWRT, 2

```

00FC 0000 OPERATOR REQUEST ACTION:

```

      .WORD Save R2,R3,R4,R5,R6,R7 : 1206
08 SE FDF0 CE 9E 00002 MOVAB -528(SP), SP : 1248
   AE 0203 8F 3C 00007 MOVZWL #515, OPC_BUFFER : 1251
      OC AE D4 0000D CLRL OPC_BUFFER+4 : 1252
   57 04 AC D0 00010 MOVL MSG_DESC, R7

```

		56		67	B0	00014	MOVW	(R7), LENGTH		
	0200	8F		56	B1	00017	CMPW	LENGTH, #512	1253	
				05	1B	0001C	BLEQU	1\$		
10	AE	04	0200	8F	B0	0001E	MOVW	#512, LENGTH	1254	
		6E		56	28	00023	1\$:	MOVW	LENGTH, @4(R7), OPC_BUFFER+8	1255
		6E		56	3C	00029	MOVZWL	LENGTH, OPC_DESC		
		AE	08	08	C0	0002C	ADDL2	#8, OPC_DESC		
				7E	D4	00034	MOVAB	OPC_BUFFER, OPC_DESC+4	1256	
			04	AE	9F	00036	CLRL	-(SP)	1262	
	00000000G	00		02	FB	00039	PUSHAB	OPC_DESC		
		09		50	E9	00040	CALLS	#2, -SYS\$SNDOPR		
	00058061	8F		50	D1	00043	BLBC	STATUS, 2\$	1263	
				1C	12	0004A	CMPL	STATUS, #360545		
				7E	7C	0004C	2\$:	BNEQ	3\$	1269
				0A	DD	0004E	CLRQ	-(SP)		
				7E	7C	00050	PUSHL	#10		
				20	DD	00052	CLRQ	-(SP)		
		7E		01	7D	00054	PUSHL	#32		
			00000000G	EF	9F	00057	MOVQ	#1, -(SP)		
				57	DD	0005D	PUSHAB	OPAO_DESC		
				7E	D4	0005F	PUSHL	R7		
	00000000G	00		0B	FB	00061	CLRL	-(SP)		
				50	D4	00068	3\$:	CALLS	#11, SYS\$BRKTHRU	1275
				04	0006A		CLRL	R0		
							RET			

; Routine Size: 107 bytes, Routine Base: CODE + 0000

```
239 1276 1 ROUTINE OPERATOR_REQUEST(SMQ,SJH): NOVALUE=  
240 1277 1  
241 1278 1 ++  
242 1279 1  
243 1280 1 FUNCTIONAL DESCRIPTION:  
244 1281 1 This routine formats and writes an operator request message to the  
245 1282 1 printer operator.  
246 1283 1  
247 1284 1 INPUT PARAMETERS:  
248 1285 1 SMQ - Pointer to SMQ.  
249 1286 1 SJH - Pointer to SJH.  
250 1287 1  
251 1288 1 IMPLICIT INPUTS:  
252 1289 1 NONE  
253 1290 1  
254 1291 1 OUTPUT PARAMETERS:  
255 1292 1 NONE  
256 1293 1  
257 1294 1 IMPLICIT OUTPUTS:  
258 1295 1 NONE  
259 1296 1  
260 1297 1 ROUTINE VALUE:  
261 1298 1 NONE  
262 1299 1  
263 1300 1 SIDE EFFECTS:  
264 1301 1 Message written to operator.  
265 1302 1  
266 1303 1 --  
267 1304 1  
268 1305 2 BEGIN  
269 1306 2 MAP  
270 1307 2 SMQ: REF BBLOCK, ! Pointer to SMQ  
271 1308 2 SJH: REF BBLOCK; ! Pointer to SJH  
272 1309 2 LOCAL  
273 1310 2 MSGVEC: VECTOR[9], ! $PUTMSG message vector  
274 1311 2 BUFFER: VECTOR[132,BYTE]; ! User's operator request text  
275 1312 2  
276 1313 2  
277 1314 2 ! Fetch the user's operator request message.  
278 1315 2  
279 1316 2 FETCH VARIABLE DATA(  
280 1317 2 SJH$$ OPERATOR_REQUEST, SJH[SJH$T_OPERATOR_REQUEST],  
281 1318 2 XALLOCATION(BUFFER), BUFFER);  
282 1319 2  
283 1320 2  
284 1321 2 ! Format the $PUTMSG buffer.  
285 1322 2  
286 1323 2 MSGVEC[0] = 8;  
287 1324 2 MSGVEC[1] = JBC$_REQUEST;  
288 1325 2 MSGVEC[2] = 6;  
289 1326 2 MSGVEC[3] = SMQ[SMQ$T_NAME];  
290 1327 2 MSGVEC[4] = SJH[SJH$T_NAME];  
291 1328 2 MSGVEC[5] = SJH$$ USERNAME;  
292 1329 2 MSGVEC[6] = SJH[SJH$T_USERNAME];  
293 1330 2 MSGVEC[7] = .BBLOCK[SJH[SJH$T_OPERATOR_REQUEST], FVDF_LENGTH];  
294 1331 2 MSGVEC[8] = BUFFER;  
295 1332 2 $PUTMSG(MSGVEC=MSGVEC, ACTRTN=OPERATOR_REQUEST_ACTION);
```

: 296

1333 1 END;

				.EXTRN SYSS\$PUTMSG		
		0004 00000 OPERATOR_REQUEST:				
		5E	FF58	CE 9E 00002	WORD Save R2	: 1276
				5E DD 00007	MOVAB -168(SP), SP	: 1317
		7E	84	8F 9A 00009	PUSHL SP	
		52	08	AC D0 0000D	MOVZBL #132, -(SP)	
			01AC	C2 9F 00011	MOVL SJH, R2	
				06 DD 00015	PUSHAB 428(R2)	
		00000000G		04 FB 00017	PUSHL #6	
				08 D0 0001E	CALLS #4, FETCH_VARIABLE_DATA	
		DC		8F D0 00022	MOVL #8, MSGVEC	: 1323
		E0	00048450	06 D0 0002A	MOVL #296016, MSGVEC+4	: 1324
		E4		8F C1 0002E	MOVL #6, MSGVEC+8	: 1325
E8	AD	04	000000B0	0C D0 00042	ADDL3 #176, SMQ, MSGVEC+12	: 1326
EC	AD	08	00000108	8F C1 00038	ADDL3 #264, SJH, MSGVEC+16	: 1327
		FO		0C D0 00042	MOVL #12, MSGVEC+20	: 1328
F4	AD	08	00000148	8F C1 00046	ADDL3 #328, SJH, MSGVEC+24	: 1329
		F8	01AC	C2 3C 00050	MOVZWL 428(R2), MSGVEC+28	: 1330
		FC		6E 9E 00056	MOVAB BUFFER, MSGVEC+32	: 1331
				7E 7C 0005A	CLRQ -(SP)	: 1332
			FF35	CF 9F 0005C	PUSHAB OPERATOR_REQUEST_ACTION	
			DC	AD 9F 00060	PUSHAB MSGVEC	
		00000000G	00	04 FB 00063	CALLS #4, SYSS\$PUTMSG	
				04 0006A	RET	: 1333

: Routine Size: 107 bytes, Routine Base: CODE + 006B


```

: 298 1334 1 ROUTINE SEND_SYMBIONT_MESSAGE(SMQ,MSG_DESC): NOVALUE=
: 299 1335 1
: 300 1336 1  +-+
: 301 1337 1
: 302 1338 1  FUNCTIONAL DESCRIPTION:
: 303 1339 1      This routine sends a message to a specified symbiont.
: 304 1340 1
: 305 1341 1  INPUT PARAMETERS:
: 306 1342 1      SMQ          - Pointer to SMQ.
: 307 1343 1      MSG_DESC     - Descriptor for message.
: 308 1344 1
: 309 1345 1  IMPLICIT INPUTS:
: 310 1346 1      NONE
: 311 1347 1
: 312 1348 1  OUTPUT PARAMETERS:
: 313 1349 1      NONE
: 314 1350 1
: 315 1351 1  IMPLICIT OUTPUTS:
: 316 1352 1      NONE
: 317 1353 1
: 318 1354 1  ROUTINE VALUE:
: 319 1355 1      NONE
: 320 1356 1
: 321 1357 1  SIDE EFFECTS:
: 322 1358 1      Message written to mailbox.
: 323 1359 1
: 324 1360 1  --
: 325 1361 1
: 326 1362 2 BEGIN
: 327 1363 2 MAP
: 328 1364 2      SMQ          REF BBLOCK,      ! Pointer to SMQ.
: 329 1365 2      MSG_DESC:   REF BBLOCK;      ! Descriptor for message
: 330 1366 2 LOCAL
: 331 1367 2      STATUS;          ! Status return
: 332 1368 2
: 333 1369 2
: 334 1370 2 ! Write the message without waiting.
: 335 1371 2
: 336 P 1372 2 STATUS = $QIO(
: 337 P 1373 2      FUNC=IOS$ WRITEVBLK OR IOS$ NOW OR IOS$ NORWAIT,
: 338 P 1374 2      CHAN=.BBLOCK[.SMQ[SMQ$L_STREAM_SCT], SCT_W_MAILBOX],
: 339 P 1375 2      P1=.MSG_DESC[DSC$A_POINTER],
: 340 1376 2      P2=.MSG_DESC[DSC$W_LENGTH]);
: 341 1377 2 IF NOT .STATUS THEN SIGNAL(JBC$_WRISMBMBX OR STS$K_ERROR, 0, .STATUS);
: 342 1378 1 END;

```

.EXTRN SYSSQIO

0000 0000 SEND_SYMBIONT_MESSAGE:

				.WORD	Save nothing
		7E	7C	00002	CLRQ -(SP)
		7E	7C	00004	CLRQ -(SP)
50	08	AC	D0	00006	MOVL MSG_DESC, R0
7E		60	3C	0000A	MOVZWL (R0), -(SP)
	04	A0	DD	0000D	PUSHL 4(R0)

```

: 1334
: 1376
:
:
:

```

		7E	7C	00010	CLRQ	-(SP)	:
		7E	D4	00012	CLRL	-(SP)	:
	7E	0470	8F	3C	MOVZWL	#1136, -(SP)	:
	50	04	AC	D0	MOVL	SMQ, R0	:
	50	00FC	C0	D0	MOVL	252(R0), R0	:
	7E	06	A0	3C	MOVZWL	6(R0), -(SP)	:
			7E	D4	CLRL	-(SP)	:
00000000G	00		0C	FB	CALLS	#12, SYS\$QIO	:
	11		50	E8	BLBS	STATUS, 1\$: 1377
			50	DD	PUSHL	STATUS	:
			7E	D4	CLRL	-(SP)	:
		0004847A	8F	DD	PUSHL	#296058	:
00000000G	00		03	FB	CALLS	#3, LIB\$SIGNAL	:
			04	00043	RET		: 1378

; Routine Size: 68 bytes, Routine Base: CODE + 00D6

```

344 1379 1 GLOBAL ROUTINE START_SYMBIONT_TASK(SMQ_N,SMQ,SJH_N,SJH,SQR_N,SQR): NOVALUE=
345 1380 1
346 1381 1 !++
347 1382 1
348 1383 1 FUNCTIONAL DESCRIPTION:
349 1384 1 This routine sends the "start task" message to a symbiont.
350 1385 1
351 1386 1 INPUT PARAMETERS:
352 1387 1 SMQ_N - Record number of SMQ.
353 1388 1 SMQ - Pointer to SMQ.
354 1389 1 SJH_N - Record number of SJH.
355 1390 1 SJH - Pointer to SJH.
356 1391 1 SQR_N - Record number of SQR.
357 1392 1 SQR - Pointer to SQR.
358 1393 1
359 1394 1 IMPLICIT INPUTS:
360 1395 1 NONE
361 1396 1
362 1397 1 OUTPUT PARAMETERS:
363 1398 1 NONE
364 1399 1
365 1400 1 IMPLICIT OUTPUTS:
366 1401 1 NONE
367 1402 1
368 1403 1 ROUTINE VALUE:
369 1404 1 NONE
370 1405 1
371 1406 1 SIDE EFFECTS:
372 1407 1 NONE
373 1408 1
374 1409 1 --
375 1410 1
376 1411 2 BEGIN
377 1412 2 MAP
378 1413 2 SMQ: REF BBLOCK, ! Pointer to SMQ
379 1414 2 SJH: REF BBLOCK, ! Pointer to SJH
380 1415 2 SQR: REF BBLOCK; ! Pointer to SQR
381 1416 2 LOCAL
382 1417 2 FIRST_FILE, ! True if first file in job
383 1418 2 LAST_FILE, ! True if last file in job
384 1419 2 SFM: REF BBLOCK, ! Pointer to SFM
385 1420 2 QSMQ: REF BBLOCK, ! Pointer to job's SMQ
386 1421 2 SMBMSG: BBLOCK[JBC$K_SMBMBXSIZ], ! Message buffer
387 1422 2 SMBITM: REF BBLOCK, ! Cursor for message items
388 1423 2 SMBMSG_DESC: VECTOR[2]; ! Descriptor for message buffer
389 1424 2
390 1425 2
391 1426 2 ! Read the form definition.
392 1427 2
393 1428 2 SFM = READ_RECORD(.SJH[SJH$L_FORM_LINK]);
394 1429 2
395 1430 2
396 1431 2 ! Message header.
397 1432 2
398 1433 2 SMBMSG[SMBMSG$W_REQUEST_CODE] = SMBMSG$K_START_TASK;
399 1434 2 SMBMSG[SMBMSG$B_STRUCTURE_LEVEL] = SMBMSG$K_STRUCTURE_LEVEL;
400 1435 2 SMBMSG[SMBMSG$B_STREAM_INDEX] = .SMQ[SMQ$B_STREAM_INDEX];

```

```

: 401      1436 2 SMBITM = SMBMSG + SMBMSGSS_REQUEST_HEADER;
: 402      1437 2
: 403      1438 2
: 404      1439 2 ! Account name.
: 405      1440 2 !
: 406      1441 2 SMBITM[SMBMSG$W_ITEM_SIZE] = SJH$S_ACCOUNT;
: 407      1442 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_ACCOUNT_NAME;
: 408      1443 2 SMBITM = .SMBITM + SMBMSGSS_ITEM_HEADER;
: 409      1444 2 MOV3(
: 410      1445 2     %REF(SJH$S_ACCOUNT),
: 411      1446 2     SJH[SJH$T_ACCOUNT],
: 412      1447 2     .SMBITM; ..., SMBITM);
: 413      1448 2
: 414      1449 2
: 415      1450 2 ! After time.
: 416      1451 2 !
: 417      1452 2 SMBITM[SMBMSG$W_ITEM_SIZE] = SJH$S_AFTER_TIME;
: 418      1453 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_AFTER_TIME;
: 419      1454 2 SMBITM = .SMBITM + SMBMSGSS_ITEM_HEADER;
: 420      1455 2 COPY TIME(SJH[SJH$Q_AFTER_TIME], .SMBITM);
: 421      1456 2 SMBITM = .SMBITM + SJH$S_AFTER_TIME;
: 422      1457 2
: 423      1458 2
: 424      1459 2 ! Form bottom margin.
: 425      1460 2 !
: 426      1461 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
: 427      1462 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_BOTTOM_MARGIN;
: 428      1463 2 SMBITM = .SMBITM + SMBMSGSS_ITEM_HEADER;
: 429      1464 2 .SMBITM = .SFM[SFM$B_MARGIN_BOTTOM];
: 430      1465 2 SMBITM = .SMBITM + 4;
: 431      1466 2
: 432      1467 2
: 433      1468 2 ! Characteristics.
: 434      1469 2 !
: 435      1470 2 SMBITM[SMBMSG$W_ITEM_SIZE] = SJH$S_CHARACTERISTICS;
: 436      1471 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_CHARACTERISTICS;
: 437      1472 2 SMBITM = .SMBITM + SMBMSGSS_ITEM_HEADER;
: 438      1473 2 MOV3(
: 439      1474 2     %REF(SJH$S_CHARACTERISTICS),
: 440      1475 2     SJH[SJH$T_CHARACTERISTICS],
: 441      1476 2     .SMBITM; ..., SMBITM);
: 442      1477 2
: 443      1478 2
: 444      1479 2 ! Checkpoint data.
: 445      1480 2 !
: 446      1481 2 IF .SJH[SJH$L_CURRENT_FILE_CHKPT] EQL .SQR N
: 447      1482 2 AND .SJH[SJH$B_JOB_COPIES_CHKPT] EQL .SJH[SJH$B_JOB_COPIES_DONE]
: 448      1483 2 AND .SJH[SJH$B_FILE_COPIES_CHKPT] EQL .SJH[SJH$B_FILE_COPIES_DONE]
: 449      1484 2 THEN
: 450      1485 2     SMBITM = FETCH VARIABLE ITEM(
: 451      1486 2     SJH$S_CHECKPOINT, SJH[SJH$T_CHECKPOINT],
: 452      1487 2     SMBMSG$K_CHECKPOINT_DATA,
: 453      1488 2     .SMBITM);
: 454      1489 2
: 455      1490 2
: 456      1491 2 ! Entry number.
: 457      1492 2 !

```

```
458 1493 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
459 1494 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_ENTRY_NUMBER;
460 1495 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
461 1496 2 .SMBITM = .SJH[SYMS$L_ENTRY_NUMBER];
462 1497 2 SMBITM = .SMBITM + 4;
463 1498 2
464 1499 2
465 1500 2 ! File copies.
466 1501 2
467 1502 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
468 1503 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_FILE_COPIES;
469 1504 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
470 1505 2 .SMBITM = .SQR[SQR$B_FILE_COPIES];
471 1506 2 SMBITM = .SMBITM + 4;
472 1507 2
473 1508 2
474 1509 2 ! File copy number.
475 1510 2
476 1511 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
477 1512 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_FILE_COUNT;
478 1513 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
479 1514 2 .SMBITM = .SJH[SJH$B_FILE_COPIES_DONE] + 1;
480 1515 2 SMBITM = .SMBITM + 4;
481 1516 2
482 1517 2
483 1518 2 ! File setup modules.
484 1519 2
485 1520 2 SMBITM = FETCH_VARIABLE_ITEM(
486 1521 2     SQR$S_FILE_SETUP_MODULES, SQR[SQR$T_FILE_SETUP_MODULES],
487 1522 2     SMBMSG$K_FILE_SETUP_MODULES,
488 1523 2     .SMBITM);
489 1524 2
490 1525 2
491 1526 2 ! First page number.
492 1527 2
493 1528 2 IF .SQR[SQR$L_FIRST_PAGE] NEQ 0
494 1529 2 THEN
495 1530 3 BEGIN
496 1531 3     SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
497 1532 3     SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_FIRST_PAGE;
498 1533 3     SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
499 1534 3     .SMBITM = .SQR[SQR$L_FIRST_PAGE];
500 1535 3     SMBITM = .SMBITM + 4;
501 1536 2 END;
502 1537 2
503 1538 2
504 1539 2 ! Form length.
505 1540 2
506 1541 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
507 1542 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_FORM_LENGTH;
508 1543 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
509 1544 2 .SMBITM = .SFM[SFM$B_LENGTH];
510 1545 2 SMBITM = .SMBITM + 4;
511 1546 2
512 1547 2
513 1548 2 ! Form name.
514 1549 2
```

```
515 1550 2 SMBITM[SMBMSG$W_ITEM_SIZE] = CHRCHAR(SFM[SFM$T_NAME]);
516 1551 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K FORM_NAME;
517 1552 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
518 1553 2 MOV3(
519 1554 2   %REF(CHRCHAR(SFM[SFM$T_NAME])),
520 1555 2   SFM[SFM$T_NAME] + 1,
521 1556 2   .SMBITM; ... SMBITM);
522 1557 2
523 1558 2
524 1559 2 ! Form setup modules.
525 1560 2 !
526 1561 2 SMBITM = FETCH_VARIABLE_ITEM(
527 1562 2   SFM$S FORM_SETUP_MODULES, SFM[SFM$T_FORM_SETUP_MODULES],
528 1563 2   SMBMSG$K FORM_SETUP_MODULES,
529 1564 2   .SMBITM);
530 1565 2
531 1566 2
532 1567 2 ! Form width.
533 1568 2 !
534 1569 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
535 1570 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K FORM_WIDTH;
536 1571 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
537 1572 2 .SMBITM = .SFM[SFM$W_WIDTH];
538 1573 2 SMBITM = .SMBITM + 4;
539 1574 2
540 1575 2
541 1576 2 ! File identification or condition vector.
542 1577 2 !
543 1578 2 IF CHRCHAR(SQR[SQR$T_FILE_ID_DVI]) NEQ 0
544 1579 2 THEN
545 1580 3 BEGIN
546 1581 3   SMBITM[SMBMSG$W_ITEM_SIZE] = SQR$S FILE_IDENTIFICATION;
547 1582 3   SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K FILE_IDENTIFICATION;
548 1583 3   SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
549 1584 3   MOV3(
550 1585 3     %REF(SQR$S FILE_IDENTIFICATION),
551 1586 3     SQR[SQR$T_FILE_IDENTIFICATION],
552 1587 3     .SMBITM; ... SMBITM);
553 1588 3   END
554 1589 2 ELSE
555 1590 3 BEGIN
556 1591 3   SMBITM[SMBMSG$W_ITEM_SIZE] = SQR$S CONDITION_VECTOR;
557 1592 3   SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K MESSAGE_VECTOR;
558 1593 3   SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
559 1594 3   MOV3(
560 1595 3     %REF(SQR$S CONDITION_VECTOR),
561 1596 3     SQR[SQR$L_CONDITION_T],
562 1597 3     .SMBITM; ... SMBITM);
563 1598 2   END;
564 1599 2
565 1600 2
566 1601 2 ! File specification.
567 1602 2 !
568 1603 2 SMBITM[SMBMSG$W_ITEM_SIZE] = CHRCHAR(SQR[SQR$T_FILE_SPECIFICATION]);
569 1604 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K FILE_SPECIFICATION;
570 1605 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
571 1606 2 MOV3(
```

```

: 572 1607 2 %REF(CH$RCHAR(SQR[SQR$T_FILE_SPECIFICATION])),
: 573 1608 2 SQR[SQR$T_FILE_SPECIFICATION]+1,
: 574 1609 2 .SMBITM; ..., SMBITM);
: 575 1610 2
: 576 1611 2
: 577 1612 2 ! Job copies.
: 578 1613 2
: 579 1614 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
: 580 1615 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_JOB_COPIES;
: 581 1616 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 582 1617 2 .SMBITM = .SJH[SJH$B_JOB_COPIES];
: 583 1618 2 SMBITM = .SMBITM + 4;
: 584 1619 2
: 585 1620 2
: 586 1621 2 ! Job copy number.
: 587 1622 2
: 588 1623 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
: 589 1624 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_JOB_COUNT;
: 590 1625 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 591 1626 2 .SMBITM = .SJH[SJH$B_JOB_COPIES_DONE] + 1;
: 592 1627 2 SMBITM = .SMBITM + 4;
: 593 1628 2
: 594 1629 2
: 595 1630 2 ! Job name.
: 596 1631 2
: 597 1632 2 SMBITM[SMBMSG$W_ITEM_SIZE] = CH$RCHAR(SJH[SJH$T_NAME]);
: 598 1633 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_JOB_NAME;
: 599 1634 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 600 1635 2 MOV3(
: 601 1636 2 %REF(CH$RCHAR(SJH[SJH$T_NAME])),
: 602 1637 2 SJH[SJH$T_NAME]+1,
: 603 1638 2 .SMBITM; ..., SMBITM);
: 604 1639 2
: 605 1640 2
: 606 1641 2 ! Job reset modules.
: 607 1642 2
: 608 1643 2 SMBITM = FETCH VARIABLE ITEM(
: 609 1644 2 SMQ$S_JOB_RESET_MODULES, SMQ[SMQ$T_JOB_RESET_MODULES],
: 610 1645 2 SMBMSG$K_JOB_RESET_MODULES,
: 611 1646 2 .SMBITM);
: 612 1647 2
: 613 1648 2
: 614 1649 2 ! Last page number.
: 615 1650 2
: 616 1651 2 IF .SQR[SQR$L_LAST_PAGE] NEQ 0
: 617 1652 2 THEN
: 618 1653 2 BEGIN
: 619 1654 3 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
: 620 1655 3 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_LAST_PAGE;
: 621 1656 3 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 622 1657 3 .SMBITM = .SQR[SQR$L_LAST_PAGE];
: 623 1658 3 SMBITM = .SMBITM + 4;
: 624 1659 2 END;
: 625 1660 2
: 626 1661 2
: 627 1662 2 ! Form left margin.
: 628 1663 2

```

```
629 1664 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
630 1665 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_LEFT_MARGIN;
631 1666 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
632 1667 2 .SMBITM = .SFM[SFM$W_MARGIN_LEFT];
633 1668 2 SMBITM = .SMBITM + 4;
634 1669 2
635 1670 2
636 1671 2 ! Note.
637 1672 2
638 1673 2 SMBITM = FETCH_VARIABLE_ITEM(
639 1674 2   SJH$S_NOTE, SJH[SJH$T_NOTE],
640 1675 2   SMBMSG$K_NOTE,
641 1676 2   .SMBITM);
642 1677 2
643 1678 2
644 1679 2 ! Page setup modules.
645 1680 2
646 1681 2 SMBITM = FETCH_VARIABLE_ITEM(
647 1682 2   SFM$S_PAGE_SETUP_MODULES, SFM[SFM$T_PAGE_SETUP_MODULES],
648 1683 2   SMBMSG$K_PAGE_SETUP_MODULES,
649 1684 2   .SMBITM);
650 1685 2
651 1686 2
652 1687 2 ! Parameters.
653 1688 2
654 1689 2 SMBITM = FETCH_VARIABLE_ITEM_LIST(
655 1690 2   SJH$S_PARAMETERS, SJH[SJH$T_PARAMETERS],
656 1691 2   SMBMSG$K_PARAMETER_1,
657 1692 2   .SMBITM);
658 1693 2
659 1694 2
660 1695 2 ! Print control flags.
661 1696 2
662 1697 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
663 1698 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_PRINT_CONTROL;
664 1699 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
665 1700 2 .SMBITM = 0;
666 1701 2 IF .SQR[SQR$V_DOUBLE_SPACE] THEN SMBITM[SMBMSG$V_DOUBLE_SPACE] = TRUE;
667 1702 2 IF .SQR[SQR$V_PAGE_HEADER] THEN SMBITM[SMBMSG$V_PAGE_HEADER] = TRUE;
668 1703 2 IF .SQR[SQR$V_PASSALL] THEN SMBITM[SMBMSG$V_PASSALL] = TRUE;
669 1704 2 IF .SFM[SFM$V_SHEET_FEED] THEN SMBITM[SMBMSG$V_SHEET_FEED] = TRUE;
670 1705 2 IF .SFM[SFM$V_TRUNCATE] THEN SMBITM[SMBMSG$V_TRUNCATE] = TRUE;
671 1706 2 IF .SFM[SFM$V_WRAP] THEN SMBITM[SMBMSG$V_WRAP] = TRUE;
672 1707 2
673 1708 2
674 1709 2 ! Compute paginate bit.
675 1710 2
676 1711 2 IF .SQR[SQR$V_PAGINATE_EXPLICIT]
677 1712 2 THEN
678 1713 3   BEGIN
679 1714 3   IF .SQR[SQR$V_PAGINATE]
680 1715 3   THEN
681 1716 3     SMBITM[SMBMSG$V_PAGINATE] = TRUE;
682 1717 3   END
683 1718 3
684 1719 2 ELSE IF .SJH[SJH$V_PAGINATE_EXPLICIT]
685 1720 2 THEN
```



```

: 686 1721 3 BEGIN
: 687 1722 3 IF .SJH[SJH$V_PAGINATE]
: 688 1723 3 THEN
: 689 1724 3 SMBITM[SMBMSG$V_PAGINATE] = TRUE;
: 690 1725 3 END
: 691 1726 3
: 692 1727 2 ELSE
: 693 1728 3 BEGIN
: 694 1729 3 IF .SMQ[SMQ$V_PAGINATE]
: 695 1730 3 THEN
: 696 1731 3 SMBITM[SMBMSG$V_PAGINATE] = TRUE;
: 697 1732 2 END;
: 698 1733 2
: 699 1734 2 SMBITM = .SMBITM + 4;
: 700 1735 2
: 701 1736 2
: 702 1737 2 ! Separation control flags.
: 703 1738 2 !
: 704 1739 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
: 705 1740 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_SEPARATION_CONTROL;
: 706 1741 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 707 1742 2 .SMBITM = 0;
: 708 1743 2 IF .BBLOCK[SMQ[SMQ$T_JOB_RESET_MODULES], FVDF_LENGTH] NEQ 0
: 709 1744 2 THEN SMBITM[SMBMSG$V_JOB_RESET_ABORT] = TRUE;
: 710 1745 2
: 711 1746 2
: 712 1747 2 ! Special actions for the first file in the job.
: 713 1748 2 !
: 714 1749 2 FIRST_FILE = FALSE;
: 715 1750 2 IF
: 716 1751 3 (.SJH[SJH$B_JOB_COPIES_DONE] EQL 0
: 717 1752 3 AND .SJH[SJH$B_FILE_COPIES_DONE] EQL 0
: 718 1753 3 AND .SJH[SJH$L_FILE_LIST] EQL .SQR_N)
: 719 1754 2 OR
: 720 1755 2 .SJH[SJH$V_RESTARTING]
: 721 1756 2 THEN
: 722 1757 3 BEGIN
: 723 1758 3 SJH[SJH$V_RESTARTING] = FALSE;
: 724 1759 3 IF .SMQ[SMQ$V_JOB_FLAG] THEN SMBITM[SMBMSG$V_JOB_FLAG] = TRUE;
: 725 1760 3 IF .SMQ[SMQ$V_JOB_BURST] THEN SMBITM[SMBMSG$V_JOB_BURST] = TRUE;
: 726 1761 3 FIRST_FILE = TRUE;
: 727 1762 2 END;
: 728 1763 2
: 729 1764 2
: 730 1765 2 ! Compute file burst bit.
: 731 1766 2 !
: 732 1767 2 IF .SQR[SQR$V_FILE_BURST_EXPLICIT]
: 733 1768 2 THEN
: 734 1769 3 BEGIN
: 735 1770 3 IF .SQR[SQR$V_FILE_BURST]
: 736 1771 3 THEN
: 737 1772 3 SMBITM[SMBMSG$V_FILE_BURST] = TRUE;
: 738 1773 3 END
: 739 1774 3
: 740 1775 2 ELSE IF .SJH[SJH$V_FILE_BURST_EXPLICIT]
: 741 1776 2 THEN
: 742 1777 3 BEGIN
```

```
.. 743 1778 3 IF .SJH[SJH$V_FILE_BURST]
.. 744 1779 4 OR (.SJH[SJH$V_FILE_BURST_ONE] AND .FIRST_FILE)
.. 745 1780 3 THEN
.. 746 1781 3 SMBITM[SMBMSG$V_FILE_BURST] = TRUE;
.. 747 1782 3 END
.. 748 1783 3
.. 749 1784 2 ELSE
.. 750 1785 3 BEGIN
.. 751 1786 3 IF .SMQ[SMQ$V_FILE_BURST]
.. 752 1787 4 OR (.SMQ[SMQ$V_FILE_BURST_ONE] AND .FIRST_FILE)
.. 753 1788 3 THEN
.. 754 1789 3 SMBITM[SMBMSG$V_FILE_BURST] = TRUE;
.. 755 1790 2 END;
.. 756 1791 2
.. 757 1792 2
.. 758 1793 2 ! Compute file flag bit.
.. 759 1794 2
.. 760 1795 2 IF .SQR[SQR$V_FILE_FLAG_EXPLICIT]
.. 761 1796 2 THEN
.. 762 1797 3 BEGIN
.. 763 1798 3 IF .SQR[SQR$V_FILE_FLAG]
.. 764 1799 3 THEN
.. 765 1800 3 SMBITM[SMBMSG$V_FILE_FLAG] = TRUE;
.. 766 1801 3 END
.. 767 1802 3
.. 768 1803 2 ELSE IF .SJH[SJH$V_FILE_FLAG_EXPLICIT]
.. 769 1804 2 THEN
.. 770 1805 3 BEGIN
.. 771 1806 3 IF .SJH[SJH$V_FILE_FLAG]
.. 772 1807 4 OR (.SJH[SJH$V_FILE_FLAG_ONE] AND .FIRST_FILE)
.. 773 1808 3 THEN
.. 774 1809 3 SMBITM[SMBMSG$V_FILE_FLAG] = TRUE;
.. 775 1810 3 END
.. 776 1811 3
.. 777 1812 2 ELSE
.. 778 1813 3 BEGIN
.. 779 1814 3 IF .SMQ[SMQ$V_FILE_FLAG]
.. 780 1815 4 OR (.SMQ[SMQ$V_FILE_FLAG_ONE] AND .FIRST_FILE)
.. 781 1816 3 THEN
.. 782 1817 3 SMBITM[SMBMSG$V_FILE_FLAG] = TRUE;
.. 783 1818 2 END;
.. 784 1819 2
.. 785 1820 2
.. 786 1821 2 ! Special actions for last file in job.
.. 787 1822 2
.. 788 1823 2 LAST FILE = FALSE;
.. 789 1824 2 IF .SJH[SJH$B_JOB_COPIES_DONE] + 1 GEQU .SJH[SJH$B_JOB_COPIES]
.. 790 1825 2 AND .SJH[SJH$B_FILE_COPIES_DONE] + 1 GEQU .SQR[SQR$B_FILE_COPIES]
.. 791 1826 2 AND .SQR[SYMSL_LINK] EQL 0
.. 792 1827 2 THEN
.. 793 1828 3 BEGIN
.. 794 1829 3 IF .SMQ[SMQ$V_JOB_TRAILER] THEN SMBITM[SMBMSG$V_JOB_TRAILER] = TRUE;
.. 795 1830 3 IF .BBLOCK[SMQ[SMQ$T_JOB_RESET_MODULES], FVDF_LENGTH] NEQ 0
.. 796 1831 3 THEN SMBITM[SMBMSG$V_JOB_RESET] = TRUE;
.. 797 1832 3 LAST_FILE = TRUE;
.. 798 1833 2 END;
.. 799 1834 2
```

```
800 1835 2
801 1836 2 ! Compute file trailer bits.
802 1837 2
803 1838 2 IF .SQR[SQR$V_FILE_TRAILER_EXPLICIT]
804 1839 2 THEN
805 1840 2 BEGIN
806 1841 3 IF .SQR[SQR$V_FILE_TRAILER]
807 1842 3 THEN
808 1843 4 BEGIN
809 1844 4 SMBITM[SMBMSG$V_FILE_TRAILER] = TRUE;
810 1845 4 SMBITM[SMBMSG$V_FILE_TRAILER_ABORT] = TRUE;
811 1846 3 END;
812 1847 3 END
813 1848 3
814 1849 2 ELSE IF .SJH[SJH$V_FILE_TRAILER_EXPLICIT]
815 1850 2 THEN
816 1851 3 BEGIN
817 1852 3 IF .SJH[SJH$V_FILE_TRAILER]
818 1853 3 THEN
819 1854 4 BEGIN
820 1855 4 SMBITM[SMBMSG$V_FILE_TRAILER] = TRUE;
821 1856 4 SMBITM[SMBMSG$V_FILE_TRAILER_ABORT] = TRUE;
822 1857 4 END
823 1858 4
824 1859 3 ELSE IF .SJH[SJH$V_FILE_TRAILER_ONE]
825 1860 3 THEN
826 1861 4 BEGIN
827 1862 4 IF .LAST FILE THEN SMBITM[SMBMSG$V_FILE_TRAILER] = TRUE;
828 1863 4 SMBITM[SMBMSG$V_FILE_TRAILER_ABORT] = TRUE;
829 1864 3 END;
830 1865 3 END
831 1866 3
832 1867 2 ELSE
833 1868 3 BEGIN
834 1869 3 IF .SMQ[SMQ$V_FILE_TRAILER]
835 1870 3 THEN
836 1871 4 BEGIN
837 1872 4 SMBITM[SMBMSG$V_FILE_TRAILER] = TRUE;
838 1873 4 SMBITM[SMBMSG$V_FILE_TRAILER_ABORT] = TRUE;
839 1874 4 END
840 1875 4
841 1876 3 ELSE IF .SMQ[SMQ$V_FILE_TRAILER_ONE]
842 1877 3 THEN
843 1878 4 BEGIN
844 1879 4 IF .LAST FILE THEN SMBITM[SMBMSG$V_FILE_TRAILER] = TRUE;
845 1880 4 SMBITM[SMBMSG$V_FILE_TRAILER_ABORT] = TRUE;
846 1881 3 END;
847 1882 2 END;
848 1883 2 SMBITM = .SMBITM + 4;
849 1884 2
850 1885 2
851 1886 2 ! Request control flags.
852 1887 2
853 1888 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
854 1889 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_REQUEST_CONTROL;
855 1890 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
856 1891 2 .SMBITM = 0;
```

```
857 1892 2 IF .SJH[SJH$V RESTARTING] THEN SMBITM[SMBMSG$V RESTARTING] = TRUE;
858 1893 2 IF .BBLOCK[SJH[SJH$T_OPERATOR_REQUEST], FVDF_LENGTH] NEQ 0
859 1894 2 AND .FIRST_FILE
860 1895 2 THEN
861 1896 2 BEGIN
862 1897 2     SMQ[SMQ$V OPERATOR REQUEST] = TRUE;
863 1898 2     SMBITM[SMBMSG$V_PAUSE_COMPLETE] = FALSE;      ! Temporarily cleared (V03-015)
864 1899 2 END;
865 1900 2 SMBITM = .SMBITM + 4;
866 1901 2
867 1902 2
868 1903 2 ! Job priority.
869 1904 2 !
870 1905 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
871 1906 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_PRIORITY;
872 1907 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
873 1908 2 .SMBITM = .SJH[SJH$B_PRIORITY];
874 1909 2 SMBITM = .SMBITM + 4;
875 1910 2
876 1911 2
877 1912 2 ! Queue name.
878 1913 2 !
879 1914 2 QSMQ = READ_RECORD(.SJH[SJH$L_QUEUE_LINK]);
880 1915 2 SMBITM[SMBMSG$W_ITEM_SIZE] = CH$RCHAR(QSMQ[SMQ$T_NAME]);
881 1916 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_QUEUE;
882 1917 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
883 1918 2 MOV3(
884 1919 2     %REF(CH$RCHAR(QSMQ[SMQ$T_NAME])),
885 1920 2     QSMQ[SMQ$T_NAME]+1,
886 1921 2     .SMBITM; .SMBITM);
887 1922 2 RELEASE_RECORD(.SJH[SJH$L_QUEUE_LINK]);
888 1923 2
889 1924 2
890 1925 2 ! Form right margin.
891 1926 2 !
892 1927 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
893 1928 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_RIGHT_MARGIN;
894 1929 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
895 1930 2 .SMBITM = .SFM[SFM$W_MARGIN_RIGHT];
896 1931 2 SMBITM = .SMBITM + 4;
897 1932 2
898 1933 2
899 1934 2 ! Time queued.
900 1935 2 !
901 1936 2 SMBITM[SMBMSG$W_ITEM_SIZE] = SJH$S_TIME;
902 1937 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_TIME_QUEUED;
903 1938 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
904 1939 2 COPY TIME(SJH[SJH$Q_TIME], .SMBITM);
905 1940 2 SMBITM = .SMBITM + SJH$S_TIME;
906 1941 2
907 1942 2
908 1943 2 ! Form top margin.
909 1944 2 !
910 1945 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
911 1946 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_TOP_MARGIN;
912 1947 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
913 1948 2 .SMBITM = .SFM[SFM$B_MARGIN_TOP];
```

```

: 914      1949 2  SMBITM = .SMBITM + 4;
: 915      1950 2
: 916      1951 2
: 917      1952 2  ! UIC.
: 918      1953 2
: 919      1954 2  SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
: 920      1955 2  SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K UIC;
: 921      1956 2  SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 922      1957 2  .SMBITM = .SJH[SJH$UIC];
: 923      1958 2  SMBITM = .SMBITM + 4;
: 924      1959 2
: 925      1960 2
: 926      1961 2  ! User name.
: 927      1962 2
: 928      1963 2  SMBITM[SMBMSG$W_ITEM_SIZE] = SJH$S USERNAME;
: 929      1964 2  SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K USER_NAME;
: 930      1965 2  SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 931      1966 2  MOV C3(
: 932      1967 2  XREF(SJH$S USERNAME),
: 933      1968 2  SJH[SJH$T_USERNAME],
: 934      1969 2  .SMBITM; ..., SMBITM);
: 935      1970 2
: 936      1971 2
: 937      1972 2  ! Trailing zero item.
: 938      1973 2
: 939      1974 2  SMBITM[SMBMSG$W_ITEM_SIZE] = 0;
: 940      1975 2  SMBITM[SMBMSG$W_ITEM_CODE] = 0;
: 941      1976 2  SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 942      1977 2
: 943      1978 2
: 944      1979 2  ! Send the message to the symbiont.
: 945      1980 2
: 946      1981 2  SMBMSG_DESC[1] = SMBMSG;
: 947      1982 2  SMBMSG_DESC[0] = .SMBITM - .SMBMSG_DESC[1];
: 948      1983 2  SEND_SYMBIONT_MESSAGE(.SMQ, SMBMSG_DESC);
: 949      1984 2
: 950      1985 2
: 951      1986 2  ! Update SMQ.
: 952      1987 2
: 953      1988 2  SMQ[SMQ$L_FORM_LINK] = .SJH[SJH$L_FORM_LINK];
: 954      1989 2
: 955      1990 2
: 956      1991 2  ! Update SJH.
: 957      1992 2
: 958      1993 2  SJH[SJH$L_CURRENT_FILE_CHKPT] = .SQR_N;
: 959      1994 2  SJH[SJH$B_JOB_COPIES_CHKPT] = .SJH[SJH$B_JOB_COPIES_DONE];
: 960      1995 2  SJH[SJH$B_FILE_COPIES_CHKPT] = .SJH[SJH$B_FILE_COPIES_DONE];
: 961      1996 2  SJH[SJH$L_CURRENT_FILE_LINK] = .SQR_N;
: 962      1997 2  DEALLOCATE VARIABLE DATA(
: 963      1998 2  SJH$S CHECKPOINT,
: 964      1999 2  SJH[SJH$T_CHECKPOINT]);
: 965      2000 2  SJH[SJH$V_EXECUTING] = TRUE;
: 966      2001 2  SJH[SJH$V_FILE_STARTING] = TRUE;
: 967      2002 2  RELEASE_RECORD(.SJH[SJH$L_FORM_LINK]);
: 968      2003 1  END;
```

				OFFC 00000	.ENTRY	START SYMBIONT TASK, Save R2,R3,R4,R5,R6,-	
				CE 9E 00002	MOVAB	R7,R8,R9,R10,RT1	1379
		SE FBF8		AC D0 00007	MOVL	-1032(SP), SP	
		56 10		C6 9F 0000B	PUSHAB	SJH, R6	1428
		00FC		BE DD 0000F	PUSHL	252(R6)	
		00		01 FB 00012	CALLS	@0(SP)	
00000000G		EF		50 D0 00019	MOVW	#1, READ_RECORD	
		59		05 B0 0001C	MOVW	R0, SFM	
	OC	AE		01 90 00020	MOVW	#5, SMBMSG	1433
	OE	AE		AC D0 00024	MOVW	#1, SMBMSG+2	1434
		58 08		C8 90 00028	MOVW	SMB, R8	1435
	OF	AE	0117	AE 9E 0002E	MOVW	279(R8), SMBMSG+3	
		53 10		8F D0 00032	MOVAB	SMBMSG+4, SMBITM	1436
63		83 00020008		08 28 00039	MOVL	#131080, (SMBITM)+	1441
	14	A6		8F D0 0003E	MOVW	#8, 20(R6), (SMBITM)	1447
		83 00030008		C6 7D 00045	MOVL	#196616, (SMBITM)+	1452
		83 0098		8F D0 0004A	MOVW	152(R6), (SMBITM)+	1455
		83 00050004		C9 9A 00051	MOVL	#327684, (SMBITM)+	1461
		83 015B		8F D0 00056	MOVZBL	347(SFM), (SMBITM)+	1464
63		83 00060010		10 28 0005D	MOVL	#393232, (SMBITM)+	1470
	00A0	C6		C6 D1 00063	MOVW	#16, 160(R6), (SMBITM)	1476
	14	AC	00EC	26 12 00069	CMPB	236(R6), SQR_N	1481
				C6 91 0006B	BNEQ	1\$	
	017C	C6	017B	1D 12 00072	CMPB	379(R6), 380(R6)	1482
				C6 91 00074	BNEQ	1\$	
	0179	C6	0178	14 12 0007B	CMPB	376(R6), 377(R6)	1483
				53 DD 0007D	BNEQ	1\$	
				07 DD 0007F	PUSHL	SMBITM	1488
				C6 9F 00081	PUSHL	#7	1486
			0180	20 DD 00085	PUSHAB	384(R6)	
				04 FB 00087	PUSHL	#32	
00000000G		EF		50 D0 0008E	CALLS	#4, FETCH_VARIABLE_ITEM	
		53		8F D0 00091	MOVL	R0, SMBITM	
		83 000B0004		A6 D0 00098	MOVL	#720900, (SMBITM)+	1493
		83 08		8F D0 0009C	MOVL	8(R6), (SMBITM)+	1496
		83 000D0004		AC D0 000A3	MOVL	#851972, (SMBITM)+	1502
		57 18		A7 9A 000A7	MOVL	SQR, R7	1505
		83 44		8F D0 000AB	MOVZBL	68(R7), (SMBITM)+	
		83 000E0004		C6 9E 000B2	MOVL	#917508, (SMBITM)+	1511
		5B 0179		6B 9A 000B7	MOVAB	377(R6), R11	1514
		63		83 D6 000BA	MOVZBL	(R11), (SMBITM)	
				53 DD 000BC	INCL	(SMBITM)+	
				0F DD 000BE	PUSHL	SMBITM	1523
				A7 9F 000C0	PUSHL	#15	1521
			45	06 DD 000C3	PUSHAB	69(R7)	
				04 FB 000C5	PUSHL	#6	
00000000G		EF		50 D0 000CC	CALLS	#4, FETCH_VARIABLE_ITEM	
		53		A7 D5 000CF	MOVL	R0, SMBITM	
			3C	0B 13 000D2	TSTL	60(R7)	1528
		83 00100004		8F D0 000D4	BEQL	2\$	
		83 3C		A7 D0 000DB	MOVL	#1048580, (SMBITM)+	1531
		83 00110004		8F D0 000DF	MOVL	60(R7), (SMBITM)+	1534
		83 015A		C9 9A 000E6	MOVL	#1114116, (SMBITM)+	1541
					MOVZBL	346(SFM), (SMBITM)+	1544

		83	0110	C9	9B	000EB	MOVZBW	272(SFM), (SMBITM)+	1550
		83		12	B0	000F0	MOVW	#18, (SMBITM)+	1551
63	0111	50	0110	C9	9A	000F3	MOVZBL	272(SFM), R0	1554
		C9		50	28	000F8	MOV3	R0, 273(SFM), (SMBITM)	1556
				53	DD	000FE	PUSHL	SMBITM	1564
				13	DD	00100	PUSHL	#19	1562
			015D	C9	9F	00102	PUSHAB	349(SFM)	
	00000000G	EF		06	DD	00106	PUSHL	#6	
		53		04	FB	00108	CALLS	#4, FETCH_VARIABLE_ITEM	
		83	00140004	50	D0	0010F	MOVL	R0, SMBITM	
		83	0158	8F	D0	00112	MOVL	#1310724, (SMBITM)+	1569
		83	1C	C9	3C	00119	MOVZWL	344(SFM), (SMBITM)+	1572
				A7	95	0011E	TSTB	28(R7)	1578
				0E	13	00121	BEQL	3\$	
63	1C	83	0015001C	8F	D0	00123	MOVL	#1376284, (SMBITM)+	1581
		A7		1C	28	0012A	MOV3	#28, 28(R7), (SMBITM)	1587
				0C	11	0012F	BRB	4\$	1578
63	10	83	001F000C	8F	D0	00131	MOVL	#2031628, (SMBITM)+	1591
		A7		0C	28	00138	MOV3	#12, 16(R7), (SMBITM)	1597
		83	4B	A7	9B	0013D	MOVZBW	75(R7), (SMBITM)+	1603
		83		16	B0	00141	MOVW	#22, (SMBITM)+	1604
		50	4B	A7	9A	00144	MOVZBL	75(R7), R0	1607
63	4C	A7		50	28	00148	MOV3	R0, 76(R7), (SMBITM)	1609
		83	00170004	8F	D0	0014D	MOVL	#1507332, (SMBITM)+	1614
		83	017A	C6	9A	00154	MOVZBL	378(R6), (SMBITM)+	1617
		83	00180004	8F	D0	00159	MOVL	#1572868, (SMBITM)+	1623
		5A	017C	C6	9E	00160	MOVAB	380(R6), R10	1626
		63		6A	9A	00165	MOVZBL	(R10), (SMBITM)	
				83	D6	00168	INCL	(SMBITM)+	
		83	0108	C6	9B	0016A	MOVZBW	264(R6), (SMBITM)+	1632
		83		19	B0	0016F	MOVW	#25, (SMBITM)+	1633
63	0109	50	0108	C6	9A	00172	MOVZBL	264(R6), R0	1636
		C6		50	28	00177	MOV3	R0, 265(R6), (SMBITM)	1638
				53	DD	0017D	PUSHL	SMBITM	1646
				1A	DD	0017F	PUSHL	#26	1644
			0118	C8	9F	00181	PUSHAB	280(R8)	
	00000000G	EF		06	DD	00185	PUSHL	#6	
		53		04	FB	00187	CALLS	#4, FETCH_VARIABLE_ITEM	
			40	50	D0	0018E	MOVL	R0, SMBITM	
				A7	D5	00191	TSTL	64(R7)	1651
				0B	13	00194	BEQL	5\$	
		83	001B0004	8F	D0	00196	MOVL	#1769476, (SMBITM)+	1654
		83	40	A7	D0	0019D	MOVL	64(R7), (SMBITM)+	1657
		83	001C0004	8F	D0	001A1	MOVL	#1835012, (SMBITM)+	1664
		83	0154	C9	3C	001A8	MOVZWL	340(SFM), (SMBITM)+	1667
				53	DD	001AD	PUSHL	SMBITM	1676
				20	DD	001AF	PUSHL	#32	1674
			01A6	C6	9F	001B1	PUSHAB	422(R6)	
	00000000G	EF		06	DD	001B5	PUSHL	#6	
		53		04	FB	001B7	CALLS	#4, FETCH_VARIABLE_ITEM	
				50	D0	001BE	MOVL	R0, SMBITM	
				53	DD	001C1	PUSHL	SMBITM	1684
				21	DD	001C3	PUSHL	#33	1682
			0163	C9	9F	001C5	PUSHAB	355(SFM)	
	00000000G	EF		06	DD	001C9	PUSHL	#6	
		53		04	FB	001CB	CALLS	#4, FETCH_VARIABLE_ITEM	
				50	D0	001D2	MOVL	R0, SMBITM	

			53	DD	001D5		PUSHL	SMBITM	1692
			22	DD	001D7		PUSHL	#34	1690
		01B2	C6	9F	001D9		PUSHAB	434(R6)	
			20	DD	001DD		PUSHL	#32	
	00000000G	EF	04	FB	001DF		CALLS	#4, FETCH_VARIABLE_ITEM_LIST	
		53	50	D0	001E6		MOVL	R0, SMBITM	
		83	8F	D0	001E9		MOVL	#2752516, (SMBITM)+	1697
			63	D4	001F0		CLRL	(SMBITM)	1700
03			A7	9E	001F2		MOVAB	12(R7), R0	1701
			02	E1	0C1F6		BBC	#2, (R0), 6\$	
03			01	88	001FA		BISB2	#1, (SMBITM)	
			09	E1	001FD	6\$:	BBC	#9, (R0), 7\$	1702
03			02	88	00201		BISB2	#2, (SMBITM)	
			0C	E1	00204	7\$:	BBC	#12, (R0), 8\$	1703
			08	88	00208		BISB2	#8, (SMBITM)	
			A9	E9	0020B	8\$:	BLBC	12(SFM), 9\$	1704
			20	88	0020F		BISB2	#32, (SMBITM)	
04	OC		01	E1	00212	9\$:	BBC	#1, 12(SFM), 10\$	1705
			8F	88	00217		BISB2	#64, (SMBITM)	
04	OC		02	E1	0021B	10\$:	BBC	#2, 12(SFM), 11\$	1706
			8F	88	00220		BISB2	#128, (SMBITM)	
06			0B	E1	00224	11\$:	BBC	#11, (R0), 12\$	1711
15			0A	E1	00228		BBC	#10, (R0), 15\$	1714
			10	11	0022C		BRB	14\$	1716
			A6	E9	0022E	12\$:	BLBC	14(R6), 13\$	1719
			A6	95	00232		TSTB	13(R6)	1722
			0A	18	00235		BGEQ	15\$	
			05	11	00237		BRB	14\$	1724
03	OE		01	E1	00239	13\$:	BBC	#1, 14(R8), 15\$	1729
			04	88	0023E	14\$:	BISB2	#4, (SMBITM)	1731
			04	C0	00241	15\$:	ADDL2	#4, SMBITM	1734
			8F	D0	00244		MOVL	#3342340, (SMBITM)+	1739
			63	D4	0024B		CLRL	(SMBITM)	1742
			55	D4	0024D		CLRL	R5	1743
			C8	B5	0024F		TSTW	280(R8)	
			06	13	00253		BEQL	16\$	
			55	D6	00255		INCL	R5	
			8F	88	00257		BISB2	#128, (SMBITM)	1744
			51	D4	0025B	16\$:	CLRL	FIRST_FILE	1749
			6A	95	0025D		TSTB	(R10)	1751
			0C	12	0025F		BNEQ	17\$	
			6B	95	00261		TSTB	(R11)	1752
			08	12	00263		BNEQ	17\$	
			C6	D1	00265		CMPL	244(R6), SQR_N	1753
			05	13	0026B		BEQL	18\$	
17	11	A6	02	E1	0026D	17\$:	BBC	#2, 17(R6), 21\$	1755
	11	A6	04	8A	00272	18\$:	BICB2	#4, 17(R6)	1758
03	OD	A8	05	E1	00276		BBC	#5, 13(R8), 19\$	1759
		63	10	88	0027B		BISB2	#16, (SMBITM)	
03	OD	A8	04	E1	0027E	19\$:	BBC	#1, 13(R8), 20\$	1760
		63	20	88	00283		BISB2	#2, (SMBITM)	
		51	01	D0	00286	20\$:	MOVL	#1, FIRST_FILE	1761
06		60	04	E1	00289	21\$:	BBC	#4, (R0), 22\$	1767
23		60	03	E1	0028D		BBC	#3, (R0), 26\$	1770
			1E	11	00291		BRB	25\$	1772
OC	OC	A6	02	E1	00293	22\$:	BBC	#2, 12(R6), 23\$	1775
14	OC	A6	01	E0	00298		BBS	#1, 12(R6), 25\$	1778

			12	OC	A6		03	E1	0029D		BBC	#3, 12(R6), 26\$	1779
							0A	11	002A2		BRB	24\$	
			08	OC	A8		04	E0	0C2A4	23\$:	BBS	#4, 12(R8), 25\$	1786
			06	OC	A8		05	E1	002A9		BBC	#5, 12(R8), 26\$	1787
							5'	E9	002AE	24\$:	BLBC	FIRST FILE, 26\$	
							01	88	002B1	25\$:	BISB2	#1, (SMBITM)	1789
			06		60		06	E1	002B4	26\$:	BBC	#6, (R0), 27\$	1795
			23		60		05	E1	002B8		BBC	#5, (R0), 31\$	1798
							1E	11	U02BC		BRB	30\$	1800
			OC	OC	A6		05	E1	002BE	27\$:	BBC	#5, 12(R6), 28\$	1803
			14	OC	A6		04	E0	002C3		BBS	#4, 12(R6), 30\$	1806
			12	OC	A6		06	E1	002C8		BBC	#6, 12(R6), 31\$	1807
							0A	11	002CD		BRB	29\$	
			08	OC	A8		06	E0	002CF	28\$:	BBS	#6, 12(R8), 30\$	1814
						OC	A8	95	002D4		TSTB	12(R8)	1815
							06	18	002D7		BGEQ	31\$	
							03	E9	002D9	29\$:	BLBC	FIRST FILE, 31\$	
			63				02	88	002DC	30\$:	BISB2	#2, (SMBITM)	1817
							52	D4	002DF	31\$:	CLRL	LAST FILE	1823
							54	6A	9A 002E1		MOVZBL	(R10), R4	1824
							54	D6	002E4		INCL	R4	
54	017A	C6					00	ED	002E6		CMPZV	#0, #8, 378(R6), R4	
							24	1A	002ED		BGTRU	34\$	
							54	6B	9A 002EF		MOVZBL	(R11), R4	1825
							54	D6	002F2		INCL	R4	
54	44	A7					00	ED	002F4		CMPZV	#0, #8, 68(R7), R4	
							17	1A	002FA		BGTRU	34\$	
							67	D5	002FC		TSTL	(R7)	1826
							13	12	002FE		BNEQ	34\$	
						OD	A8	95	00300		TSTB	13(R8)	1829
							04	18	00303		BGEQ	32\$	
				01	A3		01	88	00305		BISB2	#1, 1(SMBITM)	
							55	E9	00309	32\$:	BLBC	R5, 33\$	1830
							63	8F	88 0030C		BISB2	#6, (SMBITM)	1831
							52	01	D0 00310	33\$:	MOVL	#1, LAST FILE	1832
							06	A0	E9 00313	34\$:	BLBC	1(R0), 35\$	1838
							60	95	00317		TSTB	(R0)	1841
							27	18	00319		BGEQ	39\$	
							1F	11	0031B		BRB	37\$	1844
							0F	E9	0031D	35\$:	BLBC	13(R6), 36\$	1849
						OC	A6	95	00321		TSTB	12(R6)	1852
							16	19	00324		BLSS	37\$	
			17	OD	A6		01	E1	00326		BBC	#1, 13(R6), 39\$	1859
							52	E8	0032B		BLBS	LAST FILE, 37\$	1862
							0F	11	0032E		BRB	38\$	1863
							08	E8	00330	36\$:	BLBS	13(R8), 37\$	1869
			09	OD	A8		01	E1	00334		BBC	#1, 13(R8), 39\$	1876
							52	E9	00339		BLBC	LAST FILE, 38\$	1879
							63	04	88 0033C	37\$:	BISB2	#4, (SMBITM)	
							63	08	88 0033F	38\$:	BISB2	#8, (SMBITM)	1880
							53	04	C0 00342	39\$:	ADDL2	#4, SMBITM	1883
							83	8F	D0 00345		MOVL	#3080196, (SMBITM)+	1888
					002F0004		63	D4	0034C		CLRL	(SMBITM)	1891
			03	11	A6		02	E1	0034E		BBC	#2, 17(R6), 40\$	1892
							04	88	00353		BISB2	#4, (SMBITM)	
						01AC	C6	B5	00356	40\$:	TSTW	428(R6)	1893
							0A	13	0035A		BEQL	41\$	

10	07	51	E9	0035C	BLBC	FIRST FILE, 41\$	1894		
	A8	02	88	0035F	BISB2	#2, 16(R8)	1897		
	63	02	8A	00363	BICB2	#2, (SMBITM)	1898		
	53	04	C0	00366	ADDL2	#4, SMBITM	1900		
	83	002B0004	8F	D0	00369	MOVL	#2818052, (SMBITM)+	1905	
	83	017D	C6	9A	00370	MOVZBL	381(R6), (SMBITM)+	1908	
		0134	C6	DD	00375	PUSHL	308(R6)	1914	
00000000G	EF	01	FB	00379	CALLS	#1, READ RECORD			
	83	00B0	C0	9B	00380	MOVZBW	176(QSMQ), (SMBITM)+	1915	
	83		2C	B0	00385	MOVW	#44, (SMBITM)+	1916	
	51	00B0	C0	9A	00388	MOVZBL	176(QSMQ), R1	1919	
63	00B1	C0	51	28	0038D	MOV3	R1, 177(QSMQ), (SMBITM)	1921	
		0134	C6	DD	00393	PUSHL	308(R6)	1922	
00000000G	EF	01	FB	00397	CALLS	#1, RELEASE RECORD			
	83	00310004	8F	D0	0039E	MOVL	#3211268, (SMBITM)+	1927	
	83	0156	C9	3C	003A5	MOVZWL	342(SFM), (SMBITM)+	1930	
	83	00350008	8F	D0	003AA	MOVL	#3473416, (SMBITM)+	1936	
	83	013C	C6	7D	003B1	MOVQ	316(R6), (SMBITM)+	1939	
	83	00360004	8F	D0	003B6	MOVL	#3538948, (SMBITM)+	1945	
	83	015C	C9	9A	003BD	MOVZBL	348(SFM), (SMBITM)+	1948	
	83	00370004	8F	D0	003C2	MOVL	#3604484, (SMBITM)+	1954	
	83	0144	C6	D0	003C9	MOVL	324(R6), (SMBITM)+	1957	
	83	0038000C	8F	D0	003CE	MOVL	#3670028, (SMBITM)+	1963	
63	0148	C6	0C	28	003D5	MOV3	#12, 328(R6), (SMBITM)	1969	
			83	D4	003DB	CLRL	(SMBITM)+	1974	
	08	AE	9E	003DD	MOVAB	SMBMSG, SMBMSG_DESC+4	1981		
04	AE	53	08	AE	C3	003E2	SUBL3	SMBMSG_DESC+4, SMBITM, SMBMSG_DESC	1982
			04	AE	9F	003E8	PUSHAB	SMBMSG_DESC	1983
			58	DD	003EB	PUSHL	R8		
	FBCA	CF	02	FB	003ED	CALLS	#2, SEND SYMBIONT_MESSAGE		
	70	A8	00	BE	D0	003F2	MOVL	@0(SP), T12(R8)	1988
	00EC	C6	14	AC	D0	003F7	MOVL	SQR N, 236(R6)	1993
	017B	C6		6A	90	003FD	MOVB	(R10), 379(R6)	1994
	0178	C6		6B	90	00402	MOVB	(R11), 376(R6)	1995
	00F0	C6	14	AC	D0	00407	MOVL	SQR N, 240(R6)	1996
			0180	C6	9F	0040D	PUSHAB	384(R6)	1999
				20	DD	00411	PUSHL	#32	
00000000G	EF	02	FB	00413	CALLS	#2, DEALLOCATE_VARIABLE_DATA			
	10	A6	18	88	0041A	BISB2	#24, 16(R6)	2001	
			00	BE	DD	0041E	PUSHL	@0(SP)	2002
00000000G	EF	01	FB	00421	CALLS	#1, RELEASE_RECORD			
			04	00428	RET				2003

; Routine Size: 1065 bytes, Routine Base: CODE + 011A

```

: 970 2004 1 GLOBAL ROUTINE STOP_SYMBIONT_TASK(SMQ_N,SMQ,SJH_N,SJH): NOVALUE=
: 971 2005 1
: 972 2006 1 '++
: 973 2007 1
: 974 2008 1 FUNCTIONAL DESCRIPTION:
: 975 2009 1 This routine sends the 'stop task' message to a symbiont.
: 976 2010 1
: 977 2011 1 INPUT PARAMETERS:
: 978 2012 1 SMQ_N - Record number of SMQ.
: 979 2013 1 SMQ - Pointer to SMQ.
: 980 2014 1 SJH_N - Record number of SJH.
: 981 2015 1 SJH - Pointer to SJH.
: 982 2016 1
: 983 2017 1 IMPLICIT INPUTS:
: 984 2018 1 NONE
: 985 2019 1
: 986 2020 1 OUTPUT PARAMETERS:
: 987 2021 1 NONE
: 988 2022 1
: 989 2023 1 IMPLICIT OUTPUTS:
: 990 2024 1 NONE
: 991 2025 1
: 992 2026 1 ROUTINE VALUE:
: 993 2027 1 NONE
: 994 2028 1
: 995 2029 1 SIDE EFFECTS:
: 996 2030 1 NONE
: 997 2031 1
: 998 2032 1 --
: 999 2033 1
1000 2034 2 BEGIN
1001 2035 2 MAP
1002 2036 2 SMQ: REF BBLOCK, ! Pointer to SMQ
1003 2037 2 SJH: REF BBLOCK; ! Pointer to SJH
1004 2038 2 LOCAL
1005 2039 2 SMBMSG: BBLOCK[JBC$K_SMBMBXSIZ], ! Message buffer
1006 2040 2 SMBITM: REF BBLOCK, ! Cursor for message items
1007 2041 2 SMBMSG_DESC: VECTOR[2]; ! Descriptor for message buffer
1008 2042 2
1009 2043 2
1010 2044 2 ! Message header.
1011 2045 2
1012 2046 2 SMBMSG[SMBMSG$W_REQUEST_CODE] = SMBMSG$K_STOP_TASK;
1013 2047 2 SMBMSG[SMBMSG$B_STRUCTURE_LEVEL] = SMBMSG$K_STRUCTURE_LEVEL;
1014 2048 2 SMBMSG[SMBMSG$B_STREAM_INDEX] = .SMQ[SMQ$B_STREAM_INDEX];
1015 2049 2 SMBITM = SMBMSG + SMBMSG$S_REQUEST_HEADER;
1016 2050 2
1017 2051 2
1018 2052 2 ! Reason for stop.
1019 2053 2
1020 2054 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
1021 2055 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_STOP_CONDITION;
1022 2056 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
1023 2057 2 .SMBITM = JBC$JOBABORT OR STS$K_ERROR;
1024 2058 2 IF .SJH[SJH$V_REQUEUE] THEN .SMBITM = JBC$_JOBREQUEUE OR STS$K_ERROR;
1025 2059 2 SMBITM = .SMBITM + 4;
: 1026 2060 2
```

```

: 1027      2061 2
: 1028      2062 2 ! Trailing zero item.
: 1029      2063 2
: 1030      2064 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 0;
: 1031      2065 2 SMBITM[SMBMSG$W_ITEM_CODE] = 0;
: 1032      2066 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 1033      2067 2
: 1034      2068 2
: 1035      2069 2 ! Send the message to the symbiont.
: 1036      2070 2
: 1037      2071 2 SMBMSG_DESC[1] = SMBMSG;
: 1038      2072 2 SMBMSG_DESC[0] = .SMBITM - .SMBMSG_DESC[1];
: 1039      2073 2 SEND_SYMBIONT_MESSAGE(.SMQ, SMBMSG_DESC);
: 1040      2074 1 END;

```

			0004 00000	.ENTRY	STOP SYMBIONT_TASK, Save R2	: 2004
			CE 9E 00002	MOVAB	-1032(SP), SP	:
08	AE	FBF8	07 B0 00007	MOVW	#7, SMBMSG	: 2046
0A	AE		01 90 0000B	MOVB	#1, SMBMSG+2	: 2047
	S2	08	AC D0 0000F	MOVL	SMQ, R2	: 2048
0B	AE	0117	C2 90 00013	MOVB	279(R2), SMBMSG+3	:
	S1	0C	AE 9E 00019	MOVAB	SMBMSG+4, SMBITM	: 2049
	81	00340004	8F D0 0001D	MOVL	#3407876, (SMBITM)+	: 2054
	61	00048082	8F D0 00024	MOVL	#295042, (SMBITM)	: 2057
	S0	10	AC D0 0002B	MOVL	SJH, R0	: 2058
	07	11	A0 E9 0002F	BLBC	17(R0), 1\$:
	61	000480E2	8F D0 00033	MOVL	#295138, (SMBITM)	:
	S1		04 C0 0003A 1\$:	ADDL2	#4, SMBITM	: 2059
			81 D4 0003D	CLRL	(SMBITM)+	: 2064
	04	08	AE 9E 0003F	MOVAB	SMBMSG, SMBMSG_DESC+4	: 2071
6E	S1	04	AE C3 00044	SUBL3	SMBMSG_DESC+4, SMBITM, SMBMSG_DESC	: 2072
		4004	8F BB 00049	PUSHR	#*M<R2,SP>	: 2073
	FB41	CF	02 FB 0004D	CALLS	#2, SEND_SYMBIONT_MESSAGE	:
			04 00052	RET		: 2074

: Routine Size: 83 bytes, Routine Base: CODE + 0543

```

: 1042 2075 1 GLOBAL ROUTINE PAUSE_SYMBIONT_TASK(SMQ_N,SMQ): NOVALUE=
: 1043 2076 1
: 1044 2077 1 :++
: 1045 2078 1
: 1046 2079 1 : FUNCTIONAL DESCRIPTION:
: 1047 2080 1 : This routine sends the "pause task" message to a symbiont.
: 1048 2081 1
: 1049 2082 1 : INPUT PARAMETERS:
: 1050 2083 1 : SMQ_N - Record number of SMQ.
: 1051 2084 1 : SMQ - Pointer to SMQ.
: 1052 2085 1
: 1053 2086 1 : IMPLICIT INPUTS:
: 1054 2087 1 : NONE
: 1055 2088 1
: 1056 2089 1 : OUTPUT PARAMETERS:
: 1057 2090 1 : NONE
: 1058 2091 1
: 1059 2092 1 : IMPLICIT OUTPUTS:
: 1060 2093 1 : NONE
: 1061 2094 1
: 1062 2095 1 : ROUTINE VALUE:
: 1063 2096 1 : NONE
: 1064 2097 1
: 1065 2098 1 : SIDE EFFECTS:
: 1066 2099 1 : NONE
: 1067 2100 1
: 1068 2101 1 :--
: 1069 2102 1
: 1070 2103 2 BEGIN
: 1071 2104 2 MAP
: 1072 2105 2 SMQ: REF BBLOCK; ! Pointer to SMQ
: 1073 2106 2 LOCAL
: 1074 2107 2 SMBMSG: BBLOCK[JBC$K_SMBMBXSIZ], ! Message buffer
: 1075 2108 2 SMBITM: REF BBLOCK, ! Cursor for message items
: 1076 2109 2 SMBMSG_DESC: VECTOR[2]; ! Descriptor for message buffer
: 1077 2110 2
: 1078 2111 2
: 1079 2112 2 ! Message header.
: 1080 2113 2
: 1081 2114 2 SMBMSG[SMBMSG$W_REQUEST_CODE] = SMBMSG$K_PAUSE_TASK;
: 1082 2115 2 SMBMSG[SMBMSG$B_STRUCTURE_LEVEL] = SMBMSG$K_STRUCTURE_LEVEL;
: 1083 2116 2 SMBMSG[SMBMSG$B_STREAM_INDEX] = .SMQ[SMQ$B_STREAM_INDEX];
: 1084 2117 2 SMBITM = SMBMSG + SMBMSG$S_REQUEST_HEADER;
: 1085 2118 2
: 1086 2119 2
: 1087 2120 2 ! Trailing zero item.
: 1088 2121 2
: 1089 2122 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 0;
: 1090 2123 2 SMBITM[SMBMSG$W_ITEM_CODE] = 0;
: 1091 2124 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 1092 2125 2
: 1093 2126 2
: 1094 2127 2 ! Send the message to the symbiont.
: 1095 2128 2
: 1096 2129 2 SMBMSG_DESC[1] = SMBMSG;
: 1097 2130 2 SMBMSG_DESC[0] = .SMBITM - .SMBMSG_DESC[1];
: 1098 2131 2 SEND_SYMBIONT_MESSAGE(.SMQ, SMBMSG_DESC);

```

```

: 1099      2132 2
: 1100      2133 2
: 1101      2134 2 ! Update SMQ.
: 1102      2135 2 !
: 1103      2136 2 SMQ[SMQ$V_PAUSING] = TRUE;
: 1104      2137 1 END;

```

```

                                0004 00000      .ENTRY PAUSE_SYMBIONT_TASK, Save R2      : 2075
                                CE 9E 00002      MOVAB -1024(SP), SP      :
                                01 B0 00007      MOVW #1, SMBMSG      : 2114
                                02 AE 0000A      MOVB #1, SMBMSG+2      : 2115
                                52 08 AC D0 0000E      MOVL SMQ, R2      : 2116
                                03 AE 0117 C2 90 00012      MOVB 279(R2), SMBMSG+3      :
                                50 04 AE 9E 00018      MOVAB SMBMSG+4, SMBITM      : 2117
                                80 D4 0001C      CLRL (SMBITM)+      : 2122
                                5E DD 0001E      PUSHL SP      : 2129
                                7E 50 6E C3 00020      SUBL3 SMBMSG_DESC+4, SMBITM, SMBMSG_DESC      : 2130
                                4004 8F BB 00024      PUSHR #^M<R2,SP>      : 2131
                                FB13 CF 02 FB 00028      CALLS #2, SEND_SYMBIONT_MESSAGE      :
                                10 A2 08 88 0002D      BISB2 #8, 16(R2)      : 2136
                                04 00031      RET      : 2137

```

: Routine Size: 50 bytes, Routine Base: CODE + 0596

```

1106 2138 1 GLOBAL ROUTINE RESUME_SYMBIONT_TASK(SMQ_N,SMQ,FLAGS,ALIGNMENT_PAGES,RELATIVE_PAGE,SEARCH_LENGTH,SEARCH_ADDRE
1107 2139 1
1108 2140 1 !++
1109 2141 1
1110 2142 1 FUNCTIONAL DESCRIPTION:
1111 2143 1 This routine sends the "resume task" message to a symbiont.
1112 2144 1
1113 2145 1 INPUT PARAMETERS:
1114 2146 1 SMQ_N - Record number of SMQ.
1115 2147 1 SMQ - Pointer to SMQ.
1116 2148 1 FLAGS - Resume control flags.
1117 2149 1 ALIGNMENT_PAGES - Number of alignment pages (or 0).
1118 2150 1 RELATIVE_PAGE - Relative page position (or 0).
1119 2151 1 SEARCH_LENGTH - Descriptor for search string (or 0).
1120 2152 1 SEARCH_ADDRESS -
1121 2153 1
1122 2154 1 IMPLICIT INPUTS:
1123 2155 1 NONE
1124 2156 1
1125 2157 1 OUTPUT PARAMETERS:
1126 2158 1 NONE
1127 2159 1
1128 2160 1 IMPLICIT OUTPUTS:
1129 2161 1 NONE
1130 2162 1
1131 2163 1 ROUTINE VALUE:
1132 2164 1 NONE
1133 2165 1
1134 2166 1 SIDE EFFECTS:
1135 2167 1 NONE
1136 2168 1
1137 2169 1 !--
1138 2170 1
1139 2171 2 BEGIN
1140 2172 2 MAP
1141 2173 2 SMQ: REF BBLOCK, ! Pointer to SMQ
1142 2174 2 FLAGS: BBLOCK; ! Resume control flags
1143 2175 2 LOCAL
1144 2176 2 SMBMSG: BBLOCK[JBC$K_SMBMBXSIZ],! Message buffer
1145 2177 2 SMBITM: REF BBLOCK, ! Cursor for message items
1146 2178 2 SMBMSG_DESC: VECTOR[2]; ! Descriptor for message buffer
1147 2179 2
1148 2180 2
1149 2181 2 ! Message header.
1150 2182 2
1151 2183 2 SMBMSG[SMBMSG$W_REQUEST_CODE] = SMBMSG$K_RESUME_TASK;
1152 2184 2 SMBMSG[SMBMSG$B_STRUCTURE_LEVEL] = SMBMSG$K_STRUCTURE_LEVEL;
1153 2185 2 SMBMSG[SMBMSG$B_STREAM_INDEX] = .SMQ[SMQ$B_STREAM_INDEX];
1154 2186 2 SMBITM = SMBMSG + SMBMSG$S_REQUEST_HEADER;
1155 2187 2
1156 2188 2
1157 2189 2 ! Alignment pages.
1158 2190 2
1159 2191 2 IF .ALIGNMENT_PAGES NEQ 0
1160 2192 2 THEN
1161 2193 3 BEGIN
1162 2194 3 SMBITM[SMBMSG$W_ITEM_SIZE] = 4;

```

```

: 1163      2195      3      SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_ALIGNMENT_PAGES;
: 1164      2196      3      SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 1165      2197      3      .SMBITM = .ALIGNMENT_PAGES;
: 1166      2198      3      SMBITM = .SMBITM + 4;
: 1167      2199      3      END;
: 1168      2200
: 1169      2201
: 1170      2202      2      ! File repositioning.
: 1171      2203      2      !
: 1172      2204      2      IF .RELATIVE_PAGE NEQ 0
: 1173      2205      2      THEN
: 1174      2206      2      BEGIN
: 1175      2207      2      SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
: 1176      2208      2      SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_RELATIVE_PAGE;
: 1177      2209      2      SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 1178      2210      2      .SMBITM = .RELATIVE_PAGE;
: 1179      2211      2      SMBITM = .SMBITM + 4;
: 1180      2212      2      END;
: 1181      2213
: 1182      2214
: 1183      2215      2      ! Request control flags.
: 1184      2216      2      !
: 1185      2217      2      IF .FLAGS NEQ 0 OR .ALIGNMENT_PAGES NEQ 0
: 1186      2218      2      THEN
: 1187      2219      2      BEGIN
: 1188      2220      2      SMBITM[SMBMSG$W_ITEM_SIZE] = 4;
: 1189      2221      2      SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_REQUEST_CONTROL;
: 1190      2222      2      SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 1191      2223      2      .SMBITM = 0;
: 1192      2224      2      IF .FLAGS[ISRV V ALIGNMENT_MASK]
: 1193      2225      2      THEN SMBITM[SMBMSG$V_ALIGNMENT_MASK] = TRUE;
: 1194      2226      2      IF .ALIGNMENT_PAGES NEQ 0
: 1195      2227      2      THEN SMBITM[SMBMSG$V_PAUSE_COMPLETED] = TRUE;
: 1196      2228      2      IF .FLAGS[ISRV V TOP_OF_FILE]
: 1197      2229      2      THEN SMBITM[SMBMSG$V_TOP_OF_FILE] = TRUE;
: 1198      2230      2      SMBITM = .SMBITM + 4;
: 1199      2231      2      END;
: 1200      2232
: 1201      2233
: 1202      2234      2      ! Search string.
: 1203      2235      2      !
: 1204      2236      2      IF .SEARCH_LENGTH NEQ 0
: 1205      2237      2      THEN
: 1206      2238      2      BEGIN
: 1207      2239      2      SMBITM[SMBMSG$W_ITEM_SIZE] = .SEARCH_LENGTH;
: 1208      2240      2      SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_SEARCH_STRING;
: 1209      2241      2      SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 1210      2242      2      MOV3(
: 1211      2243      2      SEARCH_LENGTH,
: 1212      2244      2      .SEARCH_ADDRESS,
: 1213      2245      2      .SMBITM; ... SMBITM);
: 1214      2246      2      END;
: 1215      2247
: 1216      2248
: 1217      2249      2      ! Trailing zero item.
: 1218      2250      2      !
: 1219      2251      2      SMBITM[SMBMSG$W_ITEM_SIZE] = 0;
```



```

: 1220      2252 2 SMBITM[SMBMSG$W ITEM CODE] = 0;
: 1221      2253 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 1222      2254 2
: 1223      2255 2
: 1224      2256 2 ! Send the message to the symbiont.
: 1225      2257 2
: 1226      2258 2 SMBMSG_DESC[1] = SMBMSG;
: 1227      2259 2 SMBMSG_DESC[0] = .SMBITM - .SMBMSG_DESC[1];
: 1228      2260 2 SEND_SYMBIONT_MESSAGE(.SMQ, SMBMSG_DESC);
: 1229      2261 2
: 1230      2262 2
: 1231      2263 2 ! Update SMQ.
: 1232      2264 2
: 1233      2265 2 SMQ[SMQ$V RESUMING] = TRUE;
: 1234      2266 2 IF .ALIGNMENT_PAGES NEQ 0 THEN SMQ[SMQ$V_ALIGNING] = TRUE;
: 1235      2267 1 END;

```

			00FC 00000		.ENTRY RESUME_SYMBIONT_TASK, Save R2,R3,R4,R5,R6,- ;	2138
					R7	
	08	5E	FBF8	CE 9E 00002	MOVAB -1032(SP), SP	
	0A	AE		03 80 00007	MOVW #3, SMBMSG	2183
		56	08	01 90 0000B	MOVB #1, SMBMSG+2	2184
	0B	AE	0117	AC D0 0000F	MOVL SMQ, R6	2185
		53	0C	C6 90 00013	MOVB 279(R6) SMBMSG+3	
				AE 9E 00019	MOVAB SMBMSG+4, SMBITM	2186
				57 D4 0001D	CLRL R7	2191
			10	AC D5 0001F	TSTL ALIGNMENT_PAGES	
				0D 13 00022	BEQL 1\$	
				57 D6 00024	INCL R7	
		83	00040004	8F D0 00026	MOVL #262148, (SMBITM)+	2194
		83	10	AC D0 0002D	MOVL ALIGNMENT_PAGES, (SMBITM)+	2197
			14	AC D5 00031 1\$	TSTL RELATIVE_PAGE	2204
				0B 13 00034	BEQL 2\$	
		83	002E0004	8F D0 00036	MOVL #3014660, (SMBITM)+	2207
		83	14	AC D0 0003D	MOVL RELATIVE_PAGE, (SMBITM)+	2210
			0C	AC D5 00041 2\$	TSTL FLAGS	2217
				03 12 00044	BNEQ 3\$	
		21	002F0004	57 E9 00046	BLBC R7, 7\$	
		83		8F D0 00049 3\$	MOVL #3080196, (SMBITM)+	2220
				63 D4 00050	CLRL (SMBITM)	2223
		03	0C	AC E9 00052	BLBC FLAGS, 4\$	2224
		63		01 88 00056	BISB2 #1, (SMBITM)	2225
		03		57 E9 00059 4\$	BLBC R7, 5\$	2226
		63		02 88 0005C	BISB2 #2, (SMBITM)	2227
	03	0C		01 E1 0005F 5\$	BBC #1, FLAGS, 6\$	2228
		63		08 88 00064	BISB2 #8, (SMBITM)	2229
		53		04 C0 00067 6\$	ADDL2 #4, SMBITM	2230
			18	AC D5 0006A 7\$	TSTL SEARCH_LENGTH	2236
				0D 13 0006D	BEQL 8\$	
		83	18	AC B0 0006F	MOVW SEARCH_LENGTH, (SMBITM)+	2239
		83		32 B0 00073	MOVW #50, (SMBITM)+	2240
	63	1C	BC	AC 28 00076	MOVCL SEARCH_LENGTH, @SEARCH_ADDRESS, (SMBITM)	2245
				83 D4 0007C 8\$	CLRL (SMBITM)+	2251

SYMBIONT
V04-000

Symbiont communication

F 13
16-Sep-1984 00:37:14
14-Sep-1984 12:37:15

VAX-11 Bliss-32 V4.0-742
[JOBCTL.SRC]SYMBIONT.B32;1

Page 40
(9)

6E	04	AE	08	AE	9E	0007E	MOVAB	SMBMSG, SMBMSG_DESC+4	:	2258
		53	04	AE	C3	00083	SUBL3	SMBMSG_DESC+4, SMBITM, SMBMSG_DESC	:	2259
			4040	8F	BB	00088	PUSHR	#^M<R6,SP>	:	2260
	FA7D	CF		02	FB	0008C	CALLS	#2, SEND SYMBIONT_MESSAGE	:	
	10	A6	40	8F	88	00091	BISB2	#64, 16(R6)	:	2265
		04		57	E9	00096	BLBC	R7, 9\$:	2266
	10	A6		01	88	00099	BISB2	#1, 16(R6)	:	
				04	0009D	9\$:	RET		:	2267

; Routine Size: 158 bytes, Routine Base: CODE + 05C8

```

: 1237 2268 1 GLOBAL ROUTINE START_SYMBIONT_STREAM(SMQ_N,SMQ)=
: 1238 2269 1
: 1239 2270 1 !++
: 1240 2271 1
: 1241 2272 1 : FUNCTIONAL DESCRIPTION:
: 1242 2273 1 : This routine starts a symbiont stream. If necessary, it creates a
: 1243 2274 1 : symbiont process and then sends the "start stream" message.
: 1244 2275 1
: 1245 2276 1 : INPUT PARAMETERS:
: 1246 2277 1 : SMQ_N - Record number of SMQ.
: 1247 2278 1 : SMQ - Pointer to SMQ.
: 1248 2279 1
: 1249 2280 1 : IMPLICIT INPUTS:
: 1250 2281 1 : NONE
: 1251 2282 1
: 1252 2283 1 : OUTPUT PARAMETERS:
: 1253 2284 1 : NONE
: 1254 2285 1
: 1255 2286 1 : IMPLICIT OUTPUTS:
: 1256 2287 1 : NONE
: 1257 2288 1
: 1258 2289 1 : ROUTINE VALUE:
: 1259 2290 1 : Completion status.
: 1260 2291 1
: 1261 2292 1 : SIDE EFFECTS:
: 1262 2293 1 : NONE
: 1263 2294 1
: 1264 2295 1 :--
: 1265 2296 1
: 1266 2297 2 BEGIN
: 1267 2298 2 MAP
: 1268 2299 2 SMQ: REF BBLOCK; ! Pointer to SMQ
: 1269 2300 2 LOCAL
: 1270 2301 2 SCT: REF BBLOCK, ! Pointer to SCT
: 1271 2302 2 STM, ! Stream index
: 1272 2303 2 PRCNAM_BUFFER: VECTOR[15,BYTE], ! Buffer for process name
: 1273 2304 2 PRCNAM_DESC: VECTOR[2], ! Descriptor for process name
: 1274 2305 2 PRCNAM, ! Process name parameter
: 1275 2306 2 IMAGE_BUFFER: VECTOR[63,BYTE], ! Buffer for image name
: 1276 2307 2 IMAGE_DESC: VECTOR[2], ! Descriptor for image name
: 1277 2308 2 MAILBOX_BUFFER: VECTOR[30,BYTE], ! Buffer for mailbox name
: 1278 2309 2 MAILBOX_DESC: VECTOR[2], ! Descriptor for mailbox name
: 1279 2310 2 GETDVI_LIST: BBLOCK[16], ! $GETDVI item list
: 1280 2311 2 IOSB: VECTOR[4,WORD], ! I/O status block
: 1281 2312 2 STATUS_1, ! Status return
: 1282 2313 2 STATUS_2, ! Status return
: 1283 2314 2 STATUS_3, ! Status return
: 1284 2315 2 SMBMSG: BBLOCK[JBC$K_SMBMBXSIZ], ! Message buffer
: 1285 2316 2 SMBITM: REF BBLOCK, ! Cursor for message items
: 1286 2317 2 SMBMSG_DESC: VECTOR[2]; ! Descriptor for message buffer
: 1287 2318 2
: 1288 2319 2
: 1289 2320 2 OWN
: 1290 2321 2 PRIVILEGE_MASK: BBLOCK[8] ! Symbiont privileges
: 1291 2322 2 PSECT(CODE) PRESET(
: 1292 2323 2 [PRV$V_SETPRV] = TRUE);
: 1293 2324 2

```

```

1294 2325 2 ! Find a suitable symbiont.
1295 2326 2 !
1296 2327 2 !
1297 2328 2 SCT = .SYMBIONT CONTROL;
1298 2329 2 WHILE .SCT NEQ 0 DO
1299 2330 2 BEGIN
1300 2331 2
1301 2332 2 ! Locate a symbiont that is executing the desired image, that is not
1302 2333 2 ! deleting itself, and has an available stream.
1303 2334 2 !
1304 2335 2 IF CH$EQL(
1305 2336 2     CHR$RCHAR(SMQ[SMQ$T PROCESSOR]),
1306 2337 2     SMQ[SMQ$T PROCESSOR] + 1,
1307 2338 2     CHR$RCHAR(SCT[SCT T PROCESSOR]),
1308 2339 2     SCT[SCT T PROCESSOR] + 1)
1309 2340 2 AND NOT .SCT[SCT_V_DELETING]
1310 2341 2 AND NOT FFC(
1311 2342 2     %REF(0), %REF(.SCT[SCT_B_MAXSTREAMS]), SCT[SCT_L_BITMAP], STM)
1312 2343 2 THEN
1313 2344 2     EXITLOOP;
1314 2345 2
1315 2346 2
1316 2347 2 ! Advance to next.
1317 2348 2 !
1318 2349 2 SCT = .SCT[SCT_L_FLINK];
1319 2350 2 END;
1320 2351 2
1321 2352 2
1322 2353 2 ! No suitable symbiont found; create a new one.
1323 2354 2 !
1324 2355 2 IF .SCT EQL 0
1325 2356 2 THEN
1326 2357 2 BEGIN
1327 2358 2     SCT = ALLOCATE MEMORY();
1328 2359 2     SCT[SCT_L_FLINK] = .SYMBIONT CONTROL;
1329 2360 2     SCT[SCT_B_MAXSTREAMS] = SCT_R_MAXSTREAMS;
1330 2361 2     CH$MOVE(
1331 2362 2         SMQ$S PROCESSOR,
1332 2363 2         SMQ[SMQ$T PROCESSOR],
1333 2364 2         SCT[SCT T PROCESSOR]);
1334 2365 2     SYMBIONT_CONTROL = .SCT;
1335 2366 2     STM = 0;
1336 2367 2 END;
1337 2368 2
1338 2369 2
1339 2370 2 ! Create a symbiont process if needed.
1340 2371 2 !
1341 2372 2 IF .SCT[SCT_L_BITMAP] EQL 0
1342 2373 2 THEN
1343 2374 2 BEGIN
1344 2375 2
1345 2376 2 ! Set up the process name as "SYMBIONT_nnnn".
1346 2377 2 !
1347 2378 2 PRCNAM_DESC[0] = %ALLOCATION(PRCNAM_BUFFER);
1348 2379 2 PRCNAM_DESC[1] = PRCNAM_BUFFER;
1349 2380 2 SYMBIONT_COUNT = .SYMBIONT_COUNT + 1;
1350 2381 2 $FAO(

```

```

1351 P 2382 3 $DESCRIPTOR('SYMBIONT_!4ZL'),
1352 P 2383 3 PRCNAM_DESC,
1353 P 2384 3 PRCNAM_DESC,
1354 2385 3 .SYMBIONT_COUNT);
1355 2386 3
1356 2387 3
1357 2388 3 ! Set up the image name as 'SYS$SYSTEM:name.EXE'.
1358 2389 3
1359 2390 3 IMAGE_DESC[0] = %ALLOCATION(IMAGE_BUFFER);
1360 2391 3 IMAGE_DESC[1] = IMAGE_BUFFER;
1361 P 2392 3 $FAO(
1362 P 2393 3 $DESCRIPTOR('SYS$SYSTEM:!AC.EXE'),
1363 P 2394 3 IMAGE_DESC,
1364 P 2395 3 IMAGE_DESC,
1365 P 2396 3 (IF C$RCHAR(SMQ[SMQ$SMQ$PROCESSOR]) EQL 0
1366 P 2397 3 THEN UPLIT BYTE (XASCIC 'PRTSMB')
1367 2398 3 ELSE SMQ[SMQ$SMQ$PROCESSOR]));
1368 2399 3
1369 2400 3 ! Create the symbiont input mailbox.
1370 2401 3
1371 2402 3
1372 P 2403 3 STATUS_1 = $CREMBX(
1373 P 2404 3 CHAN=SCT[SCT_W_MAILBOX],
1374 P 2405 3 MAXMSG=JBC$K-SMBMBXSIZ,
1375 P 2406 3 BUFQUO=JBC$K-SMBMBXSIZ,
1376 2407 3 PROMSK=XB'11T111110000000'); ! S:RWED, O:RWED, G, W
1377 2408 3 IF NOT .STATUS_1
1378 2409 3 THEN
1379 2410 3 BEGIN
1380 2411 3 SYMBIONT_CONTROL = .SCT[SCT_L_FLINK];
1381 2412 3 DEALLOCATE MEMORY(.SCT);
1382 2413 3 RETURN .STATUS_1;
1383 2414 3 END;
1384 2415 3
1385 2416 3
1386 2417 3 ! Get a descriptor for the mailbox device name.
1387 2418 3
1388 2419 3 MAILBOX_DESC[0] = 0;
1389 2420 3 MAILBOX_DESC[1] = MAILBOX_BUFFER;
1390 2421 3 GETDVI_LIST[0,0,16,0] = %ALLOCATION(MAILBOX_BUFFER);
1391 2422 3 GETDVI_LIST[2,0,16,0] = DVI$DEVNAM;
1392 2423 3 GETDVI_LIST[4,0,32,0] = MAILBOX_BUFFER;
1393 2424 3 GETDVI_LIST[8,0,32,0] = MAILBOX_DESC;
1394 2425 3 GETDVI_LIST[12,0,32,0] = 0;
1395 P 2426 3 STATUS_2 = $GETDVIW(
1396 P 2427 3 EFN=JBC$K_SYNC_EF,
1397 P 2428 3 CHAN=.SCT[SCT_W_MAILBOX],
1398 P 2429 3 ITMLST=GETDVI_LIST,
1399 2430 3 IOSB=IOSB);
1400 2431 3 IF NOT .STATUS_2
1401 2432 3 THEN
1402 2433 3 BEGIN
1403 2434 3 $DASSGN(CHAN=.SCT[SCT_W_MAILBOX]);
1404 2435 3 SYMBIONT_CONTROL = .SCT[SCT_L_FLINK];
1405 2436 3 DEALLOCATE MEMORY(.SCT);
1406 2437 3 RETURN .STATUS_2;
1407 2438 3 END;

```

```

: 1408      2439      3
: 1409      2440      3
: 1410      2441      3
: 1411      2442      3
: 1412      2443      3
: 1413      2444      3
: 1414      2445      4
: 1415      2446      4
: 1416      2447      4
: 1417      2448      4
: 1418      P 2449      4
: 1419      P 2450      4
: 1420      P 2451      4
: 1421      P 2452      4
: 1422      P 2453      4
: 1423      P 2454      4
: 1424      P 2455      4
: 1425      P 2456      4
: 1426      P 2457      4
: 1427      P 2458      4
: 1428      P 2459      4
: 1429      2460      4
: 1430      2461      4
: 1431      2462      4
: 1432      2463      4
: 1433      2464      5
: 1434      2465      5
: 1435      2466      5
: 1436      2467      5
: 1437      2468      5
: 1438      2469      5
: 1439      2470      5
: 1440      2471      5
: 1441      2472      6
: 1442      2473      6
: 1443      2474      6
: 1444      2475      6
: 1445      2476      6
: 1446      2477      6
: 1447      2478      5
: 1448      2479      5
: 1449      2480      5
: 1450      2481      4
: 1451      2482      5
: 1452      2483      5
: 1453      2484      5
: 1454      2485      5
: 1455      2486      4
: 1456      2487      3
: 1457      2488      2
: 1458      2489      2
: 1459      2490      2
: 1460      2491      2
: 1461      2492      2
: 1462      2493      2
: 1463      2494      2
: 1464      2495      2

```

```

! The following loop is executed at most twice.
PRCNAM = PRCNAM_DESC;
WHILE TRUE DO
  BEGIN
    ! Create the symbiont process.
    !
    STATUS 3 = $CREPRC(
      PIDADR=SCT[SCT_L_PID],
      IMAGE=IMAGE_DESC,
      INPUT=MAILBOX_DESC,
      OUTPUT=JOBCTLMBX_DESC,
      ERROR=NLAO_DESC,
      PRVADR=PRIVILEGE_MASK,
      QUOTA=JBC_QUOTAS,
      PRCNAM=.PRCNAM,
      BASPRI=.SMQ[SMQ$B_BASE_PRIORITY],
      STSFLG=.IMAGE_DUMP_STSFLG,
      UIC=.JBC_UIC);

    IF NOT .STATUS_3
    THEN
      BEGIN
        ! Create failed. If the status is not 'duplicate process name', or
        ! if a create has already been tried with no name, give up.
        ! Otherwise, loop to try creation with no name.
        !
        IF .STATUS_3<0,16> NEQ SS$_DUPLNAM OR .PRCNAM EQL 0
        THEN
          BEGIN
            $DASSGN(CHAN=.SCT[SCT_W_MAILBOX]);
            SYMBIONT CONTROL = .SCT[SCT_L_FLINK];
            DEALLOCATE MEMORY(.SCT);
            SCAN_INCOMPLETE_SERVICES(ISRV_K_PURGE_SMQ, .SMQ_N);
            RETURN .STATUS_3;
          END;
          PRCNAM = 0;
        END
      ELSE
        BEGIN
          ENTER_PROCESS_DATA(PDE_K_SYMBIONT, .SCT[SCT_L_PID]);
          QUEUE_REFERENCE_COUNT = .QUEUE_REFERENCE_COUNT + 1;
          EXITLOOP;
        END;
      END;
    END;

    ! Update SMQ.
    !
    SMQ[SMQ$L_STREAM_SCT] = .SCT;
    SMQ[SMQ$B_STREAM_INDEX] = .SM;
    SMQ[SMQ$V_STARTING] = TRUE;

```

```

: 1465      2496 2 SMQ[SMQ$V_STOPPED] = FALSE;
: 1466      2497 2
: 1467      2498 2
: 1468      2499 2 ! Update SCT.
: 1469      2500 2
: 1470      2501 2 BITVECTOR[SCT[SCT_L_BITMAP], .STM] = TRUE;
: 1471      2502 2 VECTOR[SCT[SCT_L_QUEUES], .STM] = .SMQ_N;
: 1472      2503 2
: 1473      2504 2
: 1474      2505 2 ! Message header for the "start stream" command.
: 1475      2506 2
: 1476      2507 2 SMBMSG[SMBMSG$W_REQUEST_CODE] = SMBMSG$K_START_STREAM;
: 1477      2508 2 SMBMSG[SMBMSG$B_STRUCTURE_LEVEL] = SMBMSG$K_STRUCTURE_LEVEL;
: 1478      2509 2 SMBMSG[SMBMSG$B_STREAM_INDEX] = .SMQ[SMQ$B_STREAM_INDEX];
: 1479      2510 2 SMBITM = SMBMSG + SMBMSG$S_REQUEST_HEADER;
: 1480      2511 2
: 1481      2512 2
: 1482      2513 2 ! Device name.
: 1483      2514 2
: 1484      2515 2 IF CHRCHAR(SMQ[SMQ$T_DEVICE_NAME]) EQL 0
: 1485      2516 2 THEN
: 1486      2517 2 BEGIN
: 1487      2518 2   SMBITM[SMBMSG$W_ITEM_SIZE] = CHRCHAR(SMQ[SMQ$T_NAME]);
: 1488      2519 2   SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_DEVICE_NAME;
: 1489      2520 2   SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 1490      2521 2   MOV3(
: 1491      2522 2     %REF(CHRCHAR(SMQ[SMQ$T_NAME])),
: 1492      2523 2     SMQ[SMQ$T_NAME] + 1,
: 1493      2524 2     .SMBITM; ..., SMBITM);
: 1494      2525 2   END
: 1495      2526 2 ELSE
: 1496      2527 2 BEGIN
: 1497      2528 2   SMBITM[SMBMSG$W_ITEM_SIZE] = CHRCHAR(SMQ[SMQ$T_DEVICE_NAME]);
: 1498      2529 2   SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_DEVICE_NAME;
: 1499      2530 2   SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 1500      2531 2   MOV3(
: 1501      2532 2     %REF(CHRCHAR(SMQ[SMQ$T_DEVICE_NAME])),
: 1502      2533 2     SMQ[SMQ$T_DEVICE_NAME] + 1,
: 1503      2534 2     .SMBITM; ..., SMBITM);
: 1504      2535 2   END;
: 1505      2536 2
: 1506      2537 2
: 1507      2538 2 ! Queue name.
: 1508      2539 2
: 1509      2540 2 SMBITM[SMBMSG$W_ITEM_SIZE] = CHRCHAR(SMQ[SMQ$T_NAME]);
: 1510      2541 2 SMBITM[SMBMSG$W_ITEM_CODE] = SMBMSG$K_EXECUTOR_QUEUE;
: 1511      2542 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
: 1512      2543 2 MOV3(
: 1513      2544 2   %REF(CHRCHAR(SMQ[SMQ$T_NAME])),
: 1514      2545 2   SMQ[SMQ$T_NAME] + 1,
: 1515      2546 2   .SMBITM; ..., SMBITM);
: 1516      2547 2
: 1517      2548 2
: 1518      2549 2 ! Job reset modules.
: 1519      2550 2
: 1520      2551 2 SMBITM = FETCH_VARIABLE_ITEM(
: 1521      2552 2   SMQ$S_JOB_RESET_MODULES, SMQ[SMQ$T_JOB_RESET_MODULES],
```


	50	24	AB	9E	00098	MOVAB	P.AAE, R0		
			50	DD	0009C	PUSHL	R0		
			02	11	0009E	BRB	7\$		
			59	DD	000A0	PUSHL	R9		
		A0	AD	9F	000A2	PUSHAB	IMAGE_DESC		
		A0	AD	9F	000A5	PUSHAB	IMAGE_DESC		
		1C	AB	9F	000A8	PUSHAB	P.AAC		
00000000G	00		04	FB	000AB	CALLS	#4, SYSS\$FAO		
			7E	7C	000B2	CLRQ	-(SP)		2407
	7E	FF00	8F	3C	000B4	MOVZWL	#65280, -(SP)		
	7E	0400	8F	3C	000B9	MOVZWL	#1024, -(SP)		
	7E	0400	8F	3C	000BE	MOVZWL	#1024, -(SP)		
		06	A6	9F	000C3	PUSHAB	6(SCT)		
			7E	D4	000C6	CLRL	-(SP)		
00000000G	00		07	FB	000C8	CALLS	#7, SYSS\$CREMBX		
	52		50	D0	000CF	MOVL	R0, STATUS_1		
	4C		52	E9	000D2	BLBC	STATUS_1, 8\$		2408
		FF74	CD	7C	000D5	CLRQ	GETDVI_LIST+12		2425
FF7C	CD	80	AD	9E	000D9	MOVAB	MAILBOX_BUFFER, MAILBOX_DESC+4		2420
FF68	CD	0020001E	8F	D0	000DF	MOVL	#2097182, GETDVI_LIST		2421
FF6C	CD	80	AD	9E	000E8	MOVAB	MAILBOX_BUFFER, GETDVI_LIST+4		2423
FF70	CD	FF78	CD	9E	000EE	MOVAB	MAILBOX_DESC, GETDVI_LIST+8		2424
			7E	7C	000F5	CLRQ	-(SP)		2430
			7E	D4	000F7	CLRL	-(SP)		
		FF60	CD	9F	000F9	PUSHAB	IOSB		
		FF68	CD	9F	000FD	PUSHAB	GETDVI_LIST		
			7E	D4	00101	CLRL	-(SP)		
	7E	06	A6	3C	00103	MOVZWL	6(SCT), -(SP)		
			01	DD	00107	PUSHL	#1		
00000000G	00		08	FB	00109	CALLS	#8, SYSS\$GETDVIW		
	52		50	D0	00110	MOVL	R0, STATUS_2		
	1B		52	E8	00113	BLBS	STATUS_2, 9\$		2431
	7E	06	A6	3C	00116	MOVZWL	6(SCT), -(SP)		2434
00000000G	00		01	FB	0011A	CALLS	#1, SYSS\$DASSGN		
	6A		66	D0	00121	MOVL	(SCT), SYMBIONT_CONTROL		2435
			56	DD	00124	PUSHL	SCT		2436
00000000G	EF		01	FB	00126	CALLS	#1, DEALLOCATE_MEMORY		
	50		52	D0	0012D	MOVL	STATUS_2, R0		2437
			04	00130	RET				
	52	E8	AD	9E	00131	MOVAB	PRCNAM_DESC, PRCNAM		2443
			7E	D4	00135	CLRL	-(SP)		2460
		FF74	CA	DD	00137	PUSHL	IMAGE_DUMP_STSFLG		
			7E	D4	0013B	CLRL	-(SP)		
		0080	CA	DD	0013D	PUSHL	JBC_UIC		
	7E	0114	C7	9A	00141	MOVZBL	276(R7), -(SP)		
			52	DD	00146	PUSHL	PRCNAM		
		3C	AA	9F	00148	PUSHAB	JBC_QUOTAS		
		E8	AB	9F	0014B	PUSHAB	PRIVILEGE_MASK		
		00000000G	EF	9F	0014E	PUSHAB	NLAO_DESC		
		00000000G	EF	9F	00154	PUSHAB	JOBCTLMBX_DESC		
		FF78	CD	9F	0015A	PUSHAB	MAILBOX_DESC		
		A0	AD	9F	0015E	PUSHAB	IMAGE_DESC		
		08	A6	9F	00161	PUSHAB	8(SCT)		
00000000G	00		0D	FB	00164	CALLS	#13, SYSS\$CREPRC		
	53		50	D0	0016B	MOVL	R0, STATUS_3		
	36		53	E8	0016E	BLBS	STATUS_3, 73\$		2462
0094	8F		53	B1	00171	CMPW	STATUS_3, #148		2470

			04	12	00176	BNEQ	11\$		
			52	D5	00178	TSTL	PRCNAM		
			27	12	0017A	BNEQ	12\$		
			A6	3C	0017C	MOVZWL	6(SCT), -(SP)	11\$:	2473
	00000000G	7E	01	FB	00180	CALLS	#1, SYSSDASSGN		
		00	66	DO	00187	MOVL	(SCT), SYMBIONT_CONTROL		2474
		6A	56	DD	0018A	PUSHL	SCT		2475
	00000000G	EF	01	FB	0018C	CALLS	#1, DEALLOCATE_MEMORY		
			04	AC	DD	00193	PUSHL	SMQ_N	2476
			04	DD	00196	PUSHL	#4		
	00000000G	EF	02	FB	00198	CALLS	#2, SCAN_INCOMPLETE_SERVICES		
		50	53	DO	0019F	MOVL	STATUS_3, R0		2477
				04	001A2	RET			
			52	D4	001A3	CLRL	PRCNAM	12\$:	2479
			8E	11	001A5	BRB	10\$		2462
			08	A6	DD	001A7	PUSHL	8(SCT)	2483
			02	DD	001AA	PUSHL	#2		
	00000000G	EF	02	FB	001AC	CALLS	#2, ENTER_PROCESS_DATA		
			A8	AA	D6	001B3	INCL	QUEUE_REFERENCE_COUNT	2484
			56	DO	001B6	MOVL	SCT, 252(R7)	14\$:	2493
	00FC	C7	58	90	001BB	MOVB	STM, 279(R7)		2494
	0117	C7	01	88	001C0	BISB2	#1, 17(R7)		2495
	11	A7	02	8A	001C4	BICB2	#2, 17(R7)		2496
	11	A7	58	E2	001C8	BBSS	STM, 12(SCT), 15\$		2501
00	0C	A6	04	AC	DO	001CD	MOVL	SMQ_N, 60(SCT)[STM]	2502
	3C	A648	04	AC	DO	001CD	MOVL	SMQ_N, 60(SCT)[STM]	2502
	08	AE	04	BO	001D3	MOVW	#4, SMBMSG		2507
	0A	AE	01	90	001D7	MOVB	#1, SMBMSG+2		2508
	0B	AE	0117	C7	90	001DB	MOVB	279(R7), SMBMSG+3	2509
		53	0C	AE	9E	001E1	MOVAB	SMBMSG+4, SMBITM	2510
		50	50	A7	9A	001E5	MOVZBL	80(R7), R0	2515
			15	12	001E9	BNEQ	16\$		
		83	00B0	C7	9B	001EB	MOVZBW	176(R7), (SMBITM)+	2518
		83		09	BO	001F0	MOVW	#9, (SMBITM)+	2519
		51	00B0	C7	9A	001F3	MOVZBL	176(R7), R1	2522
63	00B1	C7	51	28	001F8	MOVC3	R1, 177(R7), (SMBITM)		2524
			0B	11	001FE	BRB	17\$		2515
		83	50	BO	00200	MOVW	R0, (SMBITM)+	16\$:	2528
		83	09	BO	00203	MOVW	#9, (SMBITM)+		2529
63	51	A7	50	28	00206	MOVC3	R0, 81(R7), (SMBITM)		2534
		83	00B0	C7	9B	0020B	MOVZBW	176(R7), (SMBITM)+	2540
		83	0C	BO	00210	MOVW	#12, (SMBITM)+		2541
		50	00B0	C7	9A	00213	MOVZBL	176(R7), R0	2544
63	00B1	C7	50	28	00218	MOVC3	R0, 177(R7), (SMBITM)		2546
			53	DD	0021E	PUSHL	SMBITM		2554
			1A	DD	00220	PUSHL	#26		2552
			0118	C7	9F	00222	PUSHAB	280(R7)	
			06	DD	00226	PUSHL	#6		
	00000000G	EF	04	FB	00228	CALLS	#4, FETCH_VARIABLE_ITEM		
		53	50	DO	0022F	MOVL	R0, SMBITM		
63		56	0088	C7	9A	00232	MOVZBL	136(R7), R6	2559
		56	10	A1	00237	ADDW3	#16, R6, (SMBITM)		
			58	D4	0023B	CLRL	R8		2560
			56	D5	0023D	TSTL	R6		
			05	12	0023F	BNEQ	18\$		
			58	D6	00241	INCL	R8		
		63	19	BO	00243	MOVW	#25, (SMBITM)		
	02	A3	1D	BO	00246	MOVW	#29, 2(SMBITM)	18\$:	2561

63	2B	53 AB		04 0C	C0 28	0024A 0024D	ADDL2 MOV3	#4, SMBITM #12, P.AAF, (SMBITM)	:	2562 2566
63	37	07 AB		58 09	E9 28	00252 00255	BLBC MOV3	R8, 19\$ #9, P.AAG, (SMBITM)	:	2572
63	7089	C7 83	424C542E	06 56	11 28	0025A 0025C	BRB MOV3	20\$ R6, 137(R7), (SMBITM)	:	2569 2577
				BF	D0	00262	MOVL	#1112298542, (SMBITM)+	:	2578
				93	D4	00269	CLRL	(SMBITM)+	:	2584
6E	04	AE 53	08 04 4080	AE AE	9E C3	0026B 00270	MOVAB SUBL3	SMBMSG, SMBMSG_DESC+4 SMBMSG_DESC+4, SMBITM, SMBMSG_DESC	:	2591 2592
	F798	CF 5C		EF 02	BB FB	00275 00279	PUSHR CALLS	#^M<R7,SP> #2, SEND_SYMBIONT_MESSAGE	:	2593
				01	D0	0027E	MOVL	#1, R0	:	2597
				04		00281	RET		:	

; Routine Size: 642 bytes, Routine Base: CODE + 06C0

```
1568 2598 1 GLOBAL ROUTINE STOP_SYMBIONT_STREAM(SMQ_N,SMQ): NOVALUE=
1569 2599 1
1570 2600 1 !++
1571 2601 1 !
1572 2602 1 FUNCTIONAL DESCRIPTION:
1573 2603 1 This routine sends the "stop stream" message to a symbiont.
1574 2604 1
1575 2605 1 INPUT PARAMETERS:
1576 2606 1 SMQ_N - Record number of SMQ.
1577 2607 1 SMQ - Pointer to SMQ.
1578 2608 1
1579 2609 1 IMPLICIT INPUTS:
1580 2610 1 NONE
1581 2611 1
1582 2612 1 OUTPUT PARAMETERS:
1583 2613 1 NONE
1584 2614 1
1585 2615 1 IMPLICIT OUTPUTS:
1586 2616 1 NONE
1587 2617 1
1588 2618 1 ROUTINE VALUE:
1589 2619 1 NONE
1590 2620 1
1591 2621 1 SIDE EFFECTS:
1592 2622 1 NONE
1593 2623 1
1594 2624 1 --
1595 2625 1
1596 2626 2 BEGIN
1597 2627 2 MAP
1598 2628 2 SMQ: REF BBLOCK; ! Pointer to SMQ
1599 2629 2 LOCAL
1600 2630 2 SCT: REF BBLOCK, ! Pointer to SCT
1601 2631 2 SMBMSG: BBLOCK[JBC$K_SMBMBXSIZ], ! Message buffer
1602 2632 2 SMBITM: REF BBLOCK, ! Cursor for message items
1603 2633 2 SMBMSG_DESC: VECTOR[2]; ! Descriptor for message buffer
1604 2634 2
1605 2635 2
1606 2636 2 ! Message header.
1607 2637 2 !
1608 2638 2 SMBMSG[SMBMSG$W_REQUEST CODE] = SMBMSG$K_STOP_STREAM;
1609 2639 2 SMBMSG[SMBMSG$B_STRUCTURE_LEVEL] = SMBMSG$K_STRUCTURE_LEVEL;
1610 2640 2 SMBMSG[SMBMSG$B_STREAM_INDEX] = .SMQ[SMQ$B_STREAM_INDEX];
1611 2641 2 SMBITM = SMBMSG + SMBMSG$S_ITEM_HEADER;
1612 2642 2
1613 2643 2
1614 2644 2 ! Trailing zero item.
1615 2645 2 !
1616 2646 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 0;
1617 2647 2 SMBITM[SMBMSG$W_ITEM_CODE] = 0;
1618 2648 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;
1619 2649 2
1620 2650 2
1621 2651 2 ! Send the message to the symbiont.
1622 2652 2 !
1623 2653 2 SMBMSG_DESC[1] = SMBMSG;
1624 2654 2 SMBMSG_DESC[0] = .SMBITM - .SMBMSG_DESC[1];
```

```

: 1625      2655 2 SEND_SYMBIONT_MESSAGE(.SMQ, SMBMSG_DESC);
: 1626      2656 2
: 1627      2657 2
: 1628      2658 2 ! Update SMQ.
: 1629      2659 2 !
: 1630      2660 2 SMQ[SMQ$V_STOPPING] = TRUE;
: 1631      2661 2 SMQ[SMQ$V_STOPPED] = TRUE;
: 1632      2662 1 END;

```

			0004 00000	.ENTRY	STOP SYMBIONT_STREAM, Save R2	: 2598
	5E	FC00	CE 9E 00002	MOVAB	-1024(SP), SP-	: 2638
	6E		06 B0 00007	MOVW	#6, SMBMSG	: 2639
02	AE		01 90 0000A	MOVB	#1, SMBMSG+2	: 2640
	52	08	AC D0 0000E	MOVL	SMQ, R2	: 2641
03	AE	0117	C2 90 00012	MOVB	279(R2), SMBMSG+3	: 2646
	50	04	AE 9E 00018	MOVAB	SMBMSG+4, SMBITM	: 2653
			80 D4 0001C	CLRL	(SMBITM)+	: 2654
			5E DD 0001E	PUSHL	SP	: 2655
7E			6E C3 00020	SUBL3	SMBMSG_DESC+4, SMBITM, SMBMSG_DESC	: 2661
		4004	8F BB 00024	PUSHR	#^M<R2,SP>	: 2662
	F767	CF	02 FB 00028	CALLS	#2, SEND_SYMBIONT_MESSAGE	
	11	A2	06 88 0002D	BISB2	#6, 17(R2)	
			04 00031	RET		

: Routine Size: 50 bytes, Routine Base: CODE + 0942

```
: 1634 2663 1 GLOBAL ROUTINE RESET_SYMBIONT_STREAM(SMQ_N,SMQ): NOVALUE=  
: 1635 2664 1  
: 1636 2665 1 |++  
: 1637 2666 1 |  
: 1638 2667 1 | FUNCTIONAL DESCRIPTION:  
: 1639 2668 1 | This routine sends the 'reset stream' message to a symbiont.  
: 1640 2669 1 |  
: 1641 2670 1 | INPUT PARAMETERS:  
: 1642 2671 1 | SMQ_N - Record number of SMQ.  
: 1643 2672 1 | SMQ - Pointer to SMQ.  
: 1644 2673 1 |  
: 1645 2674 1 | IMPLICIT INPUTS:  
: 1646 2675 1 | NONE  
: 1647 2676 1 |  
: 1648 2677 1 | OUTPUT PARAMETERS:  
: 1649 2678 1 | NONE  
: 1650 2679 1 |  
: 1651 2680 1 | IMPLICIT OUTPUTS:  
: 1652 2681 1 | NONE  
: 1653 2682 1 |  
: 1654 2683 1 | ROUTINE VALUE:  
: 1655 2684 1 | NONE  
: 1656 2685 1 |  
: 1657 2686 1 | SIDE EFFECTS:  
: 1658 2687 1 | NONE  
: 1659 2688 1 |  
: 1660 2689 1 | --  
: 1661 2690 1 |  
: 1662 2691 2 BEGIN  
: 1663 2692 2 MAP  
: 1664 2693 2 SMQ: REF BBLOCK; ! Pointer to SMQ  
: 1665 2694 2 LOCAL  
: 1666 2695 2 SCT: REF BBLOCK, ! Pointer to SCT  
: 1667 2696 2 SMBMSG: BBLOCK[JBC$K_SMBMBXSIZ], ! Message buffer  
: 1668 2697 2 SMBITM: REF BBLOCK, ! Cursor for message items  
: 1669 2698 2 SMBMSG_DESC: VECTOR[2]; ! Descriptor for message buffer  
: 1670 2699 2  
: 1671 2700 2  
: 1672 2701 2 ! Message header.  
: 1673 2702 2 |  
: 1674 2703 2 SMBMSG[SMBMSG$W_REQUEST_CODE] = SMBMSG$K_RESET_STREAM;  
: 1675 2704 2 SMBMSG[SMBMSG$B_STRUCTURE_LEVEL] = SMBMSG$K_STRUCTURE_LEVEL;  
: 1676 2705 2 SMBMSG[SMBMSG$B_STREAM_INDEX] = .SMQ[SMQ$B_STREAM_INDEX];  
: 1677 2706 2 SMBITM = SMBMSG + SMBMSG$S_REQUEST_HEADER;  
: 1678 2707 2  
: 1679 2708 2  
: 1680 2709 2 ! Trailing zero item.  
: 1681 2710 2 |  
: 1682 2711 2 SMBITM[SMBMSG$W_ITEM_SIZE] = 0;  
: 1683 2712 2 SMBITM[SMBMSG$W_ITEM_CODE] = 0;  
: 1684 2713 2 SMBITM = .SMBITM + SMBMSG$S_ITEM_HEADER;  
: 1685 2714 2  
: 1686 2715 2  
: 1687 2716 2 ! Send the message to the symbiont.  
: 1688 2717 2 |  
: 1689 2718 2 SMBMSG_DESC[1] = SMBMSG;  
: 1690 2719 2 SMBMSG_DESC[0] = .SMBITM - .SMBMSG_DESC[1];
```

```

: 1691      2720 2 SEND_SYMBIONT_MESSAGE(.SMQ, SMBMSG_DESC);
: 1692      2721 2
: 1693      2722 2
: 1694      2723 2 ! Update SCT.
: 1695      2724 2 !
: 1696      2725 2 SCT = .SMQ[SMQ$L_STREAM SCT];
: 1697      2726 2 BITVECTOR[SCT[SCT_L_RESETTING], .SMQ[SMQ$B_STREAM_INDEX]] = TRUE;
: 1698      2727 2 VECTOR[SCT[SCT_L_QUEUES], .SMQ[SMQ$B_STREAM_INDEX]] = 0;
: 1699      2728 1 END;

```

```

                                0004 00000      .ENTRY RESET_SYMBIONT_STREAM, Save R2      : 2663
                                SE FC00 CE 9E 00002      MOVAB -1024(SP), SP      :
                                6E 02 B0 00007      MOVW #2, SMBMSG      : 2703
                                02 AE 01 90 0000A      MOVB #1, SMBMSG+2      : 2704
                                52 08 AC D0 0000E      MOVL SMQ, R2      : 2705
                                03 AE 0117 C2 90 00012      MOVB 279(R2), SMBMSG+3
                                50 04 AE 9E 00018      MOVAB SMBMSG+4, SMBITM
                                80 D4 0001C      CLRL (SMBITM)+
                                5E DD 0001E      PUSHL SP
                                7E 50 6E C3 00020      SUBL3 SMBMSG_DESC+4, SMBITM, SMBMSG_DESC
                                4004 8F BB 00024      PUSHR #^M<R2,SP>
                                F735 CF 02 FB 00028      CALLS #2, SEND_SYMBIONT_MESSAGE
                                51 00FC C2 D0 0002D      MOVL 252(R2), -SCT
                                50 0117 C2 9A 00032      MOVZBL 279(R2), R0
                                00 10 A1 50 E2 00037      BBSS R0, 16(SCT), 1$
                                3C A140 D4 0003C 1$      CLRL 60(SCT)[R0]
                                04 00040      RET      : 2727
                                : 2728

```

; Routine Size: 65 bytes, Routine Base: CODE + 0974


```

: 1701 2729 1 ROUTINE PROCESS_SYMBIONT_MESSAGE(SMQ_N,SMQ,SCT): NOVALUE=
: 1702 2730 1
: 1703 2731 1 !**
: 1704 2732 1
: 1705 2733 1 FUNCTIONAL DESCRIPTION:
: 1706 2734 1 This routine processes a symbiont response message.
: 1707 2735 1
: 1708 2736 1 INPUT PARAMETERS:
: 1709 2737 1 SMQ_N - Record number of SMQ.
: 1710 2738 1 SMQ - Pointer to SMQ.
: 1711 2739 1 SCT - Pointer to SCT.
: 1712 2740 1
: 1713 2741 1 IMPLICIT INPUTS:
: 1714 2742 1 MBX - Pointer to buffered mailbox message.
: 1715 2743 1
: 1716 2744 1 OUTPUT PARAMETERS:
: 1717 2745 1 NONE
: 1718 2746 1
: 1719 2747 1 IMPLICIT OUTPUTS:
: 1720 2748 1 NONE
: 1721 2749 1
: 1722 2750 1 ROUTINE VALUE:
: 1723 2751 1 NONE
: 1724 2752 1
: 1725 2753 1 SIDE EFFECTS:
: 1726 2754 1 NONE
: 1727 2755 1
: 1728 2756 1 --
: 1729 2757 1
: 1730 2758 2 BEGIN
: 1731 2759 2 MAP
: 1732 2760 2 SMQ: REF BBLOCK, ! Pointer to SMQ
: 1733 2761 2 SCT: REF BBLOCK; ! Pointer to SCT
: 1734 2762 2 LOCAL
: 1735 2763 2 SMBITM: REF BBLOCK, ! Cursor for syrbiomt message
: 1736 2764 2 REQUEST_RESPONSE, ! Symbiont request response
: 1737 2765 2 CONDITION_VECTOR: VECTOR[3], ! Status of current request
: 1738 2766 2 SRQ_TYPE, ! SRQ type to be completed
: 1739 2767 2 SJH_N, ! Record number of SJH
: 1740 2768 2 SJH: REF BBLOCK; ! Pointer to SJH
: 1741 2769 2
: 1742 2770 2
: 1743 2771 2 SMBITM = .MBX + SMBMSG$$ REQUEST_HEADER;
: 1744 2772 2 REQUEST_RESPONSE = SMBMSG$K TASK_STATUS;
: 1745 2773 2 CONDITION_VECTOR[0] = JBC$_NORMAL;
: 1746 2774 2 CONDITION_VECTOR[1] = 0;
: 1747 2775 2 CONDITION_VECTOR[2] = 0;
: 1748 2776 2
: 1749 2777 2
: 1750 2778 2 ! Read the current job record, if any.
: 1751 2779 2
: 1752 2780 2 SJH_N = .SMQ[SMQ$L CURRENT_LIST];
: 1753 2781 2 IF .SJH_N NEQ 0 THEN SJH = READ_RECORD(.SJH_N);
: 1754 2782 2
: 1755 2783 2
: 1756 2784 2 ! Process the message's item list.
: 1757 2785 2

```

```
1758 2786 2 WHILE .SMBITM LSSA .MBX_END DO
1759 2787 3 BEGIN
1760 2788 3 LOCAL
1761 2789 3 ITEM_CODE, : Code of current item
1762 2790 3 ITEM_SIZE; : Size of current item
1763 2791 3
1764 2792 3
1765 2793 3 ! Get the size and item code of the current item.
1766 2794 3 !
1767 2795 3 ITEM_SIZE = .SMBITM[SMBMSG$W_ITEM_SIZE];
1768 2796 3 ITEM_CODE = .SMBITM[SMBMSG$W_ITEM_CODE];
1769 2797 3 SMBITM = .SMBITM + SMBMSG$S_ITEM_READER;
1770 2798 3
1771 2799 3
1772 2800 3 ! Process the item.
1773 2801 3 !
1774 2802 3 CASE .ITEM_CODE FROM 0 TO SMBMSG$K_USER_NAME OF
1775 2803 3 SET
1776 2804 3
1777 2805 3
1778 2806 3 [INRANGE, OVRANGE]:
1779 2807 3 CONDITION_VECTOR[0] = JBC$_INVMSG OR STS$K_ERROR;
1780 2808 3
1781 2809 3
1782 2810 3 [0]:
1783 2811 3 EXIT LOOP;
1784 2812 3
1785 2813 3
1786 2814 3 [SMBMSG$K_ACCOUNTING_DATA]:
1787 2815 4 BEGIN
1788 2816 4 IF .ITEM_SIZE EQL SMBMSG$S_ACCOUNTING_DATA
1789 2817 4 THEN
1790 2818 5 BEGIN
1791 2819 5 SMQ[SMQ$S_ACM_GETCNT] =
1792 2820 5 .SMQ[SMQ$S_ACM_GETCNT] + .SMBITM[SMBMSG$L_RMS_GETS];
1793 2821 5 SMQ[SMQ$S_ACM_QIOCNT] =
1794 2822 5 .SMQ[SMQ$S_ACM_QIOCNT] + .SMBITM[SMBMSG$L_QIO_PUTS];
1795 2823 5 SMQ[SMQ$S_ACM_PAGECNT] =
1796 2824 5 .SMQ[SMQ$S_ACM_PAGECNT] + .SMBITM[SMBMSG$L_PAGES_PRINTED];
1797 2825 5 SMQ[SMQ$S_ACM_SYMCPUTIM] =
1798 2826 5 .SMQ[SMQ$S_ACM_SYMCPUTIM] + .SMBITM[SMBMSG$L_CPU_TIME];
1799 2827 4 END;
1800 2828 3 END;
1801 2829 3
1802 2830 3
1803 2831 3 [SMBMSG$K_CHECKPOINT_DATA]:
1804 2832 4 BEGIN
1805 2833 4 LOCAL
1806 2834 4 SAVED_CHECKPOINT: BBLOCK[SJH$S_CHECKPOINT];
1807 2835 4
1808 2836 4 IF .SJH_N NEQ 0
1809 2837 4 THEN
1810 2838 5 BEGIN
1811 2839 5 CHSMOVE(
1812 2840 5 SJH$S_CHECKPOINT,
1813 2841 5 SJH[SJH$T_CHECKPOINT],
1814 2842 5 SAVED_CHECKPOINT);
```

```

1815 2843 5      CH$FILL(0, SJH$$_CHECKPOINT, SJH[SJH$T_CHECKPOINT]);
1816 2844 5
1817 2845 5      IF STORE_VARIABLE_DATA(
1818 2846 5          .SJH
1819 2847 5          SJH$$_CHECKPOINT,
1820 2848 5          SJH[SJH$T_CHECKPOINT],
1821 2849 5          SYM$K_CHECKPOINT,
1822 2850 5          .ITEM_SIZE,
1823 2851 5          .SMBITM)
1824 2852 5      THEN
1825 2853 5          DEALLOCATE_VARIABLE_DATA(
1826 2854 5              SJH$$_CHECKPOINT,
1827 2855 5              SAVED_CHECKPOINT)
1828 2856 5      ELSE
1829 2857 5          CH$MOVE(
1830 2858 5              SJH$$_CHECKPOINT,
1831 2859 5              SAVED_CHECKPOINT,
1832 2860 5              SJH[SJH$T_CHECKPOINT]);
1833 2861 4      END;
1834 2862 3      END;
1835 2863 3
1836 2864 3
1837 2865 3      [SMBMSG$K_CONDITION_VECTOR]:
1838 2866 4      BEGIN
1839 2867 4          CH$COPY(
1840 2868 4              .ITEM_SIZE, .SMBITM,
1841 2869 4              0,
1842 2870 4              %ALLOCATION(CONDITION_VECTOR), CONDITION_VECTOR);
1843 2871 3      END;
1844 2872 3
1845 2873 3
1846 2874 3      [SMBMSG$K_DEVICE_STATUS]:
1847 2875 4      BEGIN
1848 2876 4          IF .ITEM_SIZE EQL SMBMSG$$_DEVICE_STATUS
1849 2877 4          THEN
1850 2878 5              BEGIN
1851 2879 5                  SMQ[SMQ$V_LOWERCASE] = FALSE;
1852 2880 5                  SMQ[SMQ$V_REMOTE] = FALSE;
1853 2881 5                  SMQ[SMQ$V_SERVER] = FALSE;
1854 2882 5                  SMQ[SMQ$V_STALLED] = FALSE;
1855 2883 5                  SMQ[SMQ$V_TERMINAL] = FALSE;
1856 2884 5                  SMQ[SMQ$V_UNAVAILABLE] = FALSE;
1857 2885 5                  IF .SMBITM[SMBMSG$V_LOWERCASE]
1858 2886 5                      THEN SMQ[SMQ$V_LOWERCASE] = TRUE;
1859 2887 5                  IF .SMBITM[SMBMSG$V_PAUSE_TASK]
1860 2888 5                      THEN SMQ[SMQ$V_PAUSED] = TRUE;
1861 2889 5                  IF .SMBITM[SMBMSG$V_REMOTE]
1862 2890 5                      THEN SMQ[SMQ$V_REMOTE] = TRUE;
1863 2891 5                  IF .SMBITM[SMBMSG$V_SERVER]
1864 2892 5                      THEN SMQ[SMQ$V_SERVER] = TRUE;
1865 2893 5                  IF .SMBITM[SMBMSG$V_STALLED]
1866 2894 5                      THEN SMQ[SMQ$V_STALLED] = TRUE;
1867 2895 5                  IF .SMBITM[SMBMSG$V_STOP_STREAM]
1868 2896 5                      THEN SMQ[SMQ$V_STOPPED] = TRUE;
1869 2897 5                  IF .SMBITM[SMBMSG$V_TERMINAL]
1870 2898 5                      THEN SMQ[SMQ$V_TERMINAL] = TRUE;
1871 2899 5                  IF .SMBITM[SMBMSG$V_UNAVAILABLE]

```

```

: 1872
: 1873
: 1874
: 1875
: 1876
: 1877
: 1878
: 1879
: 1880
: 1881
: 1882
: 1883
: 1884
: 1885
: 1886
: 1887
: 1888
: 1889
: 1890
: 1891
: 1892
: 1893
: 1894
: 1895
: 1896
: 1897
: 1898
: 1899
: 1900
: 1901
: 1902
: 1903
: 1904
: 1905
: 1906
: 1907
: 1908
: 1909
: 1910
: 1911
: 1912
: 1913
: 1914
: 1915
: 1916
: 1917
: 1918
: 1919
: 1920
: 1921
: 1922
: 1923
: 1924
: 1925
: 1926
: 1927
: 1928

```

```

                THEN SMQ[SMQ$V_UNAVAILABLE] = TRUE;
            END;
        END;
[SMBMSG$K_MAXIMUM_STREAMS]:
    BEGIN
    IF .ITEM_SIZE EQL 4
    THEN
        SCT[SCT_B_MAXSTREAMS] = ..SMBITM;
    END;
[SMBMSG$K_REFUSE_REASON]:
    BEGIN
    LOCAL
        SAVED_REFUSAL_REASON:  BBLOCK[SJH$S_REFUSAL_REASON];

    IF .SJH_N NEQ 0
    THEN
        BEGIN
        CH$MOVE(
            SJH$S_REFUSAL_REASON,
            SJH[SJH$T_REFUSAL_REASON],
            SAVED_REFUSAL_REASON);
        CH$FILL(0, SJH$S_REFUSAL_REASON, SJH[SJH$T_REFUSAL_REASON]);

        IF STORE_VARIABLE_DATA(
            .SJH,
            SJH$S_REFUSAL_REASON,
            SJH[SJH$T_REFUSAL_REASON],
            SYM$K_REFUSAL_REASON,
            .ITEM_SIZE,
            .SMBITM)
        THEN
            DEALLOCATE_VARIABLE_DATA(
                SJH$S_REFUSAL_REASON,
                SAVED_REFUSAL_REASON)
        ELSE
            CH$MOVE(
                SJH$S_REFUSAL_REASON,
                SAVED_REFUSAL_REASON,
                SJH[SJH$T_REFUSAL_REASON]);

            SJH[SJH$V_REFUSED] = TRUE;
        END;
    END;
[SMBMSG$K_REQUEST_RESPONSE]:
    BEGIN
    IF .ITEM_SIZE EQL 4
    THEN
        IF ..SMBITM GEQU SMBMSG$K_PAUSE_TASK
        AND ..SMBITM LEQU SMBMSG$K_TASK_STATUS
        THEN
            REQUEST_RESPONSE = ..SMBITM;

```

```

: 1929      2957      3      END;
: 1930      2958      3
: 1931      2959      3
: 1932      2960      3      TES;
: 1933      2961      3
: 1934      2962      3
: 1935      2963      3      SMBITM = .SMBITM + .ITEM_SIZE;
: 1936      2964      3      END;
: 1937      2965      3
: 1938      2966      3
: 1939      2967      3      ! Update state based on the request status.
: 1940      2968      3      !
: 1941      2969      3      SRQ_TYPE = 0;
: 1942      2970      3      CASE .REQUEST_RESPONSE FROM SMBMSG$K_PAUSE_TASK TO SMBMSG$K_TASK_STATUS OF
: 1943      2971      3      SET
: 1944      2972      3
: 1945      2973      3
: 1946      2974      3      [SMBMSG$K_PAUSE_TASK]:
: 1947      2975      3      BEGIN
: 1948      2976      3      IF .CONDITION_VECTOR[0]
: 1949      2977      3      THEN
: 1950      2978      3      SMQ[SMQ$V_PAUSED] = TRUE;
: 1951      2979      3      SMQ[SMQ$V_PAUSING] = FALSE;
: 1952      2980      3      END;
: 1953      2981      3
: 1954      2982      3
: 1955      2983      3      [SMBMSG$K_RESET_STREAM]:
: 1956      2984      3      0;
: 1957      2985      3
: 1958      2986      3
: 1959      2987      3      [SMBMSG$K_RESUME_TASK]:
: 1960      2988      3      BEGIN
: 1961      2989      3      IF .CONDITION_VECTOR[0]
: 1962      2990      3      THEN
: 1963      2991      3      BEGIN
: 1964      2992      3      SMQ[SMQ$V_OPERATOR_REQUEST] = FALSE;
: 1965      2993      3      SMQ[SMQ$V_PAUSED] = FALSE;
: 1966      2994      3      IF .SMQ[SMQ$V_ALIGNING] THEN SMQ[SMQ$V_PAUSED] = TRUE;
: 1967      2995      3      END;
: 1968      2996      3      SMQ[SMQ$V_ALIGNING] = FALSE;
: 1969      2997      3      SMQ[SMQ$V_RESUMING] = FALSE;
: 1970      2998      3      END;
: 1971      2999      3
: 1972      3000      3
: 1973      3001      3      [SMBMSG$K_START_STREAM]:
: 1974      3002      3      BEGIN
: 1975      3003      3      SRQ_TYPE = SRQ$K_START_SYMBIONT;
: 1976      3004      3      SMQ[SMQ$V_STARTING] = FALSE;
: 1977      3005      3      IF NOT .CONDITION_VECTOR[0]
: 1978      3006      3      THEN
: 1979      3007      3      BEGIN
: 1980      3008      3      IF .SMQ[SMQ$B_STREAM_INDEX] GTRU .SCT[SCT_B_MAXSTREAMS]
: 1981      3009      3      THEN
: 1982      3010      3      BEGIN
: 1983      3011      3      BITVECTOR[SCT[SCT_L_BITMAP], .SMQ[SMQ$B_STREAM_INDEX]] = FALSE;
: 1984      3012      3      VECTOR[SCT[SCT_L_QUEUES], .SMQ[SMQ$B_STREAM_INDEX]] = 0;
: 1985      3013      3      CONDITION_VECTOR[0] = START_SYMBIONT_STREAM(.SMQ_N, .SMQ);

```

```

: 1986 3014 5          IF .CONDITION_VECTOR[0] THEN RETURN;
: 1987 3015 5          END
: 1988 3016 4          ELSE
: 1989 3017 4          SMQ[SMQ$V_STOPPED] = TRUE;
: 1990 3018 3          END;
: 1991 3019 2          END;
: 1992 3020 2
: 1993 3021 2
: 1994 3022 2          [SMBMSG$K_START_TASK]:
: 1995 3023 2          BEGIN
: 1996 3024 2          IF .SJH_N NEQ 0
: 1997 3025 2          THEN
: 1998 3026 2          SJH[SJH$V_FILE_STARTING] = FALSE;
: 1999 3027 2
: 2000 3028 2          IF NOT .CONDITION_VECTOR[0]
: 2001 3029 2          OR .SJH[SJH$V_REFUSED]
: 2002 3030 2          THEN
: 2003 3031 2          REQUEST_RESPONSE = SMBMSG$K_TASK_COMPLETE
: 2004 3032 3          ELSE
: 2005 3033 4          BEGIN
: 2006 3034 4          IF .SMQ[SMQ$V_OPERATOR_REQUEST]
: 2007 3035 4          THEN
: 2008 3036 5          BEGIN
: 2009 3037 5          SMQ[SMQ$V_PAUSED] = FALSE;      ! Temporarily cleared (V03-015)
: 2010 3038 5          SMQ[SMQ$V_OPERATOR_REQUEST] = FALSE;  ! Temp. added (V03-015)
: 2011 3039 5          IF .SJH_N NEQ 0 THEN OPERATOR_REQUEST(.SMQ, .SJH);
: 2012 3040 4          END;
: 2013 3041 3          END;
: 2014 3042 2          END;
: 2015 3043 2
: 2016 3044 2
: 2017 3045 2          [SMBMSG$K_STOP_STREAM]:
: 2018 3046 2          BEGIN
: 2019 3047 2          BITVECTOR[SCT[SCT_L_BITMAP], .SMQ[SMQ$B_STREAM_INDEX]] = FALSE;
: 2020 3048 2          VECTOR[SCT[SCT_L_QUEUES], .SMQ[SMQ$B_STREAM_INDEX]] = 0;
: 2021 3049 2          IF .SCT[SCT_L_BITMAP] EQL 0 THEN SCT[SCT_V_DELETING] = TRUE;
: 2022 3050 2          SMQ[SMQ$L_STREAM_SCT] = 0;
: 2023 3051 2          SMQ[SMQ$B_STREAM_INDEX] = 0;
: 2024 3052 2          SMQ[SMQ$V_PAUSED] = FALSE;
: 2025 3053 2          SMQ[SMQ$V_STALLED] = FALSE;
: 2026 3054 2          SMQ[SMQ$V_STOPPING] = FALSE;
: 2027 3055 2          END;
: 2028 3056 2
: 2029 3057 2
: 2030 3058 2          [SMBMSG$K_STOP_TASK, SMBMSG$K_TASK_COMPLETE]:
: 2031 3059 2          BEGIN
: 2032 3060 2          IF .SMQ[SMQ$V_PAUSING] THEN SMQ[SMQ$V_PAUSED] = TRUE;
: 2033 3061 2          SMQ[SMQ$V_ALIGNING] = FALSE;
: 2034 3062 2          SMQ[SMQ$V_OPERATOR_REQUEST] = FALSE;
: 2035 3063 2          SMQ[SMQ$V_PAUSING] = FALSE;
: 2036 3064 2          SMQ[SMQ$V_RESUMING] = FALSE;
: 2037 3065 2          END;
: 2038 3066 2
: 2039 3067 2
: 2040 3068 2          [SMBMSG$K_TASK_STATUS]:
: 2041 3069 2          0;
: 2042 3070 2

```

```

2043 3071 2
2044 3072 2      TES;
2045 3073 2
2046 3074 2
2047 3075 2      ! If an incomplete service has completed, notify the requestor.
2048 3076 2
2049 3077 2      IF .SRQ_TYPE NEQ 0
2050 3078 2      THEN
2051 3079 2          SCAN INCOMPLETE_SERVICES(
2052 3080 2              .SRV_K_SYMBIONT,
2053 3081 2              .SMQ_N, .SMQ,
2054 3082 2              .SRQ_TYPE,
2055 3083 2              .CONDITION_VECTOR[0]);
2056 3084 2
2057 3085 2
2058 3086 2      ! If the stream is not available for new work, we are done.
2059 3087 2
2060 3088 2      IF NOT ONEOF (.REQUEST_RESPONSE,
P      3089 2          BMSK (
P      3090 2              SMBMSG$K_START_STREAM,
P      3091 2              SMBMSG$K_STOP_TASK,
2064 3092 2              SMBMSG$K_TASK_COMPLETE))
2065 3093 2      THEN
2066 3094 2          BEGIN
2067 3095 2              IF .SJH_N NEQ 0 THEN REWRITE_RECORD(.SJH_N);
2068 3096 2              RETURN;
2069 3097 2          END;
2070 3098 2
2071 3099 2
2072 3100 2      ! Handle multi-copy and multi-file situations.
2073 3101 2
2074 3102 2      IF .SJH_N NEQ 0
2075 3103 2      THEN
2076 3104 2          BEGIN
2077 3105 2
2078 3106 2              ! Update the job status with the received status.
2079 3107 2
2080 3108 2              IF .SJH[SJH$L_CONDITION_1] EQL 0
2081 3109 2              OR (.SJH[SJH$C_CONDITION_1] AND NOT .CONDITION_VECTOR[0])
2082 3110 2              THEN
2083 3111 2                  CH$MOVE(
2084 3112 2                      SJH$$CONDITION_VECTOR,
2085 3113 2                      CONDITION_VECTOR,
2086 3114 2                      SJH[SJH$L_CONDITION_1]);
2087 3115 2
2088 3116 2
2089 3117 2              IF .SJH[SJH$V_REFUSED]
2090 3118 2              THEN
2091 3119 2                  BEGIN
2092 3120 2                      UPDATE GETQUI_DATA(.SJH_N, .SJH);
2093 3121 2                      ENQUEUE_JOB(.SJH_N, .SJR);
2094 3122 2                      SMQ[SMQ$L_CURRENT_LIST] = 0;
2095 3123 2                      SMQ[SMQ$L_CURRENT_LIST_END] = 0;
2096 3124 2                      SMQ[SMQ$B_CURRENT_JOB_COUNT] = 0;
2097 3125 2                  END
2098 3126 2
2099 3127 2

```

```

2100 3128 3 ELSE IF .SJH[SJH$V_ABORTED]
2101 3129 3 THEN
2102 3130 4 BEGIN
2103 3131 4 UPDATE GETQUI_DATA(.SJH_N, .SJH);
2104 3132 4 COMPLETE_JOB(.SJH_N, .SJH, .SMQ, 0);
2105 3133 4 SJH_N = 0;
2106 3134 4 SMQ[SMQ$SL_CURRENT_LIST] = 0;
2107 3135 4 SMQ[SMQ$SL_CURRENT_LIST_END] = 0;
2108 3136 4 SMQ[SMQ$B_CURRENT_JOB_COUNT] = 0;
2109 3137 4 END
2110 3138 4
2111 3139 4
2112 3140 3 ELSE
2113 3141 4 BEGIN
2114 3142 4 LOCAL
2115 3143 4     SQR_N,           ! Record number of SQR
2116 3144 4     SQR:             REF BBLOCK;    ! Pointer to SQR
2117 3145 4
2118 3146 4
2119 3147 4     SQR = READ_RECORD(SQR_N = .SJH[SJH$SL_CURRENT_FILE_LINK]);
2120 3148 4
2121 3149 4
2122 3150 4     SJH[SJH$SL_COMPLETED_BLOCKS] =
2123 3151 4     .SJH[SJH$SL_COMPLETED_BLOCKS] + .SQR[SQR$SL_FILE_SIZE];
2124 3152 4     SJH[SJH$SL_CURRENT_FILE_CHKPT] = 0;
2125 3153 4     SJH[SJH$B_JOB_COPIES_CHKPT] = 0;
2126 3154 4     SJH[SJH$B_FILE_COPIES_CHKPT] = 0;
2127 3155 4     DEALLOCATE_VARIABLE_DATA(
2128 3156 4     SJH$S_CHECKPOINT,
2129 3157 4     SJH[SJH$T_CHECKPOINT]);
2130 3158 4
2131 3159 4
2132 3160 4     SJH[SJH$B_FILE_COPIES_DONE] = .SJH[SJH$B_FILE_COPIES_DONE] + 1;
2133 3161 4     IF .SJH[SJH$B_FILE_COPIES_DONE] GEQU .SQR[SQR$B_FILE_COPIES]
2134 3162 4     THEN
2135 3163 5     BEGIN
2136 3164 5     IF .SQR[SYM$SL_LINK] EQL 0
2137 3165 5     THEN
2138 3166 6     BEGIN
2139 3167 6     SJH[SJH$B_JOB_COPIES_DONE] = .SJH[SJH$B_JOB_COPIES_DONE] + 1;
2140 3168 6     IF .SJH[SJH$B_JOB_COPIES_DONE] GEQU .SJR[SJR$B_JOB_COPIES]
2141 3169 6     THEN
2142 3170 7     BEGIN
2143 3171 7     RELEASE_RECORD(.SQR_N);
2144 3172 7     UPDATE GETQUI_DATA(.SJH_N, .SJH);
2145 3173 7     COMPLETE_JOB(.SJH_N, .SJH, .SMQ, 0);
2146 3174 7     SJH_N = 0;
2147 3175 7     SMQ[SMQ$SL_CURRENT_LIST] = 0;
2148 3176 7     SMQ[SMQ$SL_CURRENT_LIST_END] = 0;
2149 3177 7     SMQ[SMQ$B_CURRENT_JOB_COUNT] = 0;
2150 3178 7     END
2151 3179 6     ELSE
2152 3180 7     BEGIN
2153 3181 7     LOCAL
2154 3182 7     SQR_N2,           ! Record number of SQR
2155 3183 7     SQR_2:           REF BBLOCK;    ! Pointer to SQR
2156 3184 7

```



```

: 2157      3185 7      SQR_2 = READ RECORD(SQR_N2 = .SJH[SJH$L_FILE_LIST]);
: 2158      3186 7      SJH[SJH$B_FILE_COPIES_DONE] = 0;
: 2159      3187 7      START SYMBIONT_TASK(
: 2160      3188 7      .SMQ_N, .SMQ,
: 2161      3189 7      .SJH_N, .SJH,
: 2162      3190 7      .SQR_N2, .SQR_2);
: 2163      3191 7      END
: 2164      3192 6      END
: 2165      3193 5      ELSE
: 2166      3194 6      BEGIN
: 2167      3195 6      LOCAL
: 2168      3196 6      SQR_N2,
: 2169      3197 6      SQR_2: REF BBLOCK; ! Record number of SQR
: 2170      3198 6      ! Pointer to SQR
: 2171      3199 6      SQR_2 = READ RECORD(SQR_N2 = .SQR[SYM$L_LINK]);
: 2172      3200 6      SJH[SJH$B_FILE_COPIES_DONE] = 0;
: 2173      3201 6      START SYMBIONT_TASK(
: 2174      3202 6      .SMQ_N, .SMQ,
: 2175      3203 6      .SJH_N, .SJH,
: 2176      3204 6      .SQR_N2, .SQR_2);
: 2177      3205 6      END
: 2178      3206 5      END
: 2179      3207 4      ELSE
: 2180      3208 5      BEGIN
: 2181      3209 5      START SYMBIONT_TASK(
: 2182      3210 5      .SMQ_N, .SMQ,
: 2183      3211 5      .SJH_N, .SJH,
: 2184      3212 5      .SQR_N, .SQR);
: 2185      3213 5      END
: 2186      3214 3      END;
: 2187      3215 2      END;
: 2188      3216 2
: 2189      3217 2
: 2190      3218 2      ! Rewrite the job header, if any.
: 2191      3219 2
: 2192      3220 2      IF .SJH_N NEQ 0 THEN REWRITE_RECORD(.SJH_N);
: 2193      3221 2
: 2194      3222 2
: 2195      3223 2      ! Find the next work item for the symbiont.
: 2196      3224 2
: 2197      3225 2      IF .SMQ[SMQ$B_CURRENT_JOB_COUNT] EQL 0
: 2198      3226 2      THEN
: 2199      3227 2      IF .SMQ[SMQ$V_STOPPED]
: 2200      3228 2      THEN
: 2201      3229 2      STOP_SYMBIONT_STREAM(.SMQ_N, .SMQ)
: 2202      3230 2      ELSE
: 2203      3231 2      FIND_PENDING_JOBS(.SMQ_N, .SMQ);
: 2204      3232 1      END;

```

OFFC 0000 PROCESS_SYMBIONT MESSAGE:

SA 00000000' SE
EF

2C C2 00002
04 C1 00005

.WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
SUBL2 #44, SP
ADDL3 #4, MBX, SMBITM

: 2729
: 2771

		01	A0		08	8A	0014A		BICB2	#8, 1(R0)	2884	
			04		6A	F9	0014E		BLBC	(SMBITM), 12\$	2885	
		02	A1		01	88	00151		BISB2	#1, 2(R1)	2886	
		03	6A		01	E1	00155	12\$:	BBC	#1, (SMBITM), 13\$	2887	
			60		04	88	00159		BISB2	#4, (R0)	2888	
		03	6A		02	E1	0015C	13\$:	BBC	#2, (SMBITM), 14\$	2889	
			50		10	88	00160		BISB2	#16, (R0)	2890	
		04	6A		03	E1	00163	14\$:	BBC	#3, (SMBITM), 15\$	2891	
			6A		10	88	00167		BISB2	#16, 2(R1)	2892	
		04	6A		04	E1	0016B	15\$:	BBC	#4, (SMBITM), 16\$	2893	
			60	80	8F	88	0016F		BISB2	#128, (R0)	2894	
		04	6A		05	E1	00173	16\$:	BBC	#5, (SMBITM), 17\$	2895	
			6A		02	88	00177		BISB2	#2, 1(R0)	2896	
		05	A0		06	E1	0017B	17\$:	BBC	#6, (SMBITM), 18\$	2897	
			6A		8F	88	0017F		BISB2	#64, 2(R1)	2898	
			A1	40	6A	95	00184	18\$:	TSTB	(SMBITM)	2899	
					6A	18	00186		BGEQ	26\$		
			A0		08	88	00188		BISB2	#8, 1(R0)	2900	
					64	11	0018C	19\$:	BRB	26\$	2802	
			04		58	D1	0018E	20\$:	CMPL	ITEM_SIZE, #4	2907	
					5F	12	00191		BNEQ	26\$		
			50	0C	AC	D0	00193		MOVL	SCT, R0	2909	
			A0		6A	90	00197		MOVB	(SMBITM), 5(R0)		
					55	11	0019B	21\$:	BRB	26\$	2802	
			52		6E	E9	0019D	22\$:	BLBC	(SP), 26\$	2918	
			5B	01D2	C6	9E	001A0		MOVAB	466(SJH), R11	2923	
			6B		06	28	001A5		MOVCS	#6, (R11), SAVED_REFUSAL_REASON		
			6E		00	2C	001AA		MOVCS	#0, (SP), #0, #6, (R11)	2925	
					6B		001AF					
				0500	8F	BB	001B0		PUSHR	#*M<R8,R10>	2932	
					15	DD	001B4		PUSHL	#21	2930	
					5B	DD	001B6		PUSHL	R11		
					06	DD	001B8		PUSHL	#6		
					56	DD	001BA		PUSHL	SJH		
			00000000G		06	FB	001BC		CALLS	#6, STORE_VARIABLE_DATA		
			0E		50	E9	001C3		BLBC	R0, 23\$		
				20	AE	9F	001C6		PUSHAB	SAVED_REFUSAL_REASON	2935	
					06	DD	001C9		PUSHL	#6		
			00000000G		02	FB	001CB		CALLS	#2, DEALLOCATE_VARIABLE_DATA		
			6B	20	AE	28	001D4	23\$:	MOVCS	#6, SAVED_REFUSAL_REASON, (R11)	2942	
				10	8F	88	001D9	24\$:	BISB2	#128, 16(SJH)	2944	
					12	11	001DE		BRB	26\$	2802	
					04	58	D1	001E0	25\$:	CMPL	ITEM_SIZE, #4	2951
					0D	12	001E3		BNEQ	26\$		
					6A	D5	001E5		TSTL	(SMBITM)	2953	
					09	13	001E7		BEQL	26\$		
				09	6A	D1	001E9		CMPL	(SMBITM), #9	2954	
					04	1A	001EC		BGTRU	26\$		
			04	AE	6A	D0	001EE		MOVL	(SMBITM), REQUEST_RESPONSE	2956	
				5A	58	C0	001F2	26\$:	ADDL2	ITEM_SIZE, SMBITM	2963	
					FE3E	31	001F5		BRW	1\$	2786	
					52	D4	001F8	27\$:	CLRL	SRQ TYPE	2969	
			08	01	04	AE	CF	001FA	CASEL	REQUEST_RESPONSE, #1, #8	2970	
			0037	0020	00DA	0012	001FF	28\$:	.WORD	29\$-28\$,-		
			00CB	00CB	00A0	0073	00207			46\$-28\$,-		
					00DA	0020F				31\$-28\$,-		

									33\$-28\$,-		
									36\$-28\$,-		
									41\$-28\$,-		
									44\$-28\$,-		
									44\$-28\$,-		
									46\$-28\$		
									CONDITION VECTOR, 30\$	2976	
10	A7		28	AE	E9	00211	29\$:	BLBC		2978	
10	A7			04	88	00215		BISB2	#4, 16(R7)	2979	
				08	8A	00219	30\$:	BICB2	#8, 16(R7)	2970	
				7E	11	0021D		BRB	40\$	2989	
				AE	E9	0021F	31\$:	BLBC	CONDITION VECTOR, 32\$	2993	
10	A7		28	06	8A	00223		BICB2	#6, 16(R7)	2994	
				A7	E9	00227		BLBC	16(R7), 32\$	2997	
10	A7		10	04	88	0022B		BISB2	#4, 16(R7)	2970	
10	A7		41	8F	8A	0022F	32\$:	BICB2	#65, 16(R7)	3003	
				67	11	00234		BRB	40\$	3004	
				OC	D0	00236	33\$:	MOVL	#12, SRQ TYPE	3005	
11	A7			01	8A	00239		BICB2	#1, 17(R7)	3008	
				AE	E8	0023D		BLBS	CONDITION VECTOR, 40\$		
				C7	9A	00241		MOVZBL	279(R7), R1		
				OC	D0	00246		MOVL	SCT, R0		
				A0	91	0024A		CMPB	5(R0), R1		
				1C	1E	0024E		BGEQU	35\$		
00	OC	A0		51	E5	00250		BBCC	R1, 12(R0), 34\$	3011	
				A041	D4	00255	34\$:	CLRL	60(R0)[R1]	3012	
				57	DD	00259		PUSHL	R7	3013	
				AC	DD	0025B		PUSHL	SMQ_N		
FAA8	CF			02	FB	0025E		CALLS	#2, -START SYMBIONT STREAM		
28	AE			50	D0	00263		MOVL	R0, CONDITION VECTOR		
				AE	E9	00267		BLBC	CONDITION_VECTOR, 46\$	3014	
				04	0026B			RET			
				02	88	0026C	35\$:	BISB2	#2, 17(R7)	3017	
				67	11	00270		BRB	46\$	3005	
				6E	E9	00272	36\$:	BLBC	(SP), 37\$	3024	
				10	8A	00275		BICB2	#16, 16(SJH)	3026	
				AE	E9	00279	37\$:	BLBC	CONDITION_VECTOR, 38\$	3028	
				A6	95	0027D		TSTB	16(SJH)	3029	
				06	18	00280		BGEQ	39\$		
				08	D0	00282	38\$:	MOVL	#8, REQUEST_RESPONSE	3031	
				51	11	00286		BRB	46\$		
4C				01	E1	00288	39\$:	BBCC	#1, 16(R7), 46\$	3034	
				06	8A	0028D		BICB2	#6, 16(R7)	3038	
				6E	E9	00291		BLBC	(SP), 46\$	3039	
				56	DD	00294		PUSHL	SJH		
				57	DD	00296		PUSHL	R7		
				02	FB	00298		CALLS	#2, OPERATOR_REQUEST		
				3A	11	0029D	40\$:	BRB	46\$	2970	
				AC	D0	0029F	41\$:	MOVL	SCT, R0	3047	
				C7	9A	002A3		MOVZBL	279(R7), R1		
				51	E5	002A8		BBCC	R1, 12(R0), 42\$		
00	OC	A0		A041	D4	002AD	42\$:	CLRL	60(R0)[R1]	3048	
				A0	D5	002B1		TSTL	12(R0)	3049	
				04	12	002B4		BNEQ	43\$		
				01	88	002B6		BISB2	#1, 4(R0)		
				C7	D4	002BA	43\$:	CLRL	252(R7)	3050	
				C7	94	002BE		CLRB	279(R7)	3051	
				8F	AA	002C2		BICW2	#1156, 16(R7)	3054	

			OF	11	002C8		BRB	46\$	2970
	50		A7	9E	002CA	44\$:	MOVAB	16(R7), R0	3060
03	60		03	E1	002CE		BBC	#3, (R0), 45\$	
	60		04	88	002D2		BISB2	#4, (R0)	
	60		8F	8A	002D5	45\$:	BICB2	#7\$ (R0)	3064
		48	52	D5	002D9	46\$:	TSTL	SRQ_TYPE	3077
			13	13	002DB		BEQL	47\$	
			AE	DD	002DD	28	PUSHL	CONDITION_VECTOR	3083
			52	DD	002E0		PUSHL	SRQ_TYPE	3082
			57	DD	002E2		PUSHL	R7	3081
		04	AC	DD	002E4		PUSHL	SMQ_N	
			02	DD	002E7		PUSHL	#2	3079
00000000G	EF		05	FB	002E9		CALLS	#5, SCAN_INCOMPLETE_SERVICES	
50 09800000	8F		AE	78	002F0	47\$:	ASHL	REQUEST_RESPONSE, #T59383552, R0	3092
			0E	19	002F9		BLSS	49\$	
		01	6E	E8	002FB		BLBS	(SP), 48\$	3095
				04	002FE		RET		
			59	DD	002FF	48\$:	PUSHL	SJH_N	
00000000G	EF		01	FB	00301		CALLS	#1, -REWRITE_RECORD	
				04	00308		RET		3094
		03	6E	E8	00309	49\$:	BLBS	(SP), 50\$	3102
			00DE	31	0030C		BRW	61\$	
		50	C6	D0	0030F	50\$:	MOVL	220(SJH), R0	3108
			07	13	00314		BEQL	51\$	
		08	50	E9	00316		BLBC	R0, 52\$	3109
		07	AE	E8	00319		BLBS	CONDITION_VECTOR, 52\$	
00DC C6 28	AE		0C	28	0031D	51\$:	MOVC3	#12, CONDITION_VECTOR, 220(SJH)	3114
			A6	95	00324	52\$:	TSTB	16(SJH)	3117
			18	18	00327		BGEQ	53\$	
			56	DD	00329		PUSHL	SJH	3120
			59	DD	0032B		PUSHL	SJH_N	
00000000G	EF		02	FB	0032D		CALLS	#2, -UPDATE_GETQUI_DATA	
			56	DD	00334		PUSHL	SJH	3121
			59	DD	00336		PUSHL	SJH_N	
00000000G	EF		02	FB	00338		CALLS	#2, -ENQUEUE_JOB	
			76	11	0033F		BRB	55\$	3122
		10	A6	E8	00341	53\$:	BLBS	16(SJH), 54\$	3128
		00F0	C6	D0	00345		MOVL	240(SJH), SQR_N	3147
			54	DD	0034A		PUSHL	SQR_N	
00000000G	EF		01	FB	0034C		CALLS	#1, -READ_RECORD	
			50	D0	00353		MOVL	R0, SQR	
00DB C6			A2	C0	00356		ADDL2	56(SQR), 216(SJH)	3151
		38	C6	D4	0035C		CLRL	236(SJH)	3152
		00EC	C6	94	00360		CLRB	379(SJH)	3153
		017B	C6	94	00364		CLRB	376(SJH)	3154
		0178	C6	9F	00368		PUSHAB	384(SJH)	3157
		0180	C6	9F	00368		PUSHL	#32	
00000000G	EF		02	FB	0036E		CALLS	#2, DEALLOCATE_VARIABLE_DATA	
			53	C6	9E	00375	MOVAB	377(SJH), R3	3160
			63	96	0037A		INCB	(R3)	
44 A2			63	91	0037C		CMPB	(R3), 68(SQR)	3161
			59	1F	00380		BLSSU	59\$	
			62	D5	00382		TSTL	(SQR)	3164
			41	12	00384		BNEQ	57\$	
		017C	C6	96	00386		INCB	380(SJH)	3167
017A C6		017C	C6	91	0038A		CMPB	380(SJH), 378(SJH)	3168
			2D	1F	00391		BLSSU	56\$	

00000000G	EF		54	DD	00393		PUSHL	SQR_N	3171
			01	FB	00395		CALLS	#1,-RELEASE_RECORD	
			56	DD	0039C	54\$:	PUSHL	SJH	3172
			59	DD	0039E		PUSHL	SJH_N	
00000000G	EF		02	FB	003A0		CALLS	#2,-UPDATE_GETQUI_DATA	
			7E	D4	003A7		CLRL	-(SP)	3173
	7E		56	7D	003A9		MOVQ	SJH,-(SP)	
			59	DD	003AC		PUSHL	SJH_N	
00000000G	EF		04	FB	003AE		CALLS	#4,-COMPLETE_JOB	
			59	D4	003B5		CLRL	SJH_N	3174
		48	A7	7C	003B7	55\$:	CLRL	72(R7)	3175
		0115	C7	94	003BA		CLRB	277(R7)	3177
			2D	11	003BE		BRB	61\$	3158
	52	00F4	C6	D0	003C0	56\$:	MOVL	244(SJH),SQR_N2	3185
			03	11	003C5		BRB	58\$	
	52		62	D0	003C7	57\$:	MOVL	(SQR),SQR_N2	3199
			52	DD	003CA	58\$:	PUSHL	SQR_N2	
00000000G	EF		01	FB	003CC		CALLS	#1,-READ_RECORD	
			63	94	003D3		CLRB	(R3)	3200
			50	DD	003D5		PUSHL	SQR_2	3204
			52	DD	003D7		PUSHL	SQR_N2	
			04	11	003D9		BRB	60\$	3203
			52	DD	003DB	59\$:	PUSHL	SQR	3212
			54	DD	003DD		PUSHL	SQR_N	
			56	DD	003DF	60\$:	PUSHL	SJH	3211
		0280	8F	BB	003E1		PUSHR	#*M<R7,R9>	3210
		04	AC	DD	003E5		PUSHL	SMQ_N	
F378	CF		06	FB	003E8		CALLS	#6,-START_SYMBIONT_TASK	
			59	D5	003ED	61\$:	TSTL	SJH_N	3220
			09	13	003EF		BEQL	62\$	
			59	DD	003F1		PUSHL	SJH_N	
00000000G	EF		01	FB	003F3		CALLS	#1,-REWRITE_RECORD	
		0115	C7	95	003FA	62\$:	TSTB	277(R7)	3225
			1C	12	003FE		BNEQ	64\$	
0B	11	A7	01	E1	00400		BBC	#1,17(R7),63\$	3227
			57	DD	00405		PUSHL	R7	3229
		04	AC	DD	00407		PUSHL	SMQ_N	
FB7E	CF		02	FB	0040A		CALLS	#2,-STOP_SYMBIONT_STREAM	
			04	04	0040F		RET		
			57	DD	00410	63\$:	PUSHL	R7	3231
		04	AC	DD	00412		PUSHL	SMQ_N	
00000000G	EF		02	FB	00415		CALLS	#2,-FIND_PENDING_JOBS	
			04	04	0041C	64\$:	RET		3232

; Routine Size: 1053 bytes, Routine Base: CODE + 09B5

```

: 2206 3233 1 GLOBAL ROUTINE SYMBIONT_SERVICE: NOVALUE=
: 2207 3234 1
: 2208 3235 1 :++
: 2209 3236 1
: 2210 3237 1 FUNCTIONAL DESCRIPTION:
: 2211 3238 1 This routine processes the message type:
: 2212 3239 1 MSG$_SMBINI symbiont has completed assignment
: 2213 3240 1
: 2214 3241 1 INPUT PARAMETERS:
: 2215 3242 1 NONE
: 2216 3243 1
: 2217 3244 1 IMPLICIT INPUTS:
: 2218 3245 1 MBX - Pointer to buffered mailbox message.
: 2219 3246 1
: 2220 3247 1 OUTPUT PARAMETERS:
: 2221 3248 1 NONE
: 2222 3249 1
: 2223 3250 1 IMPLICIT OUTPUTS:
: 2224 3251 1 NONE
: 2225 3252 1
: 2226 3253 1 ROUTINE VALUE:
: 2227 3254 1 NONE
: 2228 3255 1
: 2229 3256 1 SIDE EFFECTS:
: 2230 3257 1 NONE
: 2231 3258 1
: 2232 3259 1 --
: 2233 3260 1
: 2234 3261 2 BEGIN
: 2235 3262 2 LOCAL
: 2236 3263 2 SCT: REF BBLOCK; ! Pointer to SCT
: 2237 3264 2
: 2238 3265 2
: 2239 3266 2 ! Validate the message structure level.
: 2240 3267 2
: 2241 3268 2 IF .MBX[SMBMSG$_STRUCTURE_LEVEL] NEQ SMBMSG$_K_STRUCTURE_LEVEL
: 2242 3269 2 OR .MBX[SMBMSG$_STREAM_INDEX] GEQU SCT_K_MAXSTREAMS
: 2243 3270 2 THEN
: 2244 3271 3 BEGIN
: 2245 3272 3 SIGNAL(JBC$_INVMSG OR STS$_K_ERROR);
: 2246 3273 3 RETURN;
: 2247 3274 2 END;
: 2248 3275 2
: 2249 3276 2
: 2250 3277 2 ! Search the symbiont control table for the PID of the process that sent the
: 2251 3278 2 message, which is in the second longword of the IOSB. If found, locate the
: 2252 3279 2 queue corresponding to the stream identifier.
: 2253 3280 2
: 2254 3281 2 SCT = .SYMBIONT CONTROL;
: 2255 3282 2 WHILE .SCT NEQ 0 DO
: 2256 3283 3 BEGIN
: 2257 3284 3 IF .SCT[SCT_L_PID] EQL .MBX[ACM$_L_PROCID]
: 2258 3285 3 THEN
: 2259 3286 4 BEGIN
: 2260 3287 4 LOCAL
: 2261 3288 4 SMQ_N, ! Record number of SMQ
: 2262 3289 4 SMQ: REF BBLOCK; ! Pointer to SMQ

```



```
2263 3290 4
2264 3291 4
2265 3292 4      ! Update SCT for a resetting stream.
2266 3293 4
2267 3294 4      IF .BITVECTOR[SCT[SCT_L_RESETTING], .MBX[SMBMSG$B_STREAM_INDEX]]
2268 3295 4      THEN
2269 3296 5          BEGIN
2270 3297 5              BITVECTOR[SCT[SCT_L_RESETTING], .MBX[SMBMSG$B_STREAM_INDEX]] = FALSE;
2271 3298 5              BITVECTOR[SCT[SCT_L_BITMAP], .MBX[SMBMSG$B_STREAM_INDEX]] = FALSE;
2272 3299 5              IF .SCT[SCT_L_BITMAP] EQL 0 THEN SCT[SCT_V_DELETING] = TRUE;
2273 3300 5              RETURN;
2274 3301 4          END;
2275 3302 4
2276 3303 4
2277 3304 4      ! Get the queue header corresponding to the stream index, and ensure
2278 3305 4      that it is an active stream.
2279 3306 4
2280 3307 4      SMQ_N = .VECTOR[SCT[SCT_L_QUEUES], .MBX[SMBMSG$B_STREAM_INDEX]];
2281 3308 4      IF .SMQ_N NEQ 0
2282 3309 4      THEN
2283 3310 5          BEGIN
2284 3311 5
2285 3312 5              ! Read the queue header.
2286 3313 5
2287 3314 5              LOCK_QUEUE_FILE();
2288 3315 5              SMQ = READ_RECORD(.SMQ_N);
2289 3316 5
2290 3317 5
2291 3318 5              ! Ensure that the record is a queue header that is connected to this
2292 3319 5              stream. If it is, process the message.
2293 3320 5
2294 3321 5              IF .SMQ[SYMS$B_TYPE] EQL SYMS$K_SMQ
2295 3322 5              AND .SMQ[SMQ$[STREAM_SCT]] EQL .SCT
2296 3323 5              AND .SMQ[SMQ$B_STREAM_INDEX] EQL .MBX[SMBMSG$B_STREAM_INDEX]
2297 3324 5              THEN
2298 3325 6                  BEGIN
2299 3326 6                      PROCESS_SYMBIONT_MESSAGE(.SMQ_N, .SMQ, .SCT);
2300 3327 6                      REWRITE_RECORD(.SMQ_N);
2301 3328 5                  END;
2302 3329 5
2303 3330 5
2304 3331 5              UNLOCK_QUEUE_FILE();
2305 3332 4          END;
2306 3333 4      RETURN;
2307 3334 4      END;
2308 3335 4
2309 3336 4
2310 3337 3      SCT = .SCT[SCT_L_FLINK];
2311 3338 2      END;
2312 3339 2
2313 3340 2
2314 3341 2      ! The PID was not found in the symbiont control table.
2315 3342 2
2316 3343 2      SIGNAL(JBC$_INVMSG OR STS$K_ERROR);
2317 3344 1      END;
```

				001C 00000	.ENTRY	SYMBIONT_SERVICE, Save R2,R3,R4	: 3233
	54	00000000	EF	9E 00002	MOVAB	MBX, R4	:
	50		64	D0 00009	MOVL	MBX, R0	: 3268
	01	02	A0	91 0000C	CMPB	2(R0), #1	:
			03	13 00010	BEQL	1\$:
	20		0081	31 00012	BRW	8\$:
		03	A0	91 00015	CMPB	3(R0), #32	: 3269
			7B	1E 00019	BGEQU	8\$:
	52	50	A4	D0 0001B	MOVL	SYMBIONT_CONTROL, SCT	: 3281
			75	13 0001F	BEQL	8\$: 3282
	50		64	D0 00021	MOVL	MBX, R0	: 3284
	FC	A0	08	A2 D1 00024	CMPL	8(SCT), -4(R0)	:
			66	12 00029	BNEQ	7\$:
	14	10	50	A0 9A 0002B	MOVZBL	3(R0), R0	: 3294
	00	10	A2	50 E1 0002F	BBC	R0, 16(SCT), 5\$:
		10	A2	50 E5 00034	BBCC	R0, 16(SCT), 3\$: 3297
	00	0C	A2	50 E5 00039	BBCC	R0, 12(SCT), 4\$: 3298
			0C	A2 D5 0003E	TSTL	12(SCT)	: 3299
			60	12 00041	BNEQ	9\$:
		04	A2	01 88 00043	BISB2	#1, 4(SCT)	:
				04 00047	RET		: 3296
		53	3C A240	D0 00048	MOVL	60(SCT)[R0], SMQ_N	: 3307
				54 13 0004D	BEQL	9\$: 3308
	00000000G		EF	00 FB 0004F	CALLS	#0, LOCK_QUEUE_FILE	: 3314
				53 DD 00056	PUSHL	SMQ_N	: 3315
	00000000G		EF	01 FB 00058	CALLS	#1, READ_RECORD	:
		06	04	A0 91 0005F	CMPB	4(SMQ), #6	: 3321
				24 12 00063	BNEQ	6\$:
		52	00FC	C0 D1 00065	CMPL	252(SMQ), SCT	: 3322
				1D 12 0006A	BNEQ	6\$:
		51		64 D0 0006C	MOVL	MBX, R1	: 3323
	03	A1	0117	C0 91 0006F	CMPB	279(SMQ), 3(R1)	:
				12 12 00075	BNEQ	6\$:
				05 BB 00077	PUSHR	#M<R0,R2>	: 3326
				53 DD 00079	PUSHL	SMQ_N	:
	FB63	CF		03 FB 0007B	CALLS	#3, PROCESS_SYMBIONT_MESSAGE	: 3327
				53 DD 00080	PUSHL	SMQ_N	: 3327
	00000000G	EF		01 FB 00082	CALLS	#1, REWRITE_RECORD	:
	00000000G	EF		00 FB 00089	CALLS	#0, UNLOCK_QUEUE_FILE	: 3331
				04 00090	RET		: 3286
		52		62 D0 00091	MOVL	(SCT), SCT	: 3337
				89 11 00094	BRB	2\$: 3282
	00000000G	00	00048422	8F DD 00096	PUSHL	#295970	: 3343
				01 FB 0009C	CALLS	#1, LIB\$SIGNAL	:
				04 000A3	RET		: 3344

; Routine Size: 164 bytes, Routine Base: CODE + ODD2

```

: 2319 3345 1 GLOBAL ROUTINE SYMBIONT_DELETION: NOVALUE=
: 2320 3346 1
: 2321 3347 1 :++
: 2322 3348 1
: 2323 3349 1 FUNCTIONAL DESCRIPTION:
: 2324 3350 1 This routine checks for and processes the deletion of a symbiont.
: 2325 3351 1
: 2326 3352 1 INPUT PARAMETERS:
: 2327 3353 1 NONE
: 2328 3354 1
: 2329 3355 1 IMPLICIT INPUTS:
: 2330 3356 1 NONE
: 2331 3357 1
: 2332 3358 1 OUTPUT PARAMETERS:
: 2333 3359 1 NONE
: 2334 3360 1
: 2335 3361 1 IMPLICIT OUTPUTS:
: 2336 3362 1 NONE
: 2337 3363 1
: 2338 3364 1 ROUTINE VALUE:
: 2339 3365 1 NONE
: 2340 3366 1
: 2341 3367 1 SIDE EFFECTS:
: 2342 3368 1 NONE
: 2343 3369 1
: 2344 3370 1 --
: 2345 3371 1
: 2346 3372 2 BEGIN
: 2347 3373 2 LOCAL
: 2348 3374 2 PREV,
: 2349 3375 2 SCT: REF BBLOCK, ! Pointer to predecessor of SCT
: 2350 3376 2 SJH_N, ! Pointer to symbiont control table
: 2351 3377 2 SJH: REF BBLOCK, ! Record number of SJH
: 2352 3378 2 SMQ_N, ! Pointer to SJH
: 2353 3379 2 SMQ: REF BBLOCK; ! Record number of SMQ
: 2354 3380 2 ! Pointer to SMQ
: 2355 3381 2
: 2356 3382 2 PREV = SYMBIONT_CONTROL;
: 2357 3383 2 SCT = ..PREV;
: 2358 3384 2 WHILE .SCT NEQ 0 DO
: 2359 3385 3 BEGIN
: 2360 3386 3 IF .SCT[SCT_L_PID] EQL .MBX[ACMSL_PID]
: 2361 3387 3 THEN
: 2362 3388 4 BEGIN
: 2363 3389 4
: 2364 3390 4 ! If this process deletion is unexpected, do extra processing.
: 2365 3391 4
: 2366 3392 4 IF (.SCT[SCT_L_BITMAP] AND NOT .SCT[SCT_L_RESETTING]) NEQ 0
: 2367 3393 4 THEN
: 2368 3394 5 BEGIN
: 2369 3395 5
: 2370 3396 5 ! Signal a message.
: 2371 3397 5
: 2372 3398 5 SIGNAL(JBC$ SYMDEL + STS$K WARNING, 0,
: 2373 3399 5 (.MBX[ACMSL_FINALSTS] AND NOT STS$M_INHIB_MSG) );
: 2374 3400 5
: 2375 3401 5

```

```

: 2376      3402      5      ! Stop all queues being served by this symbiont.
: 2377      3403      5
: 2378      3404      5      INCR I FROM 0 TO 31 DO
: 2379      3405      6      BEGIN
: 2380      3406      6      SMQ_N = .VECTOR[SCT[SCT_L_QUEUES], .I];
: 2381      3407      6      IF .SMQ_N NEQ 0
: 2382      3408      6      THEN
: 2383      3409      7      BEGIN
: 2384      3410      7      SMQ = READ_RECORD(.SMQ_N);
: 2385      3411      7
: 2386      3412      7      ! If a request is pending, send a response.
: 2387      3413      7      !
: 2388      3414      7      IF .SMQ[SMQ$V_PAUSING]
: 2389      3415      7      OR .SMQ[SMQ$V_RESETTING]
: 2390      3416      7      OR .SMQ[SMQ$V_RESUMING]
: 2391      3417      7      OR .SMQ[SMQ$V_STARTING]
: 2392      3418      7      OR .SMQ[SMQ$V_STOPPING]
: 2393      3419      7      THEN
: 2394      3420      7      SCAN_INCOMPLETE_SERVICES(
: 2395      3421      7      !SRV_K_SYMBIONT,
: 2396      3422      7      !SMQ_N, .SMQ,
: 2397      3423      7      !0,
: 2398      3424      7      !JBC$_SYMDEL + STS$_K_ERROR);
: 2399      3425      7
: 2400      3426      7      ! Stop the queue.
: 2401      3427      7      !
: 2402      3428      7      SMQ[SMQ$L_STREAM_SCT] = 0;
: 2403      3429      7      SMQ[SMQ$L_STATUS] = 0;
: 2404      3430      7      SMQ[SMQ$V_STOPPED] = TRUE;
: 2405      3431      7
: 2406      3432      7      ! Rewrite the SMQ record.
: 2407      3433      7      !
: 2408      3434      7      REWRITE_RECORD(.SMQ_N);
: 2409      3435      7      END;
: 2410      3436      7      END;
: 2411      3437      7
: 2412      3438      6
: 2413      3439      5      ! Requeue current jobs on all queues being served by this symbiont.
: 2414      3440      5
: 2415      3441      5
: 2416      3442      5
: 2417      3443      5
: 2418      3444      5      INCR I FROM 0 TO 31 DO
: 2419      3445      6      BEGIN
: 2420      3446      6      SMQ_N = .VECTOR[SCT[SCT_L_QUEUES], .I];
: 2421      3447      6      IF .SMQ_N NEQ 0
: 2422      3448      6      THEN
: 2423      3449      7      BEGIN
: 2424      3450      7      SMQ = READ_RECORD(.SMQ_N);
: 2425      3451      7
: 2426      3452      7      ! Requeue the current job if there is one.
: 2427      3453      7      !
: 2428      3454      7      SJH_N = .SMQ[SMQ$L_CURRENT_LIST];
: 2429      3455      7      IF .SJH_N NEQ 0
: 2430      3456      7      THEN
: 2431      3457      7      BEGIN
: 2432      3458      8

```

B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
[
,
-
.
/

```

: 2433      3459  8      SJH = READ RECORD(.SJH_N);
: 2434      3460  8      SJH[SJH$V_SYSTEM_FAILURE] = TRUE;
: 2435      3461  8      UPDATE GETQUI_DATA(.SJH_N, .SJH);
: 2436      3462  8      COMPLETE_JOB(
: 2437      3463  8          .SJH_N, .SJH, .SMQ,
: 2438      3464  8          0,
: 2439      3465  8          JBC$ SYMDEL OR STS$K_ERROR);
: 2440      3466  8      SMQ[SMQ$E_CURRENT_LIST] = 0;
: 2441      3467  8      SMQ[SMQ$L_CURRENT_LIST_END] = 0;
: 2442      3468  8      SMQ[SMQ$B_CURRENT_JOB_COUNT] = 0;
: 2443      3469  7      END;
: 2444      3470  7
: 2445      3471  7
: 2446      3472  7      ! Rewrite the SMQ record.
: 2447      3473  7      !
: 2448      3474  7      REWRITE_RECORD(.SMQ_N);
: 2449      3475  6      END;
: 2450      3476  5      END;
: 2451      3477  4      END;
: 2452      3478  4
: 2453      3479  4
: 2454      3480  4      ! Deassign the channel to the symbiont mailbox if one has been
: 2455      3481  4      assigned.
: 2456      3482  4      !
: 2457      3483  4      IF .SCT[SCT_W_MAILBOX] NEQ 0
: 2458      3484  4      THEN
: 2459      3485  4          $DASSGN(CHAN=.SCT[SCT_W_MAILBOX]);
: 2460      3486  4
: 2461      3487  4
: 2462      3488  4      ! Finally, release the SCT entry.
: 2463      3489  4      !
: 2464      3490  4      .PREV = .SCT[SCT_L_FLINK];
: 2465      3491  4      DEALLOCATE MEMORY(.SCT);
: 2466      3492  4      QUEUE_REFERENCE_COUNT = .QUEUE_REFERENCE_COUNT - 1;
: 2467      3493  4      EXITLOOP;
: 2468      3494  4      END;
: 2469      3495  3
: 2470      3496  3
: 2471      3497  3      ! Advance to next.
: 2472      3498  3      !
: 2473      3499  3      PREV = .SCT;
: 2474      3500  3      SCT = ..PREV;
: 2475      3501  2      END;
: 2476      3502  1      END;

```

```

                                OFFC 00000          .ENTRY SYMBIONT DELETION, Save R2,R3,R4,R5,R6,R7,- : 3345
                                EF 9E 00002          R8,R9,R10,R11
5B 00000000G                    MOVAB READ_RECORD, R11
5A 00000000'                    EF 9E 00009          MOVAB SYMBIONT CONTROL, PREV : 3382
53                                6A D0 00010 1$:        MOVL (PREV), SCT : 3383
                                01 12 00013          BNEQ 2$ : 3384
                                04 00015          RET
                                EF D0 00016 2$:        MOVL MBX, R0 : 3386

```

	28	A0	08	A3	D1	0001D		CMPL	8(SCT), 40(R0)	
				03	13	00022		BEQL	3\$	
				00F7	31	00024		BRW	14\$	
		51	10	A3	D2	00027	3\$:	MCOML	16(SCT), R1	3392
		51	0C	A3	D3	0002B		BITL	12(SCT), R1	
				03	12	0002F		BNEQ	4\$	
				00C7	31	00031		BRW	1\$	
7E	4C	A0	10000000	8F	CB	00034	4\$:	BICL3	#268435456, 76(R0), -(SP)	3399
				7E	D4	0003D		CLRL	-(SP)	3398
			00048468	8F	DD	0003F		PUSHL	#296040	
	00000000G	00		03	FB	00045		CALLS	#3, LIB\$SIGNAL	
		57	3C	A3	9E	0004C		MOVAB	60(SCT), R7	3406
				54	D4	00050		CLRL	I	
		56		6744	D0	00052	5\$:	MOVL	(R7)[I], SMQ_N	
				48	13	00056		BEQL	8\$	3407
				56	DD	00058		PUSHL	SMQ_N	3410
		6B		01	FB	0005A		CALLS	#1, -READ_RECORD	
		55		50	D0	0005D		MOVL	R0, SMQ	
		52	10	A5	9E	00060		MOVAB	16(SMQ), R2	3415
10		62		03	E0	00064		BBS	#3, (R2), 6\$	
0C		62		05	E0	00068		BBS	#5, (R2), 6\$	3416
08		62		06	E0	0006C		BBS	#6, (R2), 6\$	3417
		04	01	A2	E8	00070		BLBS	1(R2), 6\$	3418
15		62		0A	E1	00074		BBC	#10, (R2), 7\$	3419
			0004846A	8F	DD	00078	6\$:	PUSHL	#296042	3425
				7E	D4	0007E		CLRL	-(SP)	3421
				55	DD	00080		PUSHL	SMQ	3423
				56	DD	00082		PUSHL	SMQ_N	
				02	DD	00084		PUSHL	#2	3421
	00000000G	EF		05	FB	00086		CALLS	#5, SCAN_INCOMPLETE_SERVICES	
			00FC	C5	D4	0008D	7\$:	CLRL	252(SMQ)	3430
				62	D4	00091		CLRL	(R2)	3431
		01	A2	02	88	00093		BISB2	#2, 1(R2)	3432
				56	DD	00097		PUSHL	SMQ_N	3437
	00000000G	EF		01	FB	00099		CALLS	#1, -REWRITE_RECORD	
AE		54		1F	F3	000A0	8\$:	AOBLEQ	#31, I, 5\$	3404
				54	D4	000A4		CLRL	I	3444
		56		6744	D0	000A6	9\$:	MOVL	(R7)[I], SMQ_N	3446
				48	13	000AA		BEQL	11\$	3447
				56	DD	000AC		PUSHL	SMQ_N	3450
		6B		01	FB	000AE		CALLS	#1, -READ_RECORD	
		55		50	D0	000B1		MOVL	R0, SMQ	
		59	48	A5	D0	000B4		MOVL	72(SMQ), SJH_N	3455
				34	13	000B8		BEQL	10\$	3456
				59	DD	000BA		PUSHL	SJH_N	3459
		6B		01	FB	000BC		CALLS	#1, -READ_RECORD	
		58		50	D0	000BF		MOVL	R0, SJH	
		11	A8	40	8F	88	000C2	BISB2	#64, 17(SJH)	3460
				58	DD	000C7		PUSHL	SJH	3461
				59	DD	000C9		PUSHL	SJH_N	
	00000000G	EF		02	FB	000CB		CALLS	#2, -UPDATE_GETQUI_DATA	
			0004846A	8F	DD	000D2		PUSHL	#296042	3465
				7E	D4	000D8		CLRL	-(SP)	3462
				55	DD	000DA		PUSHL	SMQ	3463
				58	DD	000DC		PUSHL	SJH	
				59	DD	000DE		PUSHL	SJH_N	
	00000000G	EF		05	FB	000E0		CALLS	#5, -COMPLETE_JOB	

		48	A5	7C	000E7		CLRQ	72(SMQ)	:	3466
		0115	C5	94	000EA		CLRB	277(SMQ)	:	3468
			56	DD	000EE	10\$:	PUSHL	SMQ_N	:	3474
AB	00000000G	EF	01	FB	000F0		CALLS	#1, REWRITE_RECORD	:	
		54	1F	F3	000F7	11\$:	AOBLEQ	#31, I, 9\$:	3444
		06	A3	B5	000FB	12\$:	TSTW	6(SCT)	:	3483
			0B	13	000FE		BEQL	13\$:	
		06	A3	3C	00100		MOVZWL	6(SCT), -(SP)	:	3485
	000J0000G	7E	01	FB	00104		CALLS	#1, SYS\$DASSGN	:	
		00	63	D0	0010B	13\$:	MOVL	(SCT), (PREV)	:	3490
		6A	53	DD	0010E		PUSHL	SCT	:	3491
	00C00000G	EF	01	FB	00110		CALLS	#1, DEALLOCATE_MEMORY	:	
			EF	D7	00117		DECL	QUEUE_REFERENCE_COUNT	:	3492
		00000000'	04	0011D			RET		:	3388
		5A	53	D0	0011E	14\$:	MOVL	SCT, PREV	:	3499
			FEEC	31	00121		BRW	1\$:	3500
			04	00124			RET		:	3502

; Routine Size: 293 bytes, Routine Base: CODE + 0E76

```

: 2478 3503 1 GLOBAL ROUTINE DELETE_SYMBIONTS: NOVALUE=
: 2479 3504 1
: 2480 3505 1 !++
: 2481 3506 1
: 2482 3507 1 FUNCTIONAL DESCRIPTION:
: 2483 3508 1 This routine deletes all symbiont processes just before the job
: 2484 3509 1 controller restarts itself after a fatal error.
: 2485 3510 1
: 2486 3511 1 INPUT PARAMETERS:
: 2487 3512 1 NONE
: 2488 3513 1
: 2489 3514 1 IMPLICIT INPUTS:
: 2490 3515 1 NONE
: 2491 3516 1
: 2492 3517 1 OUTPUT PARAMETERS:
: 2493 3518 1 NONE
: 2494 3519 1
: 2495 3520 1 IMPLICIT OUTPUTS:
: 2496 3521 1 NONE
: 2497 3522 1
: 2498 3523 1 ROUTINE VALUE:
: 2499 3524 1 NONE
: 2500 3525 1
: 2501 3526 1 SIDE EFFECTS:
: 2502 3527 1 NONE
: 2503 3528 1
: 2504 3529 1 --
: 2505 3530 1
: 2506 3531 2 BEGIN
: 2507 3532 2 LOCAL
: 2508 3533 2 SCT: REF BBLOCK; ! Pointer to symbiont control table
: 2509 3534 2
: 2510 3535 2
: 2511 3536 2 SCT = .SYMBIONT CONTROL;
: 2512 3537 2 WHILE .SCT NEQ 0 DO
: 2513 3538 3 BEGIN
: 2514 3539 3 $DELPRC(PIDADR=SCT[SCT_L_PID]);
: 2515 3540 3 SCT = .SCT[SCT_L_FLINK];
: 2516 3541 2 END;
: 2517 3542 1 END;

```

```

                                .EXTRN  SYS$DELPRC
                                .ENTRY  DELETE_SYMBIONTS, Save R2
                                MOVL    SYMBIONT_CONTROL, SCT           : 3503
                                BEQL    2$                               : 3536
                                CLRL    -(SP)                          : 3537
                                PUSHAB  8(SCT)                          : 3539
                                CALLS   #2, SYS$DELPRC
                                MOVL    (SCT), SCT                       : 3540
                                BRB     1$                               : 3537
                                RET

```

; Routine Size: 29 bytes, Routine Base: CODE + 0F9B

SYMBIONT
V04-000

Symbiont communication

F 16
16-Sep-1984 00:37:14
14-Sep-1984 12:37:15

VAX-11 Bliss-32 V4.0-742
[JOBCTL.SRC]SYMBIONT.B32;1

Page 79
(16)

```
: 2519 3543 1 GLOBAL ROUTINE SYMBIONT_COMPLETED_BLOCKS(SJH)=
: 2520 3544 1
: 2521 3545 1 |++
: 2522 3546 1 |
: 2523 3547 1 | FUNCTIONAL DESCRIPTION:
: 2524 3548 1 | This routine analyzes the checkpoint entry for a job and returns the
: 2525 3549 1 | number of completed blocks in the current file.
: 2526 3550 1 |
: 2527 3551 1 | INPUT PARAMETERS:
: 2528 3552 1 | SJH - Pointer to SJH.
: 2529 3553 1 |
: 2530 3554 1 | IMPLICIT INPUTS:
: 2531 3555 1 | NONE
: 2532 3556 1 |
: 2533 3557 1 | OUTPUT PARAMETERS:
: 2534 3558 1 | NONE
: 2535 3559 1 |
: 2536 3560 1 | IMPLICIT OUTPUTS:
: 2537 3561 1 | NONE
: 2538 3562 1 |
: 2539 3563 1 | ROUTINE VALUE:
: 2540 3564 1 | Number of completed blocks, or 0 if indeterminate.
: 2541 3565 1 |
: 2542 3566 1 | SIDE EFFECTS:
: 2543 3567 1 | NONE
: 2544 3568 1 |
: 2545 3569 1 | --
: 2546 3570 1 |
: 2547 3571 2 BEGIN
: 2548 3572 2 MAP
: 2549 3573 2 SJH: REF BBLOCK; ! Pointer to SJH
: 2550 3574 2
: 2551 3575 2
: 2552 3576 2 ! If the checkpoint is short enough to fit into the main area, and the
: 2553 3577 2 ! structure level is correct, then return the first longword of the user
: 2554 3578 2 ! key, which is known to be the current VBN.
: 2555 3579 2 |
: 2556 3580 2 IF .BBLOCK[SJH[SJH$T_CHECKPOINT], FVDF_LENGTH] LEQU SJH$S_CHECKPOINT-2
: 2557 3581 2 THEN
: 2558 3582 3 BEGIN
: 2559 3583 3 BIND
: 2560 3584 3 CKP = BBLOCK[SJH[SJH$T_CHECKPOINT], FVDF_DATA] : BBLOCK;
: 2561 3585 3
: 2562 3586 3
: 2563 3587 3 IF .CKP[SMBMSG$B_CHECKPOINT_LEVEL] EQL SMBMSG$K_STRUCTURE_LEVEL
: 2564 3588 3 THEN
: 2565 3589 3 RETURN .(CKP[SMBMSG$Q_USER_KEY]);
: 2566 3590 2 END;
: 2567 3591 2
: 2568 3592 2
: 2569 3593 2 ! Unknown checkpoint, or none stored -- return 0.
: 2570 3594 2 |
: 2571 3595 2 0
: 2572 3596 1 END;
```

			0000 00000	.ENTRY	SYMBIONT_COMPLETED_BLOCKS, Save nothing	:	3543
50	04	AC	D0 00002	MOVL	SJH, R0	:	3580
1E	0180	CO	B1 00006	CMPW	384(R0), #30	:	
		10	1A 0000B	BGTRU	1\$:	
50	0182	CO	9E 0000D	MOVAB	386(R0), R0	:	3584
01	01	AO	91 00012	CMPB	1(R0), #1	:	3587
		05	12 00016	BNEQ	1\$:	
50	10	AO	D0 00018	MOVL	16(R0), R0	:	3589
			04 0001C	RET		:	
		50	D4 0001D 1\$:	CLRL	R0	:	3596
			04 0001F	RET		:	

; Routine Size: 32 bytes, Routine Base: CODE + 0FB8

SYMBIONT
V04-000

Symbiont communication

I 16
16-Sep-1984 00:37:14
14-Sep-1984 12:37:15

VAX-11 Bliss-32 V4.0-742
[JOBCTL.SRC]SYMBIONT.B32;1

Page 82
(18)

: 2574 3597 1 END
: 2575 3598 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	5024	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(2)
CODE	4056	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	----- Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	178	0	1000	00:01.4

COMMAND QUALIFIERS

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

: Size: 3966 code + 5114 data bytes
: Run Time: 01:06.5
: Elapsed Time: 04:11.7
: Lines/CPU Min: 3245
: Lexemes/CPU-Min: 35206
: Memory Used: 653 pages
: Compilation Complete

