

PPPPPPPPPP	RRRRRRRRRR	TTTTTTTTTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBB			
PPPPPPPPPP	RRRRRRRRRR	TTTTTTTTTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBB			
PPPPPPPPPP	RRRRRRRRRR	TTTTTTTTTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBB			
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	PPP	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPPPPPPPPP	RRRRRRRRRR	TTTTTTTTTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBB			
PPPPPPPPPP	RRRRRRRRRR	TTTTTTTTTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBB			
PPPPPPPPPP	RRRRRRRRRR	TTTTTTTTTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBB			
PPP	RRR	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	RRR	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	RRR	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	RRR	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	RRR	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	RRR	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	RRR	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPP	RRR	RRR	RRR	TTT	SSS	MMM	MMM	BBB	BBB
PPPPPPPPPP	RRRRRRRRRR	TTTTTTTTTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBB			
PPPPPPPPPP	RRRRRRRRRR	TTTTTTTTTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBB			
PPPPPPPPPP	RRRRRRRRRR	TTTTTTTTTT	SSSSSSSSSS	MMM	MMM	BBBBBBBBBB			

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

FFFFFFFFF  000000  RRRRRRRR  MM      MM  AAAAAA  TTTTTTTTTT
FFFFFFFFF  000000  RRRRRRRR  MM      MM  AAAAAA  TTTTTTTTTT
FF          00      00  RR      RR  MMMM  MMMM  AA      AA  TT
FF          00      00  RR      RR  MMMM  MMMM  AA      AA  TT
FF          00      00  RR      RR  MM  MM  MM  AA      AA  TT
FF          00      00  RR      RR  MM  MM  MM  AA      AA  TT
FFFFFFFFF  00      00  RRRRRRRR  MM      MM  AA      AA  TT
FFFFFFFFF  00      00  RRRRRRRR  MM      MM  AA      AA  TT
FF          00      00  RR  RR    MM      MM  AAAAAAAAAA  TT
FF          00      00  RR  RR    MM      MM  AAAAAAAAAA  TT
FF          00      00  RR      RR  MM      MM  AA      AA  TT
FF          00      00  RR      RR  MM      MM  AA      AA  TT
FF          00      00  RR      RR  MM      MM  AA      AA  TT
FF          000000  RR      RR  MM      MM  AA      AA  TT
FF          000000  RR      RR  MM      MM  AA      AA  TT

```

```

LL          111111  SSSSSSSS
LL          111111  SSSSSSSS
LL          11      SS
LL          11      SS
LL          11      SS
LL          11      SS
LL          11      SSSSSS
LL          11      SSSSSS
LL          11      SS
LL          11      SS
LL          11      SS
LL          11      SS
LLLLLLLLLL  111111  SSSSSSSS
LLLLLLLLLL  111111  SSSSSSSS

```



```

1 0001 0 MODULE FORMAT ( %TITLE 'Print Symbiont format routines'
2 0002 0 IDENT = 'V04-000'
3 0003 0 ADDRESSING_MODE (EXTERNAL = GENERAL)
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
12 0012 1 * ALL RIGHTS RESERVED. *
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
19 0019 1 * TRANSFERRED. *
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
23 0023 1 * CORPORATION. *
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1
32 0032 1 **
33 0033 1 FACILITY:
34 0034 1 Print Symbiont.
35 0035 1
36 0036 1 ABSTRACT:
37 0037 1 Output formatting routines.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1 VAX/VMS user mode.
41 0041 1 --
42 0042 1
43 0043 1 AUTHOR: G. Robert, CREATION DATE: 31-Aug-1982
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1 V03-010 GRR0012 Gregory R. Robert 25-Jul-1984
48 0048 1 Add a check to determine if the call to format_line_number
49 0049 1 originated from the psmSk main input. If it did then
50 0050 1 we surely wish to format the line number otherwise it
51 0051 1 could be call from the page header routine(or some such)
52 0052 1 where line numbers are not allowed.
53 0053 1
54 0054 1 V03-009 GRR0011 Gregory R. Robert 11-Jul-1984
55 0055 1 Change conditional format effector logic to only
56 0056 1 suppress form feeds allowing line feeds and vertical
57 0057 1 tabs to pass. Fix column setting in FORMAT_LINE_NUMBER

```

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89

0058 1
0059 1
0060 1
0061 1
0062 1
0063 1
0064 1
0065 1
0066 1
0067 1
0068 1
0069 1
0070 1
0071 1
0072 1
0073 1
0074 1
0075 1
0076 1
0077 1
0078 1
0079 1
0080 1
0081 1
0082 1
0083 1
0084 1
0085 1
0086 1
0087 1
0088 1
0089 1

to correct a tab computation problem.

V03-008 GRR0010 Gregory R. Robert 09-May-1984
Fix PSM\$FORMAT interface to correspond to new user interface.

V03-007 GRR0009 Gregory R. Robert 29-Apr-1984
FT2 bugfixes and margins.

V03-006 GRR0008 Gregory R. Robert 12-Apr-1984
Backoff form feed fix of preceeding update, requires
complimentary fix in cursor update for form feeds.

V03-005 GRR0007 Gregory R. Robert 12-Apr-1984
Fix tab computation and fix conditional form feeds
for the case of only one line on the page.

V03-004 GRR0006 Gregory R. Robert 26-Sep-1983
Fix ACCVIO when printing files with sequence numbers.

3B-003 GRR3003 Gregory R. Robert 23-Aug-1983
Bugfixes, page_setup_modules, form_setup_modules,
sheet_feed, symbiont_initiated_pause_task and stop_stream,
hangup code, read and write item services

3B-002 GRR3002 Gregory R. Robert 03-Aug-1983
Rewrite for new design.

3B-001 GRR3001 Gregory R. Robert 22-Jul-1983
Created separate module.

...

...

FORMAT
V04-000

Print Symbiont format routines

K 15
16-Sep-1984 02:14:45
14-Sep-1984 12:55:08

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]FORMAT.B32;1

Page 3
(2)

FC
VC

```

: 91      0090 1 LIBRARY 'SYSSLIBRARY:LIB';
: 92      0091 1 REQUIRE 'LIBS:SMBDEF';
: 93      0583 1 REQUIRE 'SRCS:SMBREQ';
: 94      1040 1
: 95      1041 1 EXTERNAL ROUTINE
: 96      1042 1     PSM$INCLUDE_MODULES,
: 97      1043 1     PSM$STORE_ERRORS;
: 98      1044 1
: 99      1045 1 FORWARD ROUTINE
100      1046 1     PSM$MAIN_FORMAT
101      1047 1
102      1048 1     FORMAT_CHARACTER
103      1049 1     FORMAT_ESCAPE_SEQUENCE
104      1050 1     FORMAT_FIRST_CHARACTER
105      1051 1     FORMAT_LINE_NUMBER
106      1052 1     FORMAT_LINE_OVERFLOW
107      1053 1     FORMAT_NEW_PAGE
108      1054 1
109      1055 1
110      1056 1 LITERAL
111      1057 1     S_VMS_OSC = %CHARCOUNT (%CHAR(PMSK_CHAR_ESC, PMSK_CHAR_OSC), 'VMS:1;')
112      1058 1
113      1059 1
```

```

115 1060 1 %SBTTL 'FORMAT -- Main format routine'
116 1061 1 Functional Description:
117 1062 1 This routine controls the formatting of output records.
118 1063 1 To the original input record the appropriate carriage
119 1064 1 control is applied, as well as line numbers if sequenced.
120 1065 1 All format effectors (LF, CR, FF, VT, BS) are scanned
121 1066 1 and the current line/column position is updated.
122 1067 1
123 1068 1 Wrap and truncate are handled.
124 1069 1
125 1070 1 Escape sequences are detected and processed.
126 1071 1
127 1072 1 New page is detected and reported.
128 1073 1
129 1074 1 Formal Parameters:
130 1075 1 SCB : SCB address
131 1076 1
132 1077 1 Implicit Inputs:
133 1078 1 various fields from the SCB.
134 1079 1
135 1080 1 Implicit Outputs:
136 1081 1 The input record is formatted and copied to the output buffer.
137 1082 1
138 1083 1 Returned Value:
139 1084 1 $$$ NORMAL - normal successful completion
140 1085 1 PSMS_BUFFEROVF - output buffer full
141 1086 1 PSMS_NEWPAGE - new page detected
142 1087 1 PSMS_ESCAPE - escape sequence detected
143 1088 1
144 1089 1 Side Effects:
145 1090 1 none
146 1091 1 --
147 1092 1 GLOBAL ROUTINE PSMSMAIN_FORMAT (
148 1093 1 SMB_CONTEXT : REF $LONGWORD, : SCB address
149 1094 1 USER_CONTEXT : REF VECTOR, : - not used
150 1095 1 FUNCTION : REF $LONGWORD, : service request code
151 1096 1 DESC_1 : REF VECTOR, : Input record descriptor
152 1097 1 ARG_1 : REF $LONGWORD, : Input carriage control
153 1098 1 DESC_2 : REF VECTOR, : Output record descriptor
154 1099 1 ARG_2 : REF $LONGWORD : - not used
155 1100 1 ) =
156 1101 2 BEGIN
157 1102 2
158 1103 2 LOCAL
159 1104 2 REMAINDER
160 1105 2 SCB : REF $BLOCK,
161 1106 2 USED
162 1107 2 XLATE_LIMIT
163 1108 2 ;
164 1109 2
165 1110 2
166 1111 2 ! If not a FORMAT call then ignore it
167 1112 2 !
168 1113 2 IF .FUNCTION[] NEQ PSMSK_FORMAT THEN RETURN PSMS_FUNNOTSUP;
169 1114 2
170 1115 2
171 1116 2 ! Get the SCB address

```

```

172 1117 2 !
173 1118 2 ! SCB = .SMB_CONTEXT[];
174 1119 2
175 1120 2
176 1121 2 ! See if there is anything to do
177 1122 2
178 1123 2 IF .SCB[PSMSB_PREFIX_COUNT] EQL 0
179 1124 2 AND .SCB_SIZE_ (INPUT_RECORD) EQL 0
180 1125 2 AND .SCB[PSMSB_POSTFIX_COUNT] EQL 0
181 1126 2 THEN
182 1127 2     RETURN SSS_NORMAL;
183 1128 2
184 1129 2
185 1130 2 ! Check for first record of input service
186 1131 2
187 1132 2 IF .SCB[PSMSV_FIRST_RECORD]
188 1133 2 THEN
189 1134 2     BEGIN
190 1135 2     RETURN IF_ERROR_ (FORMAT_FIRST_CHARACTER (.SCB));
191 1136 2     SCB[PSMSV_FIRST_RECORD] = 0;
192 1137 2     END;
193 1138 2
194 1139 2
195 1140 2 ! Output any leading carriage control
196 1141 2
197 1142 2 WHILE .SCB[PSMSB_PREFIX_COUNT] NEQ 0
198 1143 2 DO
199 1144 2     BEGIN
200 1145 2     RETURN IF_ERROR_ (FORMAT_CHARACTER (.SCB, .SCB[PSMSB_PREFIX_CHAR]));
201 1146 2     DECREMENT_ (SCB[PSMSB_PREFIX_COUNT]);
202 1147 2     END;
203 1148 2
204 1149 2
205 1150 2 ! If escape in progress then call the escape formatting routine
206 1151 2
207 1152 2 IF .SCB[PSMSV_ESCAPE_IN_PROGRESS]
208 1153 2 THEN
209 1154 2     RETURN_IF_ERROR_ (FORMAT_ESCAPE_SEQUENCE (.SCB));
210 1155 2
211 1156 2
212 1157 2 ! If starting new record see if sequence number required
213 1158 2
214 1159 2 IF .SCB[PSMSV_NEW_RECORD]
215 1160 2 THEN
216 1161 2     BEGIN
217 1162 2     IF .PRINT_FLAG_ (SEQUENCED)
218 1163 2     AND .SCB[PSMSB_SERVICE_INDEX] EQL PSM&K_MAIN_INPUT
219 1164 2     THEN
220 1165 2     RETURN IF_ERROR_ (FORMAT_LINE_NUMBER (.SCB));
221 1166 2     SCB[PSMSV_NEW_RECORD] = 0;
222 1167 2     END;
223 1168 2
224 1169 2
225 1170 2 ! Output any record contents
226 1171 2
227 1172 2 WHILE .SCB_SIZE_ (INPUT_RECORD) NEQ 0
228 1173 2 DO

```

```

229      1174      3      BEGIN
230      1175      3
231      1176      3      ! While we are in the top margin area, or are starting a new line
232      1177      3      ! let the format_character routine handle things
233      1178      3      !
234      1179      4      WHILE (
235      1180      5          (.SCB[PSM$LINE] LEQU .SCB[PSM$T_MARGIN])
236      1181      5          OR (.SCB[PSM$COLUMN] EQLU 0)
237      1182      4      )
238      1183      3      AND .SCB_SIZE_ (INPUT_RECORD) NEQ 0
239      1184      3      DO
240      1185      4          BEGIN
241      1186      4          RETURN IF ERROR (
242      1187      4              FORMAT_CHARACTER (.SCB, CH$RCHAR (.SCB_ADDR_ (INPUT_RECORD)));
243      1188      4          READ_CHAR_ ;
244      1189      4          END;
245      1190      3
246      1191      3      ! If in top margin area then we are starting a new page
247      1192      3      !
248      1193      3      IF .SCB[PSM$LINE] LEQU .SCB[PSM$T_MARGIN]
249      1194      3      THEN
250      1195      3          RETURN FORMAT_NEW_PAGE (.SCB);
251      1196      3
252      1197      3      ! Compute the maximum translation as the smaller of columns remaining,
253      1198      3      ! output buffer remaining, and input record remaining
254      1199      3      !
255      1200      3      XLATE_LIMIT = .SCB[PSM$FORM_WIDTH] -
256      1201      3          .SCB[PSM$RIGHT_MARGIN] = .SCB[PSM$COLUMN] + 1;
257      1202      3
258      1203      3      IF (.SCB[PSM$PRINT_FLAGS] AND (SMBMSG$M_TRUNCATE OR SMBMSG$M_WRAP)) EQL 0
259      1204      3      THEN
260      1205      3          XLATE_LIMIT = %X '7FFFFFFF';
261      1206      3
262      1207      3      IF .SCB_SIZE_ (INPUT_RECORD) LSSU .XLATE_LIMIT
263      1208      3      THEN
264      1209      3          XLATE_LIMIT = .SCB_SIZE_ (INPUT_RECORD);
265      1210      3
266      1211      3      ! Move the input record to the output buffer translating enroute. Stop
267      1212      3      ! if any format effectors sensed or when limit reached.
268      1213      3      !
269      1214      3      MOVTUC (
270      1215      3          XLATE_LIMIT,                ! - smaller of input size and line size
271      1216      3          .SCB_ADDR_ (INPUT_RECORD),    ! - remaining input address
272      1217      3          UPLIT BYTE (255),            ! - stop translation character
273      1218      3          .SCB[PSM$A_XLATE_TABLE],    ! - translation table address
274      1219      3          .SCB_SIZE_ (OUTPUT_BUFFER),  ! - remaining buffer size
275      1220      3          .SCB_ADDR_ (OUTPUT_BUFFER)  ! - remaining output address
276      1221      3          ;
277      1222      3          REMAINDER,                ! - R0 = unused input size
278      1223      3          .SCB_ADDR_ (INPUT_RECORD),    ! - R1 = updated input address
279      1224      3          .,                          ! - R2 = <not used>
280      1225      3          .,                          ! - R3 = <not used>
281      1226      3          .SCB_SIZE_ (OUTPUT_BUFFER),  ! - R4 = unused output size
282      1227      3          .SCB_ADDR_ (OUTPUT_BUFFER)  ! - R5 = updated output address
283      1228      3      );
284      1229      3
285      1230      3

```

B
C
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
[
\
]
^
_
`
a
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
{
|
}
~
?
@

\$
%
&
'
(
)
*
+
,
-
.
:
;
=
>
?
@

\$
%
&
'
(
)
*
+
,
-
.
:
;
=
>
?


```

: 286      1231      3      ! Update counts and position
: 287      1232      3      !
: 288      1233      3      USED = .XLATE_LIMIT - .REMAINDER;
: 289      1234      3      SCB[PSMSL_COLUMN] = .SCB[PSMSL_COLUMN] + .USED;
: 290      1235      3      SCB_SIZE_(INPUT_RECORD) = .SCB_SIZE_(INPUT_RECORD) - .USED;
: 291      1236      3      !
: 292      1237      3      !
: 293      1238      3      ! If remaining output buffer zero then return buffer overflow.
: 294      1239      3      !
: 295      1240      3      IF .SCB_SIZE_(OUTPUT_BUFFER) EQL 0 THEN RETURN PSMS_BUFFEROVF;
: 296      1241      3      !
: 297      1242      3      !
: 298      1243      3      ! if input remains we must have stopped due to a format effector --
: 299      1244      3      ! let the format character routine handle it.
: 300      1245      3      !
: 301      1246      3      IF .SCB_SIZE_(INPUT_RECORD) NEQ 0
: 302      1247      3      THEN
: 303      1248      4      BEGIN
: 304      1249      4      RETURN IF_ERROR (
: 305      1250      4      FORMAT_CHARACTER (.SCB, CH$RCHAR (.SCB_ADDR_(INPUT_RECORD)));
: 306      1251      4      READ_CHAR_;
: 307      1252      4      END;
: 308      1253      3      END;
: 309      1254      2      !
: 310      1255      2      !
: 311      1256      2      ! Output any trailing carriage control
: 312      1257      2      !
: 313      1258      2      WHILE .SCB[PSMSB_POSTFIX_COUNT] NEQ 0
: 314      1259      2      DO
: 315      1260      3      BEGIN
: 316      1261      3      RETURN IF_ERROR (FORMAT_CHARACTER (.SCB, .SCB[PSMSB_POSTFIX_CHAR]));
: 317      1262      3      DECREMENT_(SCB[PSMSB_POSTFIX_COUNT]);
: 318      1263      3      END;
: 319      1264      2      !
: 320      1265      2      SSS_NORMAL
: 321      1266      2      !
: 322      1267      1      END;

```

```

.TITLE  FORMAT Print Symbiont format routines
.IDENT  \V04-000\

.PSECT  CODE,NOWRT,2

FF 0000 P.AAA: .BYTE  -1

.EXTRN  BASSEDIT, LBR$CLOSE
.EXTRN  LBR$GET_RECORD, LBR$INI_CONTROL
.EXTRN  LBR$LOOKUP_KEY, LBR$OPEN
.EXTRN  LBR$RET_RMSSTV, LBR$SET_LOCATE
.EXTRN  LIB$TRIM_FILESPEC
.EXTRN  LIB$GET_VM, LIB$FREE_VM
.EXTRN  STR$ANALYZE_SDESC
.EXTRN  STR$ANALYZE_SDESC_R1
.EXTRN  STR$APPEND, STR$CONCAT
.EXTRN  STR$COPY_DX, STR$COPY_R
.EXTRN  STR$FREE_DX, STR$FREE1_DX_R4

```

				OFFC 00000	.EXTRN	STR\$GET1 DX, STR\$LEFT	
					.EXTRN	STR\$PREFIX, STR\$RIGHT	
					.EXTRN	PSMS HANGUP DISPATCH ENTRY	
					.EXTRN	PSMS_BUFFEROVF, PSMS_EOF	
					.EXTRN	PSMS_ESCAPE, PSMS_FLUSH	
					.EXTRN	PSMS_FUNNOTSUP, PSMS_INVITMCD	
					.EXTRN	PSMS_INVVMSOSC, PSMS_MODNOTFND	
					.EXTRN	PSMS_NEWPAGE, PSMS_NOFILEID	
					.EXTRN	PSMS_OSCTOOLON, PSMS_PENDING	
					.EXTRN	PSMS_SUSPEND, PSMS_TOOMANYLEV	
					.EXTRN	SMBS_INVSTMNBR, SMBS_INVSTRLEV	
					.EXTRN	SMBS_NOMOREITEMS	
					.EXTRN	PSMSINCLUDE MODULES	
					.EXTRN	PSMSSTORE_ERRORS	
					.ENTRY	PSMSMAIN FORMAT, Save R2,R3,R4,R5,R6,R7,R8,-;	1092
						R9,R10,RT1	
					SUBL2	#8, SP	
					CMPL	@FUNCTION, #3	1113
					BEQL	1\$	
					MOVL	#PSMS_FUNNOTSUP, R0	
					RET		
					MOVL	@SMB_CONTEXT, SCB	1118
					MOVAB	632(SCB), R2	1123
					TSTB	(R2)	
					BNEQ	2\$	
					TSTW	608(SCB)	1124
					BNEQ	2\$	
					TSTB	634(SCB)	1125
					BNEQ	2\$	
					BRW	2i\$	
					MOVAB	16(SCB), R3	1132
					BBC	#5, (R3), 3\$	
					PUSHL	SCB	1135
					CALLS	#1, FORMAT_FIRST_CHARACTER	
					BLBC	STATUS, 11\$	
					BICB2	#32, (R3)	1136
					TSTB	(R2)	1142
					BEQL	4\$	
					MOVZBL	633(SCB), -(SP)	1145
					PUSHL	SCB	
					CALLS	#2, FORMAT_CHARACTER	
					BLBC	STATUS, 11\$	
					DECB	(R2)	1146
					BRB	3\$	1142
					BBC	#3, (R3), 5\$	1152
					PUSHL	SCB	1154
					CALLS	#1, FORMAT_ESCAPE_SEQUENCE	
					BLBC	STATUS, 11\$	
					TSTB	(R3)	1159
					BGEQ	7\$	
					BBC	#4, 516(SCB), 6\$	1162
					CMPB	637(SCB), #16	1163
					BNEQ	6\$	
					PUSHL	SCB	1165
					CALLS	#1, FORMAT_LINE_NUMBER	
					BLBC	STATUS, 11\$	

			63	80	8F	8A	00084	6\$:	BICB2	#128, (R3)	1166	
			58	0260	C6	9E	00088	7\$:	MOVAB	608(SCB), R8	1172	
			5A	04	A8	9E	0008D		MOVAB	4(R8), R10	1187	
			5B	0194	C6	9E	00091		MOVAB	404(SCB), R11	1201	
			59	C1E0	C6	9E	00096		MOVAB	480(SCB), R9	1219	
					68	B5	0009B	8\$:	TSTW	(R8)	1172	
					03	12	0009D		BNEQ	9\$		
					00AC	31	0009F		BRW	19\$		
	0230		C6	01C8	C6	D1	000A2	9\$:	C MPL	456(SCB), 560(SCB)	1180	
				0194	C6	1B	000A9		BLEQU	10\$		
					C6	D5	000AB		TSTL	404(SCB)	1181	
					19	12	000AF		BNEQ	13\$		
					68	B5	000B1	10\$:	TSTW	(R8)	1183	
					15	13	000B3		BEQL	13\$		
		7E		00	BA	9A	000B5		MOVZBL	@0(R10), -(SP)	1187	
					56	DD	000B9		PUSHL	SCB		
	0000V		CF		02	FB	000BB		CALLS	#2, FORMAT CHARACTER		
			01		50	E8	000C0	11\$:	BLBS	STATUS, 12\$		
						04	000C3		RET			
					68	B7	000C4	12\$:	DECW	(R8)		
					6A	D6	000C6		INCL	(R10)		
					D8	11	000C8		BRB	9\$	1179	
	0230		C6	01C8	C6	D1	000CA	13\$:	C MPL	456(SCB), 560(SCB)	1193	
					08	1A	000D1		BGTRU	14\$		
					56	DD	000D3		PUSHL	SCB	1195	
	0000V		CF		01	FB	000D5		CALLS	#1, FORMAT_NEW_PAGE		
						04	000DA		RET			
		50	008C	C6	0148	C6	C3	000DB	14\$:	SUBL3	328(SCB), 140(SCB), R0	1201
				50		6B	C2	000E3		SUBL2	(R11), R0	
				57	01	A0	9E	000E6		MOVAB	1(R0), XLATE_LIMIT	
			C0	8F	0204	C6	93	000EA		BITB	516(SCB), #192	1203
						07	12	000F0		BNEQ	15\$	
				57	7FFFFFFF	8F	D0	000F2		MOVL	#2147483647, XLATE_LIMIT	1205
	57		68	10		00	ED	000F9	15\$:	C MPZV	#0, #16, (R8), XLATE_LIMIT	1207
						03	1E	000FE		BGEQU	16\$	
				57		68	3C	00100		MOVZWL	(R8), XLATE_LIMIT	1209
	0244	D6	FEF5	CF	00	BA	2F	00103	16\$:	MOVTUC	XLATE_LIMIT, @0(R10), P.AAA, @580(SCB), -	1227
					04	B9		0010D			(R9), @4(R9)	
					04	AE		00110		MOVL	R0, 4(SP)	
						6A		00114		MOVL	R1, (R10)	1223
						69		00117		MOVW	R4, (R9)	1226
				04		A9		0011A		MOVL	R5, 4(R9)	1227
			6E		04	AE	C3	0011E		SUBL3	REMAINDER, XLATE_LIMIT, USED	1233
						6E	C0	00123		ADDL2	USED, (R11)	1234
						6E	A2	00126		SUBW2	USED, (R8)	1235
						69	B5	00129		TSTW	(R9)	1240
						08	12	0012B		BNEQ	17\$	
			50	00000000G	8F	D0	0012D		MOVL	#PSMS_BUFFEROVF, R0		
						04	00134		RET			
						68	B5	00135	17\$:	TSTW	(R8)	1246
						12	13	00137		BEQL	18\$	
				7E	00	BA	9A	00139		MOVZBL	@0(R10), -(SP)	1250
						56	DD	0013D		PUSHL	SCB	
			0000V	CF		02	FB	0013F		CALLS	#2, FORMAT CHARACTER	
				26		50	E9	00144		BLBC	STATUS, 22\$	
						68	B7	00147		DECW	(R8)	
						6A	D6	00149		INCL	(R10)	

FORMAT
V04-000

Print Symbiont format routines
FORMAT -- Main format routine

E 16
16-Sep-1984 02:14.45
14-Sep-1984 12:55.08

VAX-11 BLISS-32 V4.0-742
[PRTSMB.SRC]FORMAT.B32;1

Page 10
(3)

		FF4D	31	0014B	18\$:	BRW	8\$:	1172
52	027A	C6	9E	0014E	19\$:	MOVAB	634(SCB), R2	:	1258
		62	95	00153	20\$:	TSTB	(R2)	:	
		13	13	00155		BEQL	21\$:	
7E	027B	C6	9A	00157		JVZBL	635(SCB), -(SP)	:	1261
		56	DD	0015C		PUSHL	SCB	:	
0000V	CF	02	FB	0015E		CALLS	#2, FORMAT CHARACTER	:	
	J7	50	E9	00163		BLBC	STATUS, 22\$:	
		62	97	00166		DECB	(R2)	:	1262
		E9	11	00168		BRB	20\$:	1258
50		01	D0	0016A	21\$:	MOVL	#1, R0	:	1267
		04	0016D	22\$:	RET			:	

; Routine Size: 366 bytes, Routine Base: CODE + 0001

```

: 324 1268 1 %SBTTL 'FORMAT_CHARACTER - format one character'
: 325 1269 1 | Functional Description:
: 326 1270 1 |     Handles special characters such as format effectors.
: 327 1271 1 |
: 328 1272 1 | Formal Parameters:
: 329 1273 1 |     SCB      : SCB address
: 330 1274 1 |     CHAR     : character to be output
: 331 1275 1 |
: 332 1276 1 | Implicit Inputs:
: 333 1277 1 |     none
: 334 1278 1 |
: 335 1279 1 | Implicit Outputs:
: 336 1280 1 |     The character is formatted and placed in the output buffer.
: 337 1281 1 |
: 338 1282 1 | Returned Value:
: 339 1283 1 |     $$$_NORMAL or any error
: 340 1284 1 |
: 341 1285 1 | Side Effects:
: 342 1286 1 |     none
: 343 1287 1 | --
: 344 1288 1 ROUTINE FORMAT_CHARACTER (
: 345 1289 1 |     SCB      : REF $BLOCK,
: 346 1290 1 |     CHAR     : BYTE
: 347 1291 1 | ) =
: 348 1292 2 BEGIN
: 349 1293 2
: 350 1294 2 LOCAL
: 351 1295 2 |     COLUMNS_LEFT : SIGNED,
: 352 1296 2 |     LINES_LEFT    : SIGNED
: 353 1297 2 | ;
: 354 1298 2
: 355 1299 2
: 356 1300 2 | If line number is zero then we are attempt to write to a new page
: 357 1301 2 |
: 358 1302 2 IF .SCB[PSM$L_LINE] EQLU 0
: 359 1303 2 THEN
: 360 1304 2 |     RETURN FORMAT_NEW_PAGE (.SCB);
: 361 1305 2
: 362 1306 2
: 363 1307 2 | Make sure there is ample room in the output buffer
: 364 1308 2 |
: 365 1309 2 IF .SCB_SIZE_ (OUTPUT_BUFFER) LSSU 50
: 366 1310 2 THEN
: 367 1311 2 |     RETURN PSM$_BUFFEROVF;
: 368 1312 2
: 369 1313 2
: 370 1314 2 | If the column is zero we are starting a new line -- handle the
: 371 1315 2 | left margin
: 372 1316 2 |
: 373 1317 2 IF .SCB[PSM$L_COLUMN] EQLU 0
: 374 1318 2 THEN
: 375 1319 3 |     BEGIN
: 376 1320 3 |     SCB_ADDR (OUTPUT_BUFFER) =
: 377 1321 3 |     CH$FILL (%C ^', .SCB[PSM$L_L_MARGIN], .SCB_ADDR (OUTPUT_BUFFER));
: 378 1322 3 |     SCB_SIZE (OUTPUT_BUFFER) = .SCB_SIZE (OUTPUT_BUFFER) = .SCB[PSM$L_L_MARGIN];
: 379 1323 3 |     SCB[PSM$_COLUMN] = .SCB[PSM$L_L_MARGIN] + 1;
: 380 1324 2 |     END;

```

```

381 1325 2
382 1326 2
383 1327 2 ! Compute columns remaining on this line
384 1328 2
385 1329 2 COLUMNS_LEFT = .SCB[PSM$FORM_WIDTH] -
386 1330 2 .SCB[PSM$RIGHT_MARGIN] - .SCB[PSM$COLUMN] + 1;
387 1331 2
388 1332 2 IF (.SCB[PSM$PRINT_FLAGS] AND (SMBMSG$M_TRUNCATE OR SMBMSG$M_WRAP)) EQL 0
389 1333 2 THEN
390 1334 2 COLUMNS_LEFT = %X '7FFFFFFF';
391 1335 2
392 1336 2
393 1337 2 ! Compute lines remaining on this page
394 1338 2
395 1339 2 LINES_LEFT = .SCB[PSM$FORM_LENGTH] - .SCB[PSM$LINE];
396 1340 2 IF .PRINT_FLAG_ (PAGINATE)
397 1341 2 THEN
398 1342 2 LINES_LEFT = .LINES_LEFT - .SCB[PSM$BOTTOM_MARGIN];
399 1343 2
400 1344 2
401 1345 2 ! Case on the character to be formatted
402 1346 2
403 1347 2 CASE .CHAR FROM PSM$K_CHAR_BS TO PSM$K_CHAR_CR OF
404 1348 2 SET
405 1349 2
406 1350 2 [OUTRANGE]:
407 1351 2 BEGIN
408 1352 2
409 1353 2 ! Exit if starting an escape sequence
410 1354 2
411 1355 2 IF .CHAR EQL PSM$K_CHAR_ESC
412 1356 2 THEN
413 1357 2 RETURN PSM$_ESCAPE;
414 1358 2
415 1359 2
416 1360 2 ! Non-special character, probably we are in wrap/truncate
417 1361 2
418 1362 2 IF .COLUMNS_LEFT LEQ 0
419 1363 2 THEN
420 1364 2 RETURN FORMAT_LINE_OVERFLOW (.SCB, .CHAR)
421 1365 2 ELSE
422 1366 2 INCREMENT_ (SCB[PSM$COLUMN]);
423 1367 2 END;
424 1368 2
425 1369 2
426 1370 2 ! Backspace -- decrement column and output character
427 1371 2
428 1372 2 [PSM$K_CHAR_BS]:
429 1373 2 IF .SCB[PSM$COLUMN] GTRU 1
430 1374 2 THEN
431 1375 2 DECREMENT_ (SCB[PSM$COLUMN]);
432 1376 2
433 1377 2
434 1378 2 ! Horizontal tab -- compute tab expansion and update column. If we
435 1379 2 are expanding tabs then convert to spaces, else allow as imbedded
436 1380 2 format effector
437 1381 2

```

```

: 438 1382 2 [PSMSK CHAR_HT]:
: 439 1383 3 BEGIN
: 440 1384 3 LOCAL SPACE_COUNT;
: 441 1385 3 IF .COLUMNS_LEFT EQL 0
: 442 1386 3 THEN
: 443 1387 3 RETURN FORMAT_LINE_OVERFLOW (.SCB, .CHAR);
: 444 1388 3 SPACE_COUNT = 8 - ((.SCB[PSMSL_COLUMN] - 1) AND %B '111');
: 445 1389 3
: 446 1390 3 IF .SPACE_COUNT GTRU .COLUMNS_LEFT
: 447 1391 3 THEN
: 448 1392 3 SPACE_COUNT = .COLUMNS_LEFT;
: 449 1393 3
: 450 1394 3 SCB[PSMSL_COLUMN] = .SCB[PSMSL_COLUMN] + .SPACE_COUNT;
: 451 1395 3
: 452 1396 3 IF .SCB[PSMSV_EXPAND_TABS]
: 453 1397 3 THEN
: 454 1398 4 BEGIN
: 455 1399 4 SCB_ADDR (OUTPUT_BUFFER) =
: 456 1400 4 CH$FILL (%C, .SPACE_COUNT, .SCB_ADDR (OUTPUT_BUFFER));
: 457 1401 4 SCB_SIZE (OUTPUT_BUFFER) = .SCB_SIZE (OUTPUT_BUFFER) - .SPACE_COUNT;
: 458 1402 4 RETURN $$$_NORMAL;
: 459 1403 3 END;
: 460 1404 3
: 461 1405 2 END;
: 462 1406 2
: 463 1407 2
: 464 1408 2 ! Line feed -- handle double spacing and page overflow
: 465 1409 2
: 466 1410 2 [PSMSK CHAR_LF]:
: 467 1411 3 BEGIN
: 468 1412 3 IF .LINES_LEFT LEQ 0
: 469 1413 4 OR (.PRINT_FLAG_ (DOUBLE_SPACE) AND .LINES_LEFT LEQ 1)
: 470 1414 3 THEN
: 471 1415 3 IF .PRINT_FLAG_ (PAGINATE)
: 472 1416 3 THEN
: 473 1417 3 RETURN FORMAT_CHARACTER (.SCB, PSMSK_CHAR_FF)
: 474 1418 3 ELSE
: 475 1419 4 BEGIN
: 476 1420 4 SCB[PSMSL_LINE] = 0;
: 477 1421 4 SCB[PSMSL_COLUMN] = 0;
: 478 1422 4 INCREMENT_ (SCB[PSMSL_PAGE]);
: 479 1423 4 END
: 480 1424 3 ELSE
: 481 1425 3 INCREMENT_ (SCB[PSMSL_LINE]);
: 482 1426 3
: 483 1427 3 IF .PRINT_FLAG_ (DOUBLE_SPACE)
: 484 1428 3 THEN
: 485 1429 4 BEGIN
: 486 1430 4 WRITE CHAR_ (.CHAR);
: 487 1431 4 INCREMENT_ (SCB[PSMSL_LINE]);
: 488 1432 3 END;
: 489 1433 3
: 490 1434 3 SCB[PSMSL_COLUMN] = 0;
: 491 1435 2 END;
: 492 1436 2
: 493 1437 2
: 494 1438 2 ! Vertical tab -- update line number and allow as imbedded format effector

```

```

495 1439 2
496 1440 2 [PSMSK_CHAR_VT]:
497 1441 2 BEGIN
498 1442 2 IF .LINES_LEFT LEQ 3
499 1443 2 THEN
500 1444 2 IF .PRINT_FLAG_ (PAGINATE)
501 1445 2 THEN
502 1446 2 RETURN FORMAT_CHARACTER (.SCB, PSMSK_CHAR_FF)
503 1447 2 ELSE
504 1448 2 BEGIN
505 1449 2 SCB[PSMSL_LINE] = 0;
506 1450 2 INCREMENT_ (SCB[PSMSL_PAGE]);
507 1451 2 SCB[PSMSL_COLUMN] = 0;
508 1452 2 END;
509 1453 2
510 1454 2 SCB[PSMSL_LINE] = .SCB[PSMSL_LINE] + 4;
511 1455 2 SCB[PSMSL_COLUMN] = 0;
512 1456 2 END;
513 1457 2
514 1458 2
515 1459 2 form feed -- update page number, reset line and column, update
516 1460 2 accounting.
517 1461 2
518 1462 2 [PSMSK_CHAR_FF]:
519 1463 2 BEGIN
520 1464 2 ! Increment the internal page number and, unless searching,
521 1465 2 ! the accounting page number
522 1466 2
523 1467 2 INCREMENT_ (SCB[PSMSL_PAGE]);
524 1468 2 IF NOT .SCB[PSMSV_SUPPRESS_OUTPUT]
525 1469 2 THEN
526 1470 2 INCREMENT_ (ACC_DATA_ (PAGES_PRINTED));
527 1471 2
528 1472 2 ! Reset the column number and update the line number
529 1473 2
530 1474 2 SCB[PSMSL_COLUMN] = 0;
531 1475 2 SCB[PSMSL_LINE] = 0;
532 1476 2
533 1477 2 END;
534 1478 2
535 1479 2
536 1480 2 ! Carriage return -- update column number
537 1481 2
538 1482 2 [PSMSK_CHAR_CR]:
539 1483 2 IF .SCB[PSMSL_COLUMN] NEQ 0
540 1484 2 THEN
541 1485 2 SCB[PSMSL_COLUMN] = 1;
542 1486 2
543 1487 2 TES;
544 1488 2
545 1489 2
546 1490 2 ! Place the character in the output buffer
547 1491 2
548 1492 2 WRITE_CHAR_ (.CHAR);
549 1493 2
550 1494 2
551 1495 2 ! Return success

```



```

: 552
: 553
: 554
: 555
1496 2 !
1497 2 $$$_NORMAL
1498 2
1499 1 END:

```

				OFFC	00000	FORMAT_CHARACTER:				
			56	04	AC	D0	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	1288
			5A	01C8	C6	9E	00006	MOVL	SCB, R6	1302
					6A	D5	0000B	MOVAB	456(R6), R10	
					08	12	0000D	TSTL	(R10)	
					56	DD	0000F	BNEQ	1\$	
		0000V	CF		01	FB	00011	PUSHL	R6	1304
					04	00	00016	CALLS	#1, FORMAT_NEW_PAGE	
			57	01E0	C6	9E	00017	RET		
			32		67	B1	0001C	MOVAB	480(R6), R7	1309
					08	1E	0001F	CMPL	(R7), #50	
			50	00000000G	8F	D0	00021	BGEQU	2\$	
					04	00	00028	MOVL	#PSMS_BUFFEROVF, R0	1311
			58	0194	C6	9E	00029	RET		
					68	D5	0002E	MOVAB	404(R6), R8	1317
					17	12	00030	TSTL	(R8)	
			59	01BC	C6	9E	00032	BNEQ	3\$	
69	20		6E		00	2C	00037	MOVAB	444(R6), R9	1321
					B7	00	0003C	MOVCS	#0, (SP), #32, (R9), @4(R7)	
		04	A7		53	D0	0003E	MOVL	R3, 4(R7)	
			67		69	A2	00042	SUBW2	(R9), (R7)	1322
	68		69		01	C1	00045	ADDL3	#1, (R9), (R8)	1323
	52	008C	C6	0148	C6	C3	00049	SUBL3	328(R6), 140(R6), R2	1330
			52		68	C2	00051	SUBL2	(R8), R2	
					52	D6	00054	INCL	COLUMNS_LEFT	
			54	0204	C6	9E	00056	MOVAB	516(R6), R4	1332
		C0	8F		64	93	0005B	BITB	(R4), #192	
					07	12	0005F	BNEQ	4\$	
			52	7FFFFFFF	8F	D0	00061	MOVL	#2147483647, COLUMNS_LEFT	1334
	53	78	A6		6A	C3	00068	SUBL3	(R10), 120(R6), LINES_LEFT	1339
	04		64		02	E1	0006D	BBC	#2, (R4), 5\$	1340
			53	30	A6	C2	00071	SUBL2	48(R6), LINES_LEFT	1342
			5B	08	AC	9A	00075	MOVZBL	CHAR, R11	1347
	05		08		5B	8F	00079	CASEB	R11, #8, #5	
0092	0066		002B		0021	00	0007D	.WORD	8\$-6\$, -	
			00C7		00B4	00	00085		10\$-6\$, -	
									14\$-6\$, -	
									18\$-6\$, -	
									23\$-6\$, -	
									25\$-6\$, -	
			1B		5B	91	00089	CMPB	R11, #27	1355
					08	12	0008C	BNEQ	7\$	
			50	00000000G	8F	D0	0008E	MOVL	#PSMS_ESCAPE, R0	1357
					04	00	00095	RET		
					52	D5	00096	TSTL	COLUMNS_LEFT	1362
					12	15	00098	BLEQ	11\$	
					68	D6	0009A	INCL	(R8)	1366

				07	11	0009C		BRB	9\$		1347
		01		68	D1	0009E	8\$:	CMPL	(R8), #1		1373
				02	1B	000A1		BLEQU	9\$		
				68	D7	000A3		DECL	(R8)		1375
				00A3	31	000A5	9\$:	BRW	26\$		1373
				52	D5	000A8	10\$:	TSTL	COLUMNS_LEFT		1385
				0A	12	000AA		BNEQ	12\$		
			0840	8F	BB	000AC	11\$:	PUSHR	#*M<R6,R11>		1387
		0000V	CF	02	FB	000B0		CALLS	#2, FORMAT_LINE_OVERFLOW		
				04	000B5			RET			
	50		68	01	C3	000B6	12\$:	SUBL3	#1, (R8), R0		1388
59	50		03	00	EF	000BA		EXTZV	#0, #3, R0, SPACE_COUNT		
	59		08	59	C3	000BF		SUBL3	SPACE_COUNT, #8, SPACE_COUNT		
			52	59	D1	000C3		CMPL	SPACE_COUNT, COLUMNS_LEFT		1390
				03	1B	000C6		BLEQU	13\$		
			59	52	D0	000C8		MOVL	COLUMNS_LEFT, SPACE_COUNT		1392
			68	59	C0	000CB	13\$:	ADDL2	SPACE_COUNT, (R8)		1394
	78	0C	A6	01	E1	000CE		BBC	#1, 12(R6), 26\$		1396
59	20		6E	00	2C	000D3		MOVC5	#0, (SP), #32, SPACE_COUNT, @4(R7)		1400
				04	B7	000D8					
		04	A7	53	D0	000DA		MOVL	R3, 4(R7)		
			67	59	A2	000DE		SUBW2	SPACE_COUNT, (R7)		1401
				71	11	000E1		BRB	27\$		1402
				53	D5	000E3	14\$:	TSTL	LINES_LEFT		1412
				08	15	000E5		BLEQ	15\$		
			13	64	E9	000E7		BLBC	(R4), 16\$		1413
			01	53	D1	000EA		CMPL	LINES_LEFT, #1		
				0E	14	000ED		BGTR	16\$		
	25		64	02	E0	000EF	15\$:	BBS	#2, (R4), 19\$		1415
				6A	D4	000F3		CLRL	(R10)		1420
				68	D4	000F5		CLRL	(R8)		1421
				01EC	C6	000F7		INCL	492(R6)		1422
				02	11	000FB		BRB	17\$		1415
				6A	D6	000FD	16\$:	INCL	(R10)		1425
				64	E9	000FF	17\$:	BLBC	(R4), 22\$		1427
		04	2B	5B	90	00102		MOVB	R11, @4(R7)		1430
				04	A7	00106		INCL	4(R7)		
				67	B7	00109		DECW	(R7)		
				6A	D6	0010B		INCL	(R10)		1431
				1E	11	0010D		BRB	22\$		1434
			03	53	D1	0010F	18\$:	CMPL	LINES_LEFT, #3		1442
				16	14	00112		BGTR	21\$		
	0A		64	02	E1	00114		BBC	#2, (R4), 20\$		1444
				0C	DD	00118	19\$:	PUSHL	#12		1446
				56	DD	0011A		PUSHL	R6		
		FEDF	CF	02	FB	0011C		CALLS	#2, FORMAT_CHARACTER		
				04	00121			RET			
				6A	D4	00122	20\$:	CLRL	(R10)		1449
				01EC	C6	00124		INCL	492(R6)		1450
				68	D4	00128		CLRL	(R8)		1451
			6A	04	C0	0012A	21\$:	ADDL2	#4, (R10)		1454
				68	D4	0012D	22\$:	CLRL	(R8)		1455
				1A	11	0012F		BRB	26\$		1347
				01EC	C6	00131	23\$:	INCL	492(R6)		1467
	04		11	A6	E0	00135		BBS	#6, 17(R6), 24\$		1468
				027E	C6	0013A		INCL	638(R6)		1470
				68	D4	0013E	24\$:	CLRL	(R8)		1474

FORMAT
V04-000

Print Symbiont format routines
FORMAT_CHARACTER - format one character

L 16
16-Sep-1984 02:14:45
14-Sep-1984 12:55:08

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]FORMAT.B32;1

Page 17
(4)

		6A	D4	00140		CLRL	(R10)	:	1475
		07	11	00142		BRB	26\$:	1347
		68	D5	00144	25\$:	TSTL	(R8)	:	1483
		03	13	00146		BEQL	26\$:	
	04	68		01	D0	00148	#1, (R8)	:	1485
		B7		5B	90	0014B	R11, @4(R7)	:	1492
				04	A7	D6	0C14F	:	
				67	B7	00152	INCL	:	
				01	D0	00154	4(R7)	:	
		50		04	00157	27\$:	DECW	:	1499
							(R7)	:	
							MOVL	:	
							#1, R0	:	
							RET	:	

; Routine Size: 344 bytes, Routine Base: CODE + 016F

```

557 1500 1 %SBTTL 'FORMAT_ESCAPE_SEQUENCE -- parse and process escape sequences'
558 1501 1 | Functional Description:
559 1502 1 | This routine parses and processes standard ANSI escape
560 1503 1 | sequences. Escapes sequences are sent to printer in
561 1504 1 | passall mode to avoid modification. OSC (operating system
562 1505 1 | commands) are watched for the special DEC private command
563 1506 1 | that requests inclusion of a module sequence.
564 1507 1 |
565 1508 1 | Formal Parameters:
566 1509 1 |     SCB      : SCB address
567 1510 1 |
568 1511 1 | Implicit Inputs:
569 1512 1 |     none
570 1513 1 |
571 1514 1 | Implicit Outputs:
572 1515 1 |     Data is copied from input to output buffer
573 1516 1 |
574 1517 1 | Returned Value:
575 1518 1 |     none
576 1519 1 |
577 1520 1 | Side Effects:
578 1521 1 |     none
579 1522 1 | --
580 1523 1 ROUTINE FORMAT_ESCAPE_SEQUENCE (
581 1524 1 |     SCB      : REF $BBLOCK
582 1525 1 | ) =
583 1526 2 BEGIN
584 1527 2
585 1528 2 LITERAL
586 1529 2 | FIRST      = 0,      | First byte (escape)
587 1530 2 | SECOND     = 1,      | Second byte (APC, CSI, DCS, OSC, PM, or other)
588 1531 2 | CSI        = 2,      | Control String in progress
589 1532 2 | DCS        = 3,      | Device control string in progress
590 1533 2 | DCS_TRM    = 4,      | DCS String terminator in progress (\)
591 1534 2 | ESC_SEQ    = 5,      | Ordinary escape sequence in progress
592 1535 2 | OSC_V      = 6,      | OSC System name first character (V)
593 1536 2 | OSC_M      = 7,      | OSC System name second character (M)
594 1537 2 | OSC_S      = 8,      | OSC System name third character (S)
595 1538 2 | OSC_SC1    = 9,      | OSC System name terminator (;)
596 1539 2 | OSC_1      = 10,     | OSC Command number (i)
597 1540 2 | OSC_SC2    = 11,     | OSC Command number terminator (;)
598 1541 2 | OSC_ML     = 12,     | Module List in progress
599 1542 2 | OSC_TRM    = 13,     | OSC String terminator in progress (\)
600 1543 2 | PAS_MSG    = 14,     | Pass through message in progress
601 1544 2 | PAS_TRM    = 15,     | Pass through terminator in progress (\)
602 1545 2 |
603 1546 2 |
604 1547 2 LOCAL
605 1548 2 | CHAR
606 1549 2 |
607 1550 2
608 1551 2 WHILE .SCB_SIZE_ (INPUT_RECORD) NEQ 0
609 1552 2 DO
610 1553 3 | BEGIN
611 1554 3 |
612 1555 3 |     IF .SCB[PSM$L_LINE] EQL 0
613 1556 3 |     THEN

```



```

: 671
: 672
: 673
: 674
: 675
: 676
: 677
: 678
: 679
: 680
: 681
: 682
: 683
: 684
: 685
: 686
: 687
: 688
: 689
: 690
: 691
: 692
: 693
: 694
: 695
: 696
: 697
: 698
: 699
: 700
: 701
: 702
: 703
: 704
: 705
: 706
: 707
: 708
: 709
: 710
: 711
: 712
: 713
: 714
: 715
: 716
: 717
: 718
: 719
: 720
: 721
: 722
: 723
: 724
: 725
: 726
: 727

```

```

1614 3      END;
1615 3
1616 3      [DCS_TRM]:
1617 3
1618 3      IF .CHAR EQL PSM$K_CHAR_ST
1619 3      THEN
1620 4          BEGIN
1621 4              SCB_SIZE (OUTPUT_BUFFER) = 1;
1622 4              SCB[PSM$V_ESCAPE_IN_PROGRESS] = 0;
1623 4          END
1624 3      ELSE
1625 3          SCB[PSM$B_ESCAPE_STATE] = DCS;
1626 3
1627 3      [ESC_SEQ]:
1628 3      IF .CHAR GEQU %X '30'      ! ANSI column/row 3/0 through
1629 3      AND .CHAR LEQU %X '7D'      ! ANSI column/row 7/14 inclusive
1630 3      THEN
1631 3          SCB[PSM$V_ESCAPE_IN_PROGRESS] = 0;
1632 3
1633 3      [OSC_V]:
1634 4      BEGIN
1635 4          WRITE_CHAR (PSM$K_CHAR_ESC);
1636 4          WRITE_CHAR (PSM$K_CHAR_OSC);
1637 4          IF .CHAR EQL 'V'
1638 4          THEN
1639 4              SCB[PSM$B_ESCAPE_STATE] = OSC_M
1640 4          ELSE
1641 4              IF .CHAR EQL PSM$K_CHAR_ESC
1642 4              THEN
1643 4                  SCB[PSM$B_ESCAPE_STATE] = PAS_TRM
1644 4              ELSE
1645 4                  SCB[PSM$B_ESCAPE_STATE] = PAS_MSG;
1646 4          END;
1647 3
1648 3      [OSC_M]:
1649 3      IF .CHAR EQL 'M'
1650 3      THEN
1651 3          SCB[PSM$B_ESCAPE_STATE] = OSC_S
1652 3      ELSE
1653 3          IF .CHAR EQL PSM$K_CHAR_ESC
1654 3          THEN
1655 3              SCB[PSM$B_ESCAPE_STATE] = PAS_TRM
1656 3          ELSE
1657 3              SCB[PSM$B_ESCAPE_STATE] = PAS_MSG;
1658 3
1659 3      [OSC_S]:
1660 3      IF .CHAR EQL 'S'
1661 3      THEN
1662 3          SCB[PSM$B_ESCAPE_STATE] = OSC_SC1
1663 3      ELSE
1664 3          IF .CHAR EQL PSM$K_CHAR_ESC
1665 3          THEN
1666 3              SCB[PSM$B_ESCAPE_STATE] = PAS_TRM
1667 3          ELSE
1668 3              SCB[PSM$B_ESCAPE_STATE] = PAS_MSG;
1669 3
1670 3      [OSC_SC1]:

```

```

FORI
V04:
: 10
: 10
: 10
: 10
: 10
: 10
: 10
: 10

```

: R

```

: 728 1671 3      IF .CHAR EQL ':'
: 729 1672 3      THEN
: 730 1673 3      SCB[PSM$B_ESCAPE_STATE] = OSC_1
: 731 1674 3      ELSE
: 732 1675 3      IF .CHAR EQL PSM$K_CHAR_ESC
: 733 1676 3      THEN
: 734 1677 3      SCB[PSM$B_ESCAPE_STATE] = PAS_TRM
: 735 1678 3      ELSE
: 736 1679 3      SCB[PSM$B_ESCAPE_STATE] = PAS_MSG;
: 737 1680 3
: 738 1681 3      [OSC_1]:
: 739 1682 3      IF .CHAR EQL '1'
: 740 1683 3      THEN
: 741 1684 3      SCB[PSM$B_ESCAPE_STATE] = OSC_SC2
: 742 1685 3      ELSE
: 743 1686 3      IF .CHAR EQL PSM$K_CHAR_ESC
: 744 1687 3      THEN
: 745 1688 3      SCB[PSM$B_ESCAPE_STATE] = PAS_TRM
: 746 1689 3      ELSE
: 747 1690 3      SCB[PSM$B_ESCAPE_STATE] = PAS_MSG;
: 748 1691 3
: 749 1692 3      [OSC_SC2]:
: 750 1693 3      IF .CHAR EQL ':'
: 751 1694 3      THEN
: 752 1695 3      SCB[PSM$B_ESCAPE_STATE] = OSC_ML
: 753 1696 3      ELSE
: 754 1697 3      IF .CHAR EQL PSM$K_CHAR_ESC
: 755 1698 3      THEN
: 756 1699 3      SCB[PSM$B_ESCAPE_STATE] = PAS_TRM
: 757 1700 3      ELSE
: 758 1701 3      SCB[PSM$B_ESCAPE_STATE] = PAS_MSG;
: 759 1702 3
: 760 1703 3      [OSC_ML]:
: 761 1704 4      BEGIN
: 762 1705 4      IF .SCB_SIZE_ (OUTPUT_BUFFER) LEQU 1
: 763 1706 4      THEN
: 764 1707 5      BEGIN
: 765 1708 5      PSM$STORE ERRORS (.SCB,
: 766 1709 5      PSM$ OSC_TOLON, 1, .SCB[PSM$L_RECORD_NUMBER]);
: 767 1710 5      RETURN PSM$_BUFFEROVF;
: 768 1711 4      END;
: 769 1712 4      IF .CHAR EQL PSM$K_CHAR_ESC
: 770 1713 4      THEN
: 771 1714 4      SCB[PSM$B_ESCAPE_STATE] = OSC_TRM;
: 772 1715 3      END;
: 773 1716 3
: 774 1717 3      [OSC_TRM]:
: 775 1718 3      IF .CHAR EQL PSM$K_CHAR_ST
: 776 1719 3      THEN
: 777 1720 4      BEGIN
: 778 1721 4      LOCAL DESC : VECTOR [2], IOB : REF $BLOCK;
: 779 1722 4      IOB = .SCB[PSM$A_IOB];
: 780 1723 4      DESC[0] = .SCB_ADDR (OUTPUT_BUFFER) -
: 781 1724 4      .DESC_ADDR (IOB[IOB_Q_BUFFER]) - S_VMS_OSC - 1;
: 782 1725 4      DESC[1] = .DESC_ADDR (IOB[IOB_Q_BUFFER]) + S_VMS_OSC;
: 783 1726 4      PSM$INCLUDE MODULES (SCB, DESC);
: 784 1727 4      SCB_SIZE_ (OUTPUT_BUFFER) = .DESC_SIZE_ (IOB[IOB_Q_BUFFER]);

```

```

: 785 1728 4 SCB_ADDR (OUTPUT_BUFFER) = .DESC_ADDR_ (IOB[IOB_Q_BUFFER]);
: 786 1729 4 SCB[PSM$V_ESCAPE_IN_PROGRESS] = 0;
: 787 1730 4 RETURN PSM$_SUSPEND;
: 788 1731 4 END
: 789 1732 3 ELSE
: 790 1733 4 BEGIN
: 791 1734 4 PSM$STORE_ERRORS (.SCB,
: 792 1735 4 PSM$INVMSOSC, 1, .SCB[PSM$L_RECORD_NUMBER]);
: 793 1736 4 RETURN PSM$_BUFFEROVF;
: 794 1737 3 END;
: 795 1738 3
: 796 1739 3 [PAS_MSG]:
: 797 1740 3 IF .CHAR EQL PSM$K_CHAR_ESC
: 798 1741 3 THEN
: 799 1742 3 SCB[PSM$B_ESCAPE_STATE] = PAS_TRM;
: 800 1743 3
: 801 1744 3 [PAS_TRM]:
: 802 1745 3 IF .CHAR EQL PSM$K_CHAR_ST
: 803 1746 3 THEN
: 804 1747 3 SCB[PSM$V_ESCAPE_IN_PROGRESS] = 0
: 805 1748 3 ELSE
: 806 1749 3 SCB[PSM$B_ESCAPE_STATE] = PAS_MSG;
: 807 1750 3
: 808 1751 3 TES:
: 809 1752 3
: 810 1753 3 WRITE CHAR (.CHAR);
: 811 1754 3 IF .SCB_SIZE_ (OUTPUT_BUFFER) LEQU 0
: 812 1755 3 THEN
: 813 1756 3 RETURN PSM$_BUFFEROVF;
: 814 1757 3
: 815 1758 2 END;
: 816 1759 2
: 817 1760 2 SS$_NORMAL
: 818 1761 2
: 819 1762 1 END;

```

007C 0000 FORMAT_ESCAPE_SEQUENCE:									
					.WORD	Save R2,R3,R4,R5,R6			: 1523
5E		08	C2	00002	SUBL2	#8, SP			: 1551
53	04	AC	D0	00005	MOVL	SCB, R3			
52	0260	C3	9E	00009	MOVAB	608(R3), R2			
		62	B5	0000E	TSTW	(R2)			
		03	12	00010	BNEQ	2\$			
		0227	31	00012	BRW	46\$			
	01C8	C3	D5	00015	TSTL	456(R3)			: 1555
		08	12	00019	BNEQ	3\$			
		53	DD	0001B	PUSHL	R3			: 1557
0000V	CF	01	FB	0001D	CALLS	#1, FORMAT_NEW_PAGE			
		04	00	00022	RET				
		62	B7	00023	DECW	(R2)			: 1559
55	04	B2	9A	00025	MOVZBL	24(R2), CHAR			
	04	A2	D6	00029	INCL	4(R2)			
54	02A3	C3	9E	0002C	MOVAB	675(R3), R4			: 1561

007D	0F	00	64	8F	00031	CASEB	(R4), #0, #15			
00F5	0076	0025	0020		00035	.WORD	5\$-4\$,-			
0125	00B9	00A5	0090		0003D		6\$-4\$,-			
01C6	011B	0111	0103		00045		11\$-4\$,-			
	01BC	0159	013A		0004D		12\$-4\$,-			
							13\$-4\$,-			
							16\$-4\$,-			
							19\$-4\$,-			
							21\$-4\$,-			
							23\$-4\$,-			
							25\$-4\$,-			
							26\$-4\$,-			
							28\$-4\$,-			
							32\$-4\$,-			
							35\$-4\$,-			
							38\$-4\$,-			
							40\$-4\$			
		64	01	90	00055	5\$:	MOVB	#1, (R4)	1565	
			7E	11	00058		BRB	15\$		
	0000005B	8F	55	D1	0005A	6\$:	CMPL	CHAR, #91	1571	
			05	12	00061		BNEQ	7\$		
		64	02	90	00063		MOVB	#2, (R4)	1572	
			70	11	00066		BRB	15\$		
	00000050	8F	55	D1	00068	7\$:	CMPL	CHAR, #80	1574	
			07	12	0006F		BNEQ	8\$		
	01E0	C3	01	B0	00071		MOVW	#1, 480(R3)	1576	
			5D	11	00076		BRB	14\$	1577	
	0000005D	8F	55	D1	00078	8\$:	CMPL	CHAR, #93	1580	
			10	12	0007F		BNEQ	9\$		
		50	01E0	C3	9E	00081	MOVAB	480(R3), R0	1582	
				04	B6	00086	INCW	(R0)		
		64	A0	D7	00088		DECL	4(R0)	1583	
			06	90	0008B		MOVB	#6, (R4)	1584	
			01A3	31	0008E		BRW	45\$	1585	
	0000005E	8F	55	D1	00091	9\$:	CMPL	CHAR, #94	1588	
			0C	1F	00098		BLSSU	10\$		
	0000005F	8F	55	D1	0009A		CMPL	CHAR, #95		
			03	1A	000A1		BGTRU	10\$		
			0164	31	000A3		BRW	43\$		
		64	05	90	000A6	10\$:	MOVB	#5, (R4)	1593	
			2F	11	000A9		BRB	16\$	1594	
		3F	55	D1	000AB	11\$:	CMPL	CHAR, #63	1603	
			78	1B	000AE		BLEQU	20\$		
			30	11	000B0		BRB	17\$	1604	
		50	01AC	C3	D0	000B2	12\$:	MOVL	428(R3), R0	1610
	2C	A0	04	88	000B7		BISB2	#4, 44(R0)		
		1B	55	D1	000BB		CMPL	CHAR, #27	1611	
			76	12	000BE		BNEQ	22\$		
		64	04	90	000C0		MOVB	#4, (R4)	1613	
			7F	11	000C3		BRB	24\$	1561	
	0000005C	8F	55	D1	000C5	13\$:	CMPL	CHAR, #92	1618	
			07	12	000CC		BNEQ	14\$		
	01E0	C3	01	B0	000CE		MOVW	#1, 480(R3)	1621	
			16	11	000D3		BRB	18\$	1622	
		64	03	90	000D5	14\$:	MOVB	#3, (R4)	1625	
			7E	11	000D8	15\$:	BRB	27\$	1618	
		30	55	D1	000DA	16\$:	CMPL	CHAR, #48	1628	

			03	1E	000DD	BGEQU	17\$				
			0080	31	000DF	BRW	29\$				
	0000007D	8F	55	D1	000E2	17\$:	CMPL	CHAR, #125		1629	
			77	1A	000E9		BGTRU	29\$			
			0116	31	000EB	18\$:	BRW	41\$		1631	
		51	01E4	C3	9E	000EF	19\$:	MOVAB	484(R3), R1	1635	
		B1		1B	90	000F3		MOVB	#27, @0(R1)		
		00		61	D6	000F7		INCL	(R1)		
		50		53	D0	000F9		MOVL	R3, R0		
		52		53	D0	000FC		MOVL	R3, R2		
01E0	C0	01E0		01	A3	000FF		SUBW3	#1, 480(R2), 480(R0)		
		00		5D	8F	90	00107	MOVB	#93, @0(R1)	1636	
				61	D6	0010C		INCL	(R1)		
		50		53	D0	0010E		MOVL	R3, R0		
		51		53	D0	00111		MOVL	R3, R1		
01E0	C0	01E0		01	A3	00114		SUBW3	#1, 480(R1), 480(R0)		
		00000056		55	D1	0011C		CMPL	CHAR, #86	1637	
				3F	12	00123		BNEQ	30\$		
		64		07	9C	00125		MCVB	#7, (R4)	1639	
				62	11	00128	20\$:	BRB	34\$		
	0000004D	8F		55	D1	0012A	21\$:	CMPL	CHAR, #77	1649	
				31	12	00131		BNEQ	30\$		
		64		08	90	00133		MOVB	#8, (R4)	1651	
				54	11	00136	22\$:	BRB	34\$		
	00000053	8F		55	D1	00138	23\$:	CMPL	CHAR, #83	1660	
				23	12	0013F		BNEQ	30\$		
		64		09	90	00141		MOVB	#9, (R4)	1662	
				46	11	00144	24\$:	BRB	34\$		
		3B		55	D1	00146	25\$:	CMPL	CHAR, #59	1671	
				19	12	00149		BNEQ	30\$		
		64		0A	90	0014B		MOVB	#10, (R4)	1673	
				3C	11	0014E		BRB	34\$		
		31		55	D1	00150	26\$:	CMPL	CHAR, #49	1682	
				0F	12	00153		BNEQ	30\$		
		64		0B	90	00155		MOVB	#11, (R4)	1684	
				32	11	00158	27\$:	BRB	34\$		
		3B		55	D1	0015A	28\$:	CMPL	CHAR, #59	1693	
				05	12	0015D		BNEQ	30\$		
		64		0C	90	0015F		MOVB	#12, (R4)	1695	
				28	11	00162	29\$:	BRB	34\$		
		1B		55	D1	00164	30\$:	CMPL	CHAR, #27	1697	
				03	12	00167		BNEQ	31\$		
				008A	31	00169		BRW	39\$		
				009B	31	0016C	31\$:	BRW	43\$		
		01	01F0	C3	B1	0016F	32\$:	CMPL	480(R3), #1	1705	
				0E	1A	00174		BGTRU	33\$		
				026C	C3	DD	00176	PUSHL	620(R3)	1709	
				01	DD	0017A		PUSHL	#1	1708	
			00000000G	8F	DD	0017C		PUSHL	#PSMS_OSCTOOLON		
				62	11	00182		BRB	37\$		
		1B		55	D1	00184	33\$:	CMPL	CHAR, #27	1712	
				7F	12	00187		BNEQ	42\$		
		64		0D	90	00189		MOVB	#13, (R4)	1714	
				7F	11	0018C	34\$:	BRB	44\$	1561	
	0000005C	8F		55	D1	0018E	35\$:	CMPL	CHAR, #92	1718	
				43	12	00195		BNEQ	36\$		
		52	01AC	C3	D0	00197		MOVL	428(R3), 10B	1722	

50	01E4	C3	20	A2	C3	0019C	SUBL3	32(IOB), 484(R3), R0	: 1724	
		6E	F7	A0	9E	001A3	MOVAB	-9(R0), DESC	: 1725	
04	AE	20	A2	08	C1	001A7	ADDL3	#8, 32(IOB), DESC+4	: 1726	
				5E	DD	001AD	PUSHL	SP	: 1727	
			04	AC	9F	001AF	PUSHAB	SCB	: 1728	
	00000000G	00		02	FB	001B2	CALLS	#2, PSM\$INCLUDE_MODULES	: 1729	
		50	04	AC	DD	001B9	MOVL	SCB, R0	: 1730	
		56	01E0	C0	9E	001BD	MOVAB	480(R0), R6	: 1731	
		51		52	DD	001C2	MOVI	IOB, R1	: 1732	
		66	1C	A1	B0	001C5	MOVW	28(R1), (R6)	: 1733	
		04	20	A2	DD	001C9	MOVL	32(IOB), 4(R6)	: 1734	
		10		08	8A	001CE	BICB2	#8, 16(R0)	: 1735	
		50	00000000G	8F	DD	001D2	MOVL	#PSM\$_SUSPEND, R0	: 1736	
				04	001D9	RET			: 1737	
			026C	C3	DD	001DA	PUSHL	620(R3)	: 1738	
				01	DD	001DE	PUSHL	#1	: 1739	
			00000000G	8F	DD	001E0	PUSHL	#PSM\$_INVMSOSC	: 1740	
				53	DD	001E6	PUSHL	R3	: 1741	
	00000000G	00		04	FB	001E8	CALLS	#4, PSM\$STORE_ERRORS	: 1742	
				43	11	001EF	BRB	45\$: 1743	
		18		55	D1	001F1	CMPL	CHAR, #27	: 1744	
				17	12	001F4	BNEQ	44\$: 1745	
		64		0F	90	001F6	MOVB	#15, (R4)	: 1746	
				12	11	001F9	BRB	44\$: 1747	
	0000005C	8F		55	D1	001FB	CMPL	CHAR, #92	: 1748	
				06	12	00202	BNEQ	43\$: 1749	
		10	A3	08	8A	00204	BICB2	#8, 16(R3)	: 1750	
				03	11	00208	BRB	44\$: 1751	
		64		0E	90	0020A	MOVB	#14, (R4)	: 1752	
		53	04	AC	DD	0020D	MOVL	SCB, R3	: 1753	
		01E4	D3	55	90	00211	MOVB	CHAR, @484(R3)	: 1754	
			01E4	C3	D6	00216	INCL	484(R3)	: 1755	
		50		53	DD	0021A	MOVL	R3, R0	: 1756	
		51		53	DD	0021D	MOVL	R3, R1	: 1757	
01E0	C0	01E0	C1	01	A3	00220	SUBW3	#1, 480(R1), 480(R0)	: 1758	
			50	53	DD	00228	MOVL	R3, R0	: 1759	
				01E0	C0	B5	0022B	TSTW	480(R0)	: 1760
				03	13	0022F	BEQL	45\$: 1761	
				FDD5	31	00231	BRW	1\$: 1762	
		50	00000000G	8F	DD	00234	MOVL	#PSM\$_BUFFEROVF, R0	: 1763	
				04	0023B	RET			: 1764	
		50		01	DD	0023C	MOVL	#1, R0	: 1765	
				04	0023F	RET			: 1766	

; Routine Size: 576 bytes, Routine Base: CODE + 02C7

```

821 1763 1 XSBTTL 'FORMAT_FIRST_CHARACTER - format first character from an input service'
822 1764 1 Functional Description:
823 1765 1 Processes the first character derived from each input service.
824 1766 1 If the character is a form feed and we are currently at line
825 1767 1 one column one, then it is discarded.
826 1768 1
827 1769 1 Formal Parameters:
828 1770 1 SCB : SCB address
829 1771 1
830 1772 1 Implicit Inputs:
831 1773 1 none
832 1774 1
833 1775 1 Implicit Outputs:
834 1776 1 none
835 1777 1
836 1778 1 Returned Value:
837 1779 1 none
838 1780 1
839 1781 1 Side Effects:
840 1782 1 The first character may be discarded.
841 1783 1 --
842 1784 1 ROUTINE FORMAT_FIRST_CHARACTER (
843 1785 1 SCB : REF $BLOCK
844 1786 1 ) =
845 1787 2 BEGIN
846 1788 2
847 1789 2
848 1790 2 ! If not at line/column 1 then return
849 1791 2 !
850 1792 2 IF .SCB[PSM$L_COLUMN] GTRU 0
851 1793 2 OR .SCB[PSM$L_LINE] GTRU 1
852 1794 2 THEN
853 1795 2 RETURN SSS_NORMAL;
854 1796 2
855 1797 2
856 1798 2 ! Check the prefix carriage control for any data
857 1799 2 !
858 1800 2 IF .SCB[PSM$B_PREFIX_COUNT] GTRU 0
859 1801 2 THEN
860 1802 3 BEGIN
861 1803 3 IF .SCB[PSM$B_PREFIX_CHAR] EQL PSM$K_CHAR_FF
862 1804 3 THEN
863 1805 3 DECREMENT (.SCB[PSM$B_PREFIX_COUNT]);
864 1806 3 RETURN SSS_NORMAL;
865 1807 2 END;
866 1808 2
867 1809 2
868 1810 2 ! Check the data record for any data
869 1811 2 !
870 1812 2 IF .SCB_SIZE_ (INPUT_RECORD) GTRU 0
871 1813 2 THEN
872 1814 3 BEGIN
873 1815 3 LOCAL CHAR;
874 1816 3 CHAR = CH$RCHAR (.SCB_ADDR_ (INPUT_RECORD));
875 1817 3 IF .CHAR EQL PSM$K_CHAR_FF
876 1818 3 THEN
877 1819 3 READ_CHAR_ ;

```

```

: 878 1820 3 RETURN SSS_NORMAL;
: 879 1821 2 END;
: 880 1822 2
: 881 1823 2
: 882 1824 2 ! Check the postfix carriage control for any data
: 883 1825 2
: 884 1826 2 IF .SCB[PSMSB_POSTFIX_COUNT] GTRU 0
: 885 1827 2 THEN
: 886 1828 3 BEGIN
: 887 1829 3 IF .SCB[PSMSB_POSTFIX_CHAR] EQL PSMSK_CHAR_FF
: 888 1830 3 THEN
: 889 1831 3 DECREMENT (SCB[PSMSB_POSTFIX_COUNT]);
: 890 1832 3 RETURN SSS_NORMAL;
: 891 1833 2 END;
: 892 1834 2
: 893 1835 2 SSS_NORMAL
: 894 1836 2
: 895 1837 1 END;

```

		0004 0000		FORMAT_FIRST_CHARACTER:			
50	04	AC	D0	00002	.WORD	Save R2	1784
	0194	C0	D5	00006	MOVL	SCB, R0	1792
		42	12	0000A	TSTL	404(R0)	
01	01C8	C0	D1	0000C	BNEQ	4\$	1793
		3B	1A	00011	CMPL	456(R0), #1	
51	0278	C0	9E	00013	BGTRU	4\$	1800
		61	95	00018	MOVAB	632(R0), R1	
		07	13	0001A	TSTB	(R1)	
0C	0279	C0	91	0001C	BEQL	1\$	1803
		27	11	00021	CMPB	633(R0), #12	
51	0260	C0	9E	00023	BRB	3\$	1812
		61	B5	00028	MOVAB	608(R0), R1	
		10	13	0002A	TSTW	(R1)	
52	04	B1	9A	0002C	BEQL	2\$	1816
0C		52	D1	00030	MOVZBL	24(R1), CHAR	1817
		19	12	00033	CMPL	CHAR, #12	
		61	B7	00035	BNEQ	4\$	1818
	04	A1	D6	00037	DECB	(R1)	
		12	11	0003A	INCL	4(R1)	
51	027A	C0	9E	0003C	BRB	4\$	1820
		61	95	00041	MOVAB	634(R0), R1	1826
		09	13	00043	TSTB	(R1)	
0C	027B	C0	91	00045	BEQL	4\$	1829
		02	12	0004A	CMPB	635(R0), #12	
		61	97	0004C	BNEQ	4\$	1831
50		01	D0	0004E	DECB	(R1)	1837
		04	00	00051	MOVL	#1, R0	
					RET		

; Routine Size: 82 bytes, Routine Base: CODE + 0507

```

: 897 1838 1 %SBTTL 'FORMAT_LINE_NUMBER -- print line numbers'
: 898 1839 1 | Functional Description:
: 899 1840 1 | This routine formats the line numbers for sequenced files.
: 900 1841 1 |
: 901 1842 1 | Formal Parameters:
: 902 1843 1 |     SCB      : SCB address
: 903 1844 1 |
: 904 1845 1 | Implicit Inputs:
: 905 1846 1 |     none
: 906 1847 1 |
: 907 1848 1 | Implicit Outputs:
: 908 1849 1 |     The binary line number is converted to an ASCII string and
: 909 1850 1 |     placed in the output buffer.
: 910 1851 1 |
: 911 1852 1 | Returned Value:
: 912 1853 1 |     $$$_NORMAL
: 913 1854 1 |
: 914 1855 1 | Side Effects:
: 915 1856 1 |     none
: 916 1857 1 | --
: 917 1858 1 ROUTINE FORMAT_LINE_NUMBER (
: 918 1859 1     SCB      : REF $BBLOCK
: 919 1860 1     )
: 920 1861 2 BEGIN
: 921 1862 2
: 922 1863 2 | If starting a new page then report that
: 923 1864 2 |
: 924 1865 2 IF .SCB[PSM$L_LINE] EQL 0
: 925 1866 2 THEN
: 926 1867 2     RETURN FORMAT_NEW_PAGE (.SCB);
: 927 1868 2
: 928 1869 2
: 929 1870 2 | If column is zero we are starting a new line -- however new lines
: 930 1871 2 | currently receive no special processing
: 931 1872 2 |
: 932 1873 2 IF .SCB[PSM$L_COLUMN] EQL 0
: 933 1874 2 THEN
: 934 1875 2     INCREMENT_ (SCB[PSM$L_COLUMN]);
: 935 1876 2
: 936 1877 2
: 937 1878 2 | Insure adequate room in the output buffer
: 938 1879 2 |
: 939 1880 2 IF .SCB_SIZE_ (OUTPUT_BUFFER) LSSU 8
: 940 1881 2 THEN
: 941 1882 2     RETURN PSM$_BUFFEROVF;
: 942 1883 2
: 943 1884 2
: 944 1885 2 | Expand the line number to text
: 945 1886 2 |
: 946 P 1887 2 SIGNAL_IF_ERROR ($FAO (
: 947 P 1888 2     $DESCRIPTOR ('!8<!5UW!>'),
: 948 P 1889 2     0,
: 949 P 1890 2     SCB[PSM$Q_OUTPUT_BUFFER],
: 950 1891 2     .SCB[PSM$E_RECORD_HEADER]));
: 951 1892 2
: 952 1893 2
: 953 1894 2 | Update the output buffer descriptor and column number

```

```

! - format string
! - don't return length
! - outbuf desc
! - sequence number

```

```

: 954 1895 2 !
: 955 1896 2 SCB_SIZE_ (OUTPUT_BUFFER) = .SCB_SIZE_ (OUTPUT_BUFFER) - 8;
: 956 1897 2 SCB_ADDR_ (OUTPUT_BUFFER) = .SCB_ADDR_ (OUTPUT_BUFFER) + 8;
: 957 1898 2 SCB[PSM$C_COLUMN] = .SCB[PSM$L_COLUMN] + 8;
: 958 1899 2
: 959 1900 2 SSS_NORMAL
: 960 1901 2
: 961 1902 1 END;

```

```

3E 21 57 55 35 21 3C 38 21 00559 P.AAC: .ASCII \.8<!5UW!>\
                                00562 .BLKB 2
                                00000009 00564 P.AAB: .LONG 9
                                00000000' 00568 .ADDRESS P.AAC
                                .EXTRN SYSSFAO

```

001C 0000 FORMAT_LINE NUMBER:

```

                                .WORD Save R2,R3,R4
53 04 AC D0 00002 MOVL SCB, R3 : 1858
01C8 C3 D5 00006 TSTL 456(R3) : 1865
08 12 0000A BNEQ 1$
0000V CF 53 DD 0000C PUSHL R3 : 1867
01 FB 0000E CALLS #1, FORMAT_NEW_PAGE
04 00013 RET
54 0194 C3 9E 00014 1$: MOVAB 404(R3), R4 : 1873
64 D5 00019 TSTL (R4)
02 12 0001B BNEQ 2$
64 D6 0001D INCL (R4) : 1875
52 01E0 C3 9E 0001F 2$: MOVAB 480(R3), R2 : 1880
08 62 B1 00024 CMPW (R2), #8
08 1E 00027 BGEQU 3$
50 00000000G 8F D0 00029 MOVL #PSM$_BUFFEROVF, R0 : 1882
04 00030 RET
0268 C3 DD 00031 3$: PUSHL 616(R3) : 1891
52 DD 00035 PUSHL R2
7E D4 00037 CLRL -(SP)
BC AF 9F 00039 PUSHAB P.AAB
00000000G 00 04 FB 0003C CALLS #4, SYSSFAO
53 50 D0 00043 MOVL R0, STATUS
09 53 E8 00046 BLBS STATUS, 4$
53 DD 00049 PUSHL STATUS
00000000G 00 01 FB 0004B CALLS #1, LIB$SIGNAL
62 08 A2 00052 4$: SUBW2 #8, (R2) : 1896
04 A2 08 C0 00055 ADDL2 #8, 4(R2) : 1897
64 08 C0 00059 ADDL2 #8, (R4) : 1898
50 01 D0 0005C MOVL #1, R0 : 1902
04 0005F RET

```

; Routine Size: 96 bytes, Routine Base: CODE + 056C

; R0

```

: 963 1903 1 %SBTTL 'FORMAT_LINE_OVERFLOW - handle line overflow'
: 964 1904 1 Functional Description:
: 965 1905 1 This routine is called when the right margin is exceeded.
: 966 1906 1 It truncates the line, wraps the line, or allows the overflow
: 967 1907 1 depending on the settings of /WRAP and /TRUNCATE as specified
: 968 1908 1 with DEFINE/FORM
: 969 1909 1
: 970 1910 1 Formal Parameters:
: 971 1911 1 SCB : SCB address
: 972 1912 1
: 973 1913 1 Implicit Inputs:
: 974 1914 1 none
: 975 1915 1
: 976 1916 1 Implicit Outputs:
: 977 1917 1 none
: 978 1918 1
: 979 1919 1 Returned Value:
: 980 1920 1 none
: 981 1921 1
: 982 1922 1 Side Effects:
: 983 1923 1 none
: 984 1924 1 --
: 985 1925 1 ROUTINE FORMAT_LINE_OVERFLOW (
: 986 1926 1 SCB : REF $BBLOCK,
: 987 1927 1 CHAR : BYTE
: 988 1928 1 ) =
: 989 1929 2 BEGIN
: 990 1930 2
: 991 1931 2 ! If we are wrapping then insert a <CR><LF> pair in the output buffer
: 992 1932 2 !
: 993 1933 2 IF .PRINT_FLAG_ (WRAP)
: 994 1934 2 THEN
: 995 1935 3 BEGIN
: 996 1936 3 FORMAT_CHARACTER (.SCB, PSM$K_CHAR_CR);
: 997 1937 3 FORMAT_CHARACTER (.SCB, PSM$K_CHAR_LF);
: 998 1938 3 RETURN (FORMAT_CHARACTER (.SCB, .CHAR));
: 999 1939 2 END;
1000 1940 2
1001 1941 2 ! Else do nothing special -- truncate handled elsewhere
1002 1942 2
1003 1943 2 SSS_NORMAL
1004 1944 2
1005 1945 1 END;

```

000C 00000 FORMAT_LINE_OVERFLOW:

```

53 FB9D CF 9E 00002 .WORD Save R2,R3
52 04 AC D0 00007 MOVAB FORMAT_CHARACTER, R3
0204 C2 95 0000B TSTB 516(R2)
18 18 0000F BGEQ 1$
06 DD 00011 PUSHL #13
52 DD 00013 PUSHL R2
63 02 FB 00015 CALLS #2, FORMAT_CHARACTER

```

: 1925
: 1933
: 1936
:


```

1007 1946 1 %SBTTL 'FORMAT_NEW_PAGE - handles the start of a new page'
1008 1947 1 ! Functional Description:
1009 1948 1 ! This routine is called when an attempt to place data into
1010 1949 1 ! a page never written to before is detected. Top margins
1011 1950 1 ! are handled by inserting <LF>'s as necessary. An LP11
1012 1951 1 ! controller problem with dropping line feeds that follow
1013 1952 1 ! form feeds is also handled.
1014 1953 1
1015 1954 1 ! Formal Parameters:
1016 1955 1 ! SCB : SCB address
1017 1956 1
1018 1957 1 ! Implicit Inputs:
1019 1958 1 ! none
1020 1959 1
1021 1960 1 ! Implicit Outputs:
1022 1961 1 ! none
1023 1962 1
1024 1963 1 ! Returned Value:
1025 1964 1 ! none
1026 1965 1
1027 1966 1 ! Side Effects:
1028 1967 1 ! none
1029 1968 1 ! --
1030 1969 1 ROUTINE FORMAT_NEW_PAGE (
1031 1970 1 ! SCB : REF $BLOCK
1032 1971 1 ! )
1033 1972 2 BEGIN
1034 1973 2
1035 1974 2 ! Assure adequate room for the top margin sequence
1036 1975 2 !
1037 1976 2 IF .SCB_SIZE_ (OUTPUT_BUFFER) LSSU .SCB[PSM$L_T_MARGIN] + 5
1038 1977 2 THEN
1039 1978 2 ! RETURN PSM$_BUFFEROVF;
1040 1979 2
1041 1980 2
1042 1981 2 ! Output a space, carriage return sequence to workaround
1043 1982 2 ! LP111 controller bug that drops line feeds that follow
1044 1983 2 ! form feeds too quickly
1045 1984 2 !
1046 1985 2 IF .SCB[PSM$L_T_MARGIN] NEQU 0
1047 1986 2 THEN
1048 1987 3 ! BEGIN
1049 1988 3 ! WRITE_CHAR_ (%C ' ');
1050 1989 3 ! WRITE_CHAR_ (PSM$K_CHAR_CR);
1051 1990 2 ! END;
1052 1991 2
1053 1992 2
1054 1993 2 ! Place TOP_MARGIN line feeds in output buffer
1055 1994 2 !
1056 1995 2 SCB_ADDR_ (OUTPUT_BUFFER) = CH$FILL (PSM$K_CHAR_LF,
1057 1996 2 ! .SCB[PSM$L_T_MARGIN], .SCB_ADDR_ (OUTPUT_BUFFER));
1058 1997 2 SCB_SIZE_ (OUTPUT_BUFFER) =
1059 1998 2 ! .SCB_SIZE_ (OUTPUT_BUFFER) - .SCB[PSM$L_T_MARGIN];
1060 1999 2
1061 2000 2
1062 2001 2 ! Set the current line to top margin plus 1
1063 2002 2 !

```

```
: 1064 2003 2 SCB[PSM$L_LINE] = .SCB[PSM$L_T_MARGIN] + 1;  
: 1065 2004 2  
: 1066 2005 2  
: 1067 2006 2 ! Return new page status to trigger page headers  
: 1068 2007 2 !  
: 1069 2008 2 PSM$_NEWPAGE  
: 1070 2009 2  
: 1071 2010 1 END;
```

01FC 00000 FORMAT_NEW_PAGE:

			57	04	AC	D0	00002		WORD	Save R2,R3,R4,R5,R6,R7,R8	1969
			56	01E0	C7	9E	00006		MOVL	SCB, R7	1976
			58	0230	C7	9E	0000B		MOVAB	480(R7), R6	
50		50	68		05	C1	00010		MOVAB	560(R7), R8	
		66	10		00	ED	00014		ADDL3	#5, (R8), R0	
					08	1E	00019		CMPZV	#0, #16, (R6), R0	
			50	00000000G	8F	D0	0001B		BGEQU	1\$	
						04	00022		MOVL	#PSM\$_BUFFEROVF, R0	1978
					68	D5	00023	1\$:	RET	(R8)	1985
					12	13	00025		TSTL	2\$	
			04	B6		20	90	00027	BEQL	#32, @4(R6)	1988
					04	A6	D6	0002B	MOVB	4(R6)	
					66	B7	0002E		INCL	(R6)	
			04	B6		0D	90	00030	D'CW	(R6)	
					04	A6	D6	00034	MOVB	#13, @4(R6)	1989
					66	B7	00037		INCL	4(R6)	
68		0A			6E	0C	2C	00039	DECW	(R6)	
					04	B6		0003E	MOVC5	#0, (SP), #10, (R8), @4(R6)	1996
			04	A6		53	D0	00040	MOVL	R3, 4(R6)	
					66	68	A2	00044	SUBW2	(R8), (R6)	1998
					68	01	C1	00047	ADDL3	#1, (R8), 456(R7)	2003
	01C8	C7			50	00000000G	8F	D0	MOVL	#PSM\$_NEWPAGE, R0	2010
						04	00054		RET		

: Routine Size: 85 bytes, Routine Base: CODE + 05F9

FORMAT
V04-000

Print Symbiont format routines
FORMAT_NEW_PAGE - handles the start of a new pa

D 2
16-Sep-1984 02:14:45
14-Sep-1984 12:55:08

VAX-11 Bliss-32 V4.0-742
[PRTSMB.SRC]FORMAT.B32;1

Page 34
(10)

INPL
V04-

: 1073
: 1074
2011 1 END
2012 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
CODE	1614	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	33	0	1000	00:01.7

COMMAND QUALIFIERS

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

: Size: 1594 code + 20 data bytes
: Run Time: 00:44.3
: Elapsed Time: 01:58.5
: Lines/CPU Min: 2726
: Lexemes/CPU-Min: 31368
: Memory Used: 326 pages
: Compilation Complete

0309 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

(This section contains a dense grid of faint, illegible text and graphics, likely representing a large software manual or a collection of technical documents. Visible labels include:)

- BANNER LIS
- PLWRITE LIS
- SMBREQ REQ
- PLIVECTOR LIS
- SMBRSUSHR MAP
- PLTODATA LIS
- SMBDEF SOL
- PRTSMB
- DISPATCH LIS
- PLISTRING LIS
- PRTSMB MAP
- FORMAT LIS

0310 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

