





```

SSSSSSSS  NN      NN  DDDDDDDD  EEEEEEEEE  RRRRRRR  LL
SSSSSSSS  NN      NN  DDDDDDDD  EEEEEEEEE  RRRRRRR  LL
SS        NN      NN  DD      DD  EE        RR      RR  LL
SS        NN      NN  DD      DD  EE        RR      RR  LL
SS        NNNN    NN  DD      DD  EE        RR      RR  LL
SS        NNNN    NN  DD      DD  EE        RR      RR  LL
          SSSSSS  NN  NN  NN  DD      DD  EEEEEEEE  RRRRRRR  LL
          SSSSSS  NN  NN  NN  DD      DD  EEEEEEEE  RRRRRRR  LL
          SS      NN      NNNN  DD      DD  EE        RR  RR  LL
          SS      NN      NNNN  DD      DD  EE        RR  RR  LL
          SS      NN      NN  DD      DD  EE        RR  RR  LL
          SS      NN      NN  DD      DD  EE        RR  RR  LL
          SS      NN      NN  DD      DD  EE        RR  RR  LL
SSSSSSSS  NN      NN  DDDDDDDD  EEEEEEEEE  RR      RR  LLLLLLLLLL
SSSSSSSS  NN      NN  DDDDDDDD  EEEEEEEEE  RR      RR  LLLLLLLLLL
          .....
          .....
          .....
          .....

```

```

LL          IIIII  SSSSSSS
LL          IIIII  SSSSSSS
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 IIIII  SSSSSSS
LLLLLLLLLL IIIII  SSSSSSS

```

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

```

0001 0 MODULE SNDRERL (
0002 0
0003 0     LANGUAGE (BLISS32),
0004 0     IDENT = 'V04-000'
0005 1 ) =
0006 1 BEGIN
0007 1
0008 1 *****
0009 1 *
0010 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 *  ALL RIGHTS RESERVED.
0013 1 *
0014 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 *  TRANSFERRED.
0020 1 *
0021 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 *  CORPORATION.
0024 1 *
0025 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1 **
0032 1
0033 1 FACILITY:  F11ACP Structure Level 1
0034 1
0035 1 ABSTRACT:
0036 1
0037 1     This routine sends a message to the error logger to inform it of a
0038 1     volume mount or dismount.
0039 1
0040 1 ENVIRONMENT:
0041 1
0042 1     STARLET operating system, including privileged system services
0043 1     and internal exec routines.
0044 1
0045 1 --
0046 1
0047 1
0048 1 AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  23-Jun-1978  18:47
0049 1
0050 1 MODIFIED BY:
0051 1
0052 1     V03-001  LMP0221      L. Mark Pilant,      27-Mar-1984  14:46
0053 1     Change UCBSL_OWNUIC to ORBSL_OWNER and UCBSW_VPROT to
0054 1     ORBSW_PROT.
0055 1
0056 1     A0101  ACG0113      Andrew C. Goldstein,  15-Jan-1980  22:58
0057 1     Fill in volume set data in error log message

```

```
.. 58 0058 1 |  
.. 59 0059 1 |  
.. 60 0060 1 | A0100 ACG00001 Andrew C. Goldstein, 10-Oct-1978 20:03  
.. 61 0061 1 | Previous revision history moved to F11A.REV  
.. 62 0062 1 |  
.. 63 0063 1 |  
.. 64 0064 1 | LIBRARY 'SYSSLIBRARY:LIB.L32';  
.. 65 0065 1 | REQUIRE 'SRCS:FCPDEF.B32';  
.. 66 1056 1 |  
.. 67 1057 1 |  
.. 68 1058 1 | This routine is called at raised IPL and must be locked into the working set.  
.. 69 1059 1 |  
.. 70 1060 1 |  
.. 71 1061 1 | LOCK_CODE;
```

```

73 1062 1 GLOBAL ROUTINE SEND_ERRLOG (MODE, UCB) =
74 1063 1
75 1064 1 |++
76 1065 1 |
77 1066 1 | FUNCTIONAL DESCRIPTION:
78 1067 1 |
79 1068 1 |     This routine sends a message to the error logger to inform it of a
80 1069 1 |     volume mount or dismount.
81 1070 1 |
82 1071 1 |
83 1072 1 | CALLING SEQUENCE:
84 1073 1 |     SEND_ERRLOG (ARG1, ARG2)
85 1074 1 |
86 1075 1 | INPUT PARAMETERS:
87 1076 1 |     ARG1: 1 to signal mount
88 1077 1 |           0 to signal dismount
89 1078 1 |     ARG3: address of UCB
90 1079 1 |
91 1080 1 | IMPLICIT INPUTS:
92 1081 1 |     NONE
93 1082 1 |
94 1083 1 | OUTPUT PARAMETERS:
95 1084 1 |     NONE
96 1085 1 |
97 1086 1 | IMPLICIT OUTPUTS:
98 1087 1 |     NONE
99 1088 1 |
100 1089 1 | ROUTINE VALUE:
101 1090 1 |     1
102 1091 1 |
103 1092 1 | SIDE EFFECTS:
104 1093 1 |     Message sent to error logger
105 1094 1 |
106 1095 1 | --
107 1096 1 |
108 1097 2 BEGIN
109 1098 2
110 1099 2 MAP
111 1100 2     UCB           : REF BBLOCK;    ! UCB argument
112 1101 2
113 1102 2 LINKAGE
114 1103 2     L_ERL_ALLOC   = JSB (REGISTER = 1) :
115 1104 2                   GLOBAL (ADDRESS = 2)
116 1105 2                   NOTUSED (3, 4, 5, 6, 7, 8, 9, 10, 11);
117 1106 2
118 1107 2     L_ERL_RELEASE = JSB (REGISTER = 2) :
119 1108 2                   NOTUSED (3, 4, 5, 6, 7, 8, 9, 10, 11);
120 1109 2
121 1110 2 LOCAL
122 1111 2     ORB           : REF BBLOCK,    ! local address of ORB
123 1112 2     MSG_BUFFER   : REF BBLOCK;    ! other buffer pointer to dodge MOVC
124 1113 2
125 1114 2 EXTERNAL ROUTINE
126 1115 2     ERL$ALLOCEMB  : L_ERL_ALLOC ADDRESSING_MODE (GENERAL);
127 1116 2                   ! allocate error log buffer
128 1117 2     ERL$RELEASEMB : L_ERL_RELEASE ADDRESSING_MODE (GENERAL);
129 1118 2                   ! release error log buffer

```

```

130 1119 2
131 1120 2
132 1121 2 ! Allocate an error log buffer. If this fails, forget it.
133 1122 2 !
134 1123 2
135 1124 2 BEGIN
136 1125 2 GLOBAL REGISTER
137 1126 2 ADDRESS = 2 : REF BBLOCK; ! pointer to error log buffer
138 1127 2
139 1128 2 IF NOT ERL$ALLOCEMB (EMBSK_VM_LENGTH)
140 1129 2 THEN RETURN 1;
141 1130 2 MSG_BUFFER = .ADDRESS;
142 1131 2 END;
143 1132 2
144 1133 2 ! Now fill in the message buffer.
145 1134 2 !
146 1135 2
147 1136 2 IF .MODE
148 1137 2 THEN MSG_BUFFER[EMBSW_VM_ENTRY] = EMBSK_VM
149 1138 2 ELSE MSG_BUFFER[EMBSW_VM_ENTRY] = EMBSK_VD; ! log entry type
150 1139 2
151 1140 2 ORB = .UCB[UCBSL_ORB];
152 1141 2 MSG_BUFFER[EMBSL_VM_OWNUIC] = .ORB[ORB$OWNER];
153 1142 2 MSG_BUFFER[EMBSL_VM_ERRCNT] = .UCB[UCBSW_ERRCNT];
154 1143 2 MSG_BUFFER[EMBSL_VM_OPRCNT] = .UCB[UCBSL_OPCNT];
155 1144 2 MSG_BUFFER[EMBSW_VM_UNIT] = .UCB[UCBSW_UNIT];
156 1145 2
157 1146 2 MSG_BUFFER[EMBSW_VM_VOLNUM] = 0;
158 1147 2 MSG_BUFFER[EMBSW_VM_NUMSET] = 0;
159 1148 2
160 1149 2 CH$MOVE (. (BBLOCK [.UCB[UCBSL_DDB], DDB$T_NAME]) <0,8> + 1,
161 1150 2 BBLOCK [.UCB[UCBSL_DDB], DDB$T_NAME],
162 1151 2 MSG_BUFFER[EMBSB_VM_NAMLANG]);
163 1152 2
164 1153 2 IF .BBLOCK[UCB[UCBSL_DEVCHAR], DEV$V FOR]
165 1154 2 OR NOT .BBLOCK[UCB[UCBSL_DEVCHAR], DEV$V_SQD]
166 1155 2 THEN
167 1156 2 BEGIN
168 1157 2 LOCAL
169 1158 2 VCB : REF BBLOCK, ! address of volume control block
170 1159 2 RVT : REF BBLOCK; ! address of relative volume table
171 1160 2
172 1161 2 VCB = .UCB[UCBSL_VCB];
173 1162 2 IF .VCB[VCBSW_RVN] NEQ 0
174 1163 2 THEN
175 1164 2 BEGIN
176 1165 2 RVT = .VCB[VCBSL_RVT];
177 1166 2 MSG_BUFFER[EMBSW_VM_VOLNUM] = .VCB[VCBSW_RVN];
178 1167 2 MSG_BUFFER[EMBSW_VM_NUMSET] = .RVT[RVT$B_NVOLS];
179 1168 2 END;
180 1169 2 CH$MOVE (VCB$S_VOLNAME,
181 1170 2 BBLOCK [.UCB[UCBSL_VCB], VCB$T_VOLNAME],
182 1171 2 MSG_BUFFER[EMBSB_VM_LABEL]);
183 1172 2 END
184 1173 2 ELSE
185 1174 2 BEGIN
186 1175 2 LOCAL

```

```

187      1176      3      MVL          : REF BBLOCK,      : magtape volume labels
188      1177      3      MVL_ENTRY   : REF BBLOCK,      : address of label entry
189      1178      3      RUN          :          : relative unit number
190      1179      3      RVT          : REF BBLOCK,      : relative volume table
191      1180      3      UCBLIST      : REF VECTOR,      : address of UCB list
192      1181      3      VCB          : REF BBLOCK;      : volume control block
193      1182      3      VCB = .UCB[UCBSL_VCB];
194      1183      3      RVT = .VCB[VCBSL_RVT];
195      1184      3      UCBLIST = RVT[RVT$L_UCBLST];
196      1185      3      MVL = .VCB[VCBSL_MVC];
197      1186      3      MSG_BUFFER[EMBSW_VM_NUMSET] = .MVL[MVL$B_NVOLS];      : no of volumes in vol set known
198      1187      3      CH$FILL(' ',VCBS$VOLNAME,MSG_BUFFER[EMB$T_VM_LABEL]);
199      1188      3      INCR I FROM 0 TO .RVT[RVT$B_NVOLS] - 1 DO
200      1189      4      BEGIN
201      1190      4      RUN = .I;
202      1191      4      IF .UCBLIST[I] EQL .UCB THEN EXITLOOP;
203      1192      3      END;
204      1193      3      MVL_ENTRY = .MVL + MVL$K_FIXLEN;
205      1194      3      INCR I FROM 0 TO .MVL[MV$B_NVOLS] - 1 DO
206      1195      4      BEGIN
207      1196      4      IF .MVL_ENTRY[MVL$B_RVN] EQL .RUN
208      1197      4      AND .MVL_ENTRY[MVL$B_MOUNTED]
209      1198      4      THEN
210      1199      5      BEGIN
211      1200      5      MSG_BUFFER[EMBSW_VM_VOLNUM] = .I + 1;
212      1201      5      CH$COPY(MVL$S_VO[LBC],MVL_ENTRY[MVL$T_VOLLBL],' ',
213      1202      5      VCB$S_VOLNAME,MSG_BUFFER[EMB$T_VM_LABEL]);
214      1203      5      EXITLOOP;
215      1204      4      END;
216      1205      4      MVL_ENTRY = .MVL_ENTRY + MVL$K_LENGTH;
217      1206      3      END;
218      1207      2      END;
219      1208      2      ! Finally release the buffer and make the entry.
220      1209      2      !
221      1210      2      !
222      1211      2      !
223      1212      2      ERL$RELEASEMB (.MSG_BUFFER);
224      1213      2      !
225      1214      2      RETURN 1;
226      1215      2      !
227      1216      1      END;

```

! end of routine SEND\_ERRLOG

```

.TITLE SNDRRL
.IDENT \V04-000\

.EXTRN ERL$ALLOCEMB, ERL$RELEASEMB

.PSECT $LOCKEDC1$,NOWRT,2

.ENTRY SEND_ERRLOG, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 1062
R10
MOVL #62, R1 ; 1128
JSB ERL$ALLOCEMB
BLBS R0, 1$
BRW 14$
MOVL ADDRESS, MSG_BUFFER ; 1130

```

```

07FC 00000
51 00000000G 3E DO 00002
03 50 E8 0000B
00F4 31 0000E
59 52 DO 00011 1$

```

|    |    |    |    |      |    |       |       |                                    |   |      |
|----|----|----|----|------|----|-------|-------|------------------------------------|---|------|
|    |    | 07 | 04 | AC   | E9 | 00014 | BLBC  | MODE, 2\$                          | 1136                                      |      |
|    |    | 04 | A9 | 40   | 8F | 9B    | 00018 | MOVZBW                             | #64, 4(MSG_BUFFER)                        | 1137 |
|    |    |    |    |      | 05 | 11    | 0001D | BRB                                | 3\$                                       |      |
|    |    | 04 | A9 | 41   | 8F | 9B    | 0001F | MOVZBW                             | #65, 4(MSG_BUFFER)                        | 1138 |
|    |    |    |    |      | 08 | AC    | 00024 | MOVL                               | UCB, R0                                   | 1140 |
|    |    | 50 | 50 | 1C   | A0 | D0    | 00028 | MOVL                               | 28(R0), ORB                               |      |
|    |    | 10 | A9 |      | 60 | D0    | 0002C | MOVL                               | (ORB), 16(MSG_BUFFER)                     | 1141 |
|    |    |    |    |      | 08 | AC    | 00030 | MOVL                               | UCB, R0                                   | 1142 |
|    |    | 14 | A9 | 0082 | C0 | 3C    | 00034 | MOVZWL                             | 130(R0), 20(MSG_BUFFER)                   |      |
|    |    |    |    |      | 08 | AC    | 0003A | MOVL                               | UCB, R0                                   | 1143 |
|    |    | 18 | A9 | 70   | A0 | D0    | 0003E | MOVL                               | 112(R0), 24(MSG_BUFFER)                   |      |
|    |    |    |    |      | 08 | AC    | 00043 | MOVL                               | UCB, R0                                   | 1144 |
|    |    | 1C | A9 | 54   | A0 | B0    | 00047 | MOVW                               | 84(R0), 28(MSG_BUFFER)                    |      |
|    |    |    |    |      | 2E | A9    | 0004C | CLRL                               | 46(MSG_BUFFER)                            | 1146 |
|    |    |    |    |      | 08 | AC    | 0004F | MOVL                               | UCB, R0                                   | 1149 |
|    |    |    |    |      | 50 | A0    | 00053 | MOVL                               | 40(R0), R0                                |      |
|    |    |    |    |      | 51 | A0    | 00057 | MOVZBL                             | 20(R0), R1                                |      |
|    |    |    |    |      | 51 | D6    | 0005B | INCL                               | R1  |      |
| 1E | A9 |    | 14 | A0   | 51 | 28    | 0005D | MOV3                               | R1, 20(R0), 30(MSG_BUFFER)                | 1151 |
|    |    |    |    |      | 08 | AC    | 00063 | MOVL                               | UCB, R1                                   | 1161 |
|    |    |    |    |      | 08 | AC    | 00067 | MOVL                               | UCB, R0                                   | 1153 |
|    |    |    |    |      | 05 | A0    | 0006B | BLBS                               | 59(R0), 4\$                               |      |
|    | 27 |    | 38 | A0   | 05 | E0    | 0006F | BBS                                | #5, 56(R0), 6\$                           | 1154 |
|    |    |    |    |      | 34 | A1    | 00074 | MOVL                               | 52(R1), VCB                               | 1161 |
|    |    |    |    |      | 0E | A0    | 00078 | TSTW                               | 14(VCB)                                   | 1162 |
|    |    |    |    |      | 0E | 13    | 0007B | BEQL                               | 5\$                                       |      |
|    |    |    |    |      | 51 | A0    | 0007D | MOVL                               | 32(VCB), RVT                              | 1165 |
|    |    | 2E | A9 | 0E   | A0 | B0    | 00081 | MOVW                               | 14(VCB), 46(MSG_BUFFER)                   | 1166 |
|    |    | 30 | A9 | 0B   | A1 | 9B    | 00086 | MOVZBW                             | 11(RVT), 48(MSG_BUFFER)                   | 1167 |
|    |    |    |    |      | 08 | AC    | 0008B | MOVL                               | UCB, R0                                   | 1170 |
|    |    |    |    |      | 34 | A0    | 0008F | MOVL                               | 52(R0), R0                                |      |
| 32 | A9 |    | 14 | A0   | 0C | 28    | 00093 | MOV3                               | #12, 20(R0), 50(MSG_BUFFER)               | 1171 |
|    |    |    |    |      | 61 | 11    | 00099 | BRB                                | 13\$                                      | 1153 |
|    |    |    |    |      | 50 | A1    | 0009B | MOVL                               | 52(R1), VCB                               | 1182 |
|    |    |    |    |      | 56 | A0    | 0009F | MOVL                               | 32(VCB), RVT                              | 1183 |
|    |    |    |    |      | 58 | A6    | 000A3 | MOVAB                              | 68(R6), UCBLIST                           | 1184 |
|    |    |    |    |      | 57 | A0    | 000A7 | MOVL                               | 52(VCB), MVL                              | 1185 |
|    |    |    | 30 | A9   | 0B | A7    | 000AB | MOVZBW                             | 11(MVL), 48(MSG_BUFFER)                   | 1186 |
| OC |    | 20 | 6E | 00   | 2C | 000B0 | MOV3  | #0, (SP), #32, #12, 50(MSG_BUFFER) | 1187                                      |      |
|    |    |    |    |      | 32 | A9    | 000B5 |                                    |   |      |
|    |    |    |    |      | 51 | A6    | 000B7 | MOVZBL                             | 11(RVT), R1                               | 1188 |
|    |    |    |    |      | 50 | 01    | 000BB | MNEGL                              | #1, I                                     | 1191 |
|    |    |    |    |      | 0A | 11    | 000BE | BRB                                | 8\$                                       |      |
|    |    |    |    |      | 5A | 50    | 000C0 | MOVL                               | I, RUN                                    | 1190 |
|    |    | 08 | AC | 6840 | D1 | 000C3 | CMPL  | (UCBLIST)[I], UCB                  | 1191                                      |      |
|    |    |    |    |      | 04 | 13    | 000C8 | BEQL                               | 9\$                                       |      |
|    |    |    |    |      | 51 | F2    | 000CA | AOBLS                              | R1, I, 7\$                                | 1188 |
|    |    |    |    |      | 58 | A7    | 000CE | MOVAB                              | 36(R7), MVL_ENTRY                         | 1193 |
|    |    |    |    |      | 57 | A7    | 000D2 | MOVZBL                             | 11(MVL), R7                               | 1194 |
|    |    |    |    |      | 56 | 01    | 000D6 | MNEGL                              | #1, I                                     |      |
|    |    |    |    |      | 1D | 11    | 000D9 | BRB                                | 12\$                                      |      |
| SA | 06 | A8 | 08 | 00   | ED | 000DB | CMPTV | #0, #8, 6(MVL_ENTRY), RUN          | 1196                                      |      |
|    |    |    |    |      | 12 | 12    | 000E1 | BNEQ                               | 11\$                                      |      |
|    |    |    |    |      | 0E | A8    | 000E3 | BLBC                               | 7(MVL_ENTRY), 11\$                        | 1197 |
|    |    |    |    |      | 56 | 01    | 000E7 | ADDW3                              | #1, I, 46(MSG_BUFFER)                     | 1200 |
| OC |    | 2E | A9 | 20   | 06 | 2C    | 000EC | MOV3                               | #6, (MVL_ENTRY), #32, #12, 50(MSG_BUFFER) | 1202 |
|    |    |    |    |      | 32 | A9    | 000F1 |                                    |   |      |



|    |           |    |       |       |       |       |                |                |      |      |
|----|-----------|----|-------|-------|-------|-------|----------------|----------------|------|------|
|    |           | 07 | 11    | 000F3 |       | BRB   | 13\$           | :              | 1199 |      |
|    |           | 08 | C0    | 000F5 | 11\$: | ADDL2 | #8, MVL_ENTRY  | :              | 1205 |      |
| DF |           | 56 | 57    | F2    | 000F8 | 12\$: | AOBLSS         | R7, I, T0\$    | :    | 1194 |
|    |           | 52 | 59    | D0    | 000FC | 13\$: | MOVL           | MSG_BUFFER, R2 | :    | 1212 |
|    | 00000000G | 00 | 16    | 000FF |       | JSB   | ERL\$RELEASEMB | :              |      |      |
|    |           | 01 | D0    | 00105 | 14\$: | MOVL  | #1, R0         | :              | 1214 |      |
|    |           | 04 | 00108 |       |       | RET   |                | :              | 1216 |      |

: Routine Size: 265 bytes, Routine Base: \$LOCKEDC1\$ + 0000

```

: 228      1217  1
: 229      1218  1 END
: 230      1219  0 ELUDOM

```

PSECT SUMMARY

| Name         | Bytes | Attributes   |
|--------------|-------|--|
| \$LOCKEDC1\$ | 265   | NOVEC,NJWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2) |

Library Statistics

| File                            | ----- Symbols ----- |                | Pages Mapped | Processing Time |
|---------------------------------|---------------------|----------------|--------------|-----------------|
|                                 | Total               | Loaded Percent |              |                 |
| _\$255\$DUA28:[SYSLIB]LIB.L32;1 | 18619               | 54 0           | 1000         | 00:02.0         |

COMMAND QUALIFIERS

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

```

: Size:      265 code + 0 data bytes
: Run Time:   00:12.6
: Elapsed Time: 00:25.5
: Lines/CPU Min: 795
: Lexemes/CPU-Min: 2.396
: Memory Used: 178 pages
: Compilation Complete

```

