



```

BBBBBBBB      AAAAAA      SSSSSSSS      000000      PPPPPPPP      EEEEEEEEEE      NN      NN      DDDDDDDD      EEEEEEEEEE
BBBBBBBB      AAAAAA      SSSSSSSS      000000      PPPPPPPP      EEEEEEEEEE      NN      NN      DDDDDDDD      EEEEEEEEEE
BB      BB      AA      AA      SS      00      00      PP      PP      EE      NN      NN      DD      DD      EE
BB      BB      AA      AA      SS      00      00      PP      PP      EE      NN      NN      DD      DD      EE
BB      BB      AA      AA      SS      00      00      PP      PP      EE      NNNN      NN      DD      DD      EE
BB      BB      AA      AA      SS      00      00      PP      PP      EE      NNNN      NN      DD      DD      EE
BBBBBBBB      AA      AA      SSSSSS      00      00      PPPPPPPP      EEEEEEEEEE      NN      NN      DD      DD      EEEEEEEE
BBBBBBBB      AA      AA      SSSSSS      00      00      PPPPPPPP      EEEEEEEEEE      NN      NN      DD      DD      EEEEEEEE
BB      BB      AAAAAAAAAA      SS      00      00      PP      NN      NNNN      DD      DD      EE
BB      BB      AAAAAAAAAA      SS      00      00      PP      NN      NNNN      DD      DD      EE
BB      BB      AA      AA      SS      00      00      PP      NN      NN      DD      DD      EE
BB      BB      AA      AA      SS      00      00      PP      NN      NN      DD      DD      EE
BBBBBBBB      AA      AA      SSSSSSSS      000000      PP      EEEEEEEEEE      NN      NN      DDDDDDDD      EEEEEEEEEE
BBBBBBBB      AA      AA      SSSSSSSS      000000      PP      EEEEEEEEEE      NN      NN      DDDDDDDD      EEEEEEEEEE

```

....  
....  
....  
....

```

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 BAS$$OPEN_DEFLT (
2 0002 0 IDENT = '1-041'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 **
31 0031 1 FACILITY: BASIC-PLUS-2 I/O Processing
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains BAS$$OPEN_DEFLT, the routine which
36 0036 1 implicitly opens channel 0 for a BASIC-PLUS-2 program.
37 0037 1
38 0038 1 ENVIRONMENT: VAX-11 User Mode
39 0039 1
40 0040 1 AUTHOR: John Sauter, CREATION DATE: 30-NOV-78
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. JBS 30-NOV-78
45 0045 1 1-002 - Change REQUIRE file names from FOR... to OTS... JBS 06-DEC-78
46 0046 1 1-003 - Use symbols for LUN values for PRINT and INPUT. JBS 07-DEC-78
47 0047 1 1-004 - Change OPEN$K symbols to LUB$K symbols. JBS 08-DEC-78
48 0048 1 1-005 - Get LUB$K PBUF $IZ bytes for the prompt buffer. JBS 12-DEC-78
49 0049 1 1-006 - REQUIRE BASOPN to get default record length. JBS 12-DEC-78
50 0050 1 1-007 - Remove FILE_NAME_DESC, since BAS$$STOP_IO now gets the file
51 0051 1 name from the LOB. JBS 12-DEC-78
52 0052 1 1-008 - Use %REF to insure that the length argument to LIB$GET_VM is
53 0053 1 always a longword. JBS 12-DEC-78
54 0054 1 1-009 - Set up FAB fields RFM and ROP. JBS 12-DEC-78
55 0055 1 1-010 - Put attempted file name in the LUB in case the OPEN fails
56 0056 1 so badly that no name is returned. JBS 19-DEC-78
57 0057 1 1-011 - Put OPEN or CONNECT error status in the RAB in case it is not
    
```

! File: BASOPENDE.B32 Edit:MDL1041

```

58 0058 1  stored by the OPEN, CREATE or CONNECT call. JBS 19-DEC-78
59 0059 1 1-012 - Set up margin and default margin. JBS 20-DEC-78
60 0060 1 1-013 - Set record buffer size to the default if this is a new file.
61 0061 1  JBS 26-DEC-78
62 0062 1 1-014 - Undo edit 11: make BAS$$STOP_IO extract the error code from
63 0063 1  the FAB if it is not in the RAB. In support of this,
64 0064 1  we must tell BAS$$STOP_IO whether the error happened on the
65 0065 1  $OPEN (OR $CREATE) or the $CONNECT. JBS 27-DEC-78
66 0066 1 1-015 - Open the terminal in CR format rather than PRN format.
67 0067 1  JBS 10-JAN-1979
68 0068 1 1-016 - Set the TERM FOR bit in the LUB to indicate terminal format.
69 0069 1  JBS 11-JAN-1979
70 0070 1 1-017 - Signal Internal OTS failure rather than Program lost-Sorry
71 0071 1  on an OPEN to a wrong logical unit number. JBS 15-JAN-1979
72 0072 1 1-018 - Declare the OTS exit handler to purge I/O buffers and close
73 0073 1  the file. JBS 24-JAN-1979
74 0074 1 1-019 - Put two dollar signs on the non-user entry points. JBS 26-JAN-1979
75 0075 1 1-020 - Don't set record size in RAB because reading an EOF will clear
76 0076 1  it, even for fixed-length records. JBS 31-JAN-1979
77 0077 1 1-021 - Set SQO, since we will perform only sequential operations
78 0078 1  on the PRINT and INPUT files. JBS 14-FEB-1979
79 0079 1 1-022 - Use BASIDERR.REQ to define the I/O error codes. JBS 20-FEB-1979
80 0080 1 1-023 - Set the margin based on the BLS field returned in the FAB by
81 0081 1  OPEN, if the device being opened is a terminal. JBS 22-FEB-1979
82 0082 1 1-024 - Change BAS$$STOP to BAS$$STOP_IO, so that the channel number
83 0083 1  gets reported with the error message. JBS 17-APR-1979
84 0084 1 1-025 - Do not set RAB$B_RAC, record level does. JBS 14-MAY-1979
85 0085 1 1-026 - Make the margin 16 bits. JBS 30-MAY-1979
86 0086 1 1-027 - Mark the LUBs OPENed as being terminal format. JBS 31-MAY-1979
87 0087 1 1-028 - Do not use locate-mode processing, since it causes problems
88 0088 1  for BAS$MARGIN. This is little loss, since RMS does not
89 0089 1  really do locate-mode processing on process permanent files,
90 0090 1  and channel 0 normally refers to process permanent files.
91 0091 1  JBS 04-JUN-1979
92 0092 1 1-029 - Set the language byte in the LUB, so this file can be
93 0093 1  used only by BASIC programs. This is no loss, since these
94 0094 1  LUNs are only intended for use by BASIC anyway. JBS 30-JUN-1979
95 0095 1 1-030 - If the device is a terminal, change it to PRN format,
96 0096 1  so it can be forcible. JBS 10-JUL-1979
97 0097 1 1-031 - If the LUB is opened with PRN format, mark it so. JBS 12-JUL-1979
98 0098 1 1-032 - PRN format requires VFC. JBS 17-JUL-1979
99 0099 1 1-033 - Change BAS$INPUT and BAS$PRINT to SY$$INPUT and SY$$OUTPUT.
100 0100 1  JBS 30-JUL-1979
101 0101 1 1-034 - Use the BASIC-specific exit handler. JBS 17-AUG-1979
102 0102 1 1-035 - Make the initial margin on disk files be LUB$K_D_MARGIN, and on
103 0103 1  terminals be infinite. JBS 24-AUG-1979
104 0104 1 1-036 - Set up LUB$A_UBF. JBS 15-NOV-1979
105 0105 1 1-037 - Don't set the CIF or EOF bits in the FAB and APPEND bits in the LUB.
106 0106 1  DGP 26-Feb-80
107 0107 1 1-038 - Set FAB$B_RAT and LUB$B_RAT so that BAS$REC_RSLO will add CR to
108 0108 1  records input from command files. PLL 18-Aug-81
109 0109 1 1-039 - Don't set FAB$B_RAT, RMS sets it. PLL 19-Nov-81
110 0110 1 1-040 - LIB$STOP should be declared EXTERNAL. PLL 20-Nov-81
111 0111 1 1-041 - only set PRN for the output side of channel 0. MDL 2-Sep-1983
112 0112 1  --
113 0113 1
114 0114 1 !<BLF/PAGE>

```

```

116 0115 1 |
117 0116 1 | SWITCHES:
118 0117 1 |
119 0118 1 |
120 0119 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
121 0120 1 |
122 0121 1 |
123 0122 1 | LINKAGES:
124 0123 1 |
125 0124 1 |
126 0125 1 | REQUIRE 'RTLIN:OTSLNK'; ! define linkages
127 0554 1 |
128 0555 1 |
129 0556 1 | TABLE OF CONTENTS:
130 0557 1 |
131 0558 1 |
132 0559 1 | FORWARD ROUTINE
133 0560 1 |     BAS$OPEN_DEFLT : CALL_CCB NOVALUE; ! Do a default open
134 0561 1 |
135 0562 1 |
136 0563 1 | INCLUDE FILES:
137 0564 1 |
138 0565 1 |
139 0566 1 | REQUIRE 'RTLIN:RTLPSECT'; ! Macros for defining psects
140 0661 1 |
141 0662 1 | REQUIRE 'RTLML:OTSLUB'; ! Logical Unit Block definitions
142 0802 1 |
143 0803 1 | REQUIRE 'RTLIN:BASOPN'; ! OPEN literals
144 0851 1 |
145 0852 1 | REQUIRE 'RTLIN:BASIOERR'; ! Define I/O error codes
146 0905 1 |
147 0906 1 | LIBRARY 'RTLSTARLE'; ! system definitions
148 0907 1 |
149 0908 1 |
150 0909 1 | MACROS:
151 0910 1 |
152 0911 1 |     NONE
153 0912 1 |
154 0913 1 | EQUATED SYMBOLS:
155 0914 1 |
156 0915 1 |     NONE
157 0916 1 |
158 0917 1 | PSECTS:
159 0918 1 |
160 0919 1 | DECLARE_PSECTS (BAS); ! Declare psects for BAS$ facility
161 0920 1 |
162 0921 1 | OWN STORAGE:
163 0922 1 |
164 0923 1 |     NONE
165 0924 1 |
166 0925 1 | EXTERNAL REFERENCES:
167 0926 1 |
168 0927 1 |
169 0928 1 | EXTERNAL ROUTINE
170 0929 1 |     LIB$STOP : NOVALUE, ! signal fatal error
171 0930 1 |     BAS$$STOP IO : NOVALUE, ! signals fatal I/O error
172 0931 1 |     LIB$GET_VM, ! get virtual storage

```

```
: 173      0932 1      BAS$$DECL_EXITH : NOVALUE;          ! Declare BASIC exit handler
: 174      0933 1
: 175      0934 1  EXTERNAL
: 176      0935 1      BAS$$L_XIT_LOCK;          ! True if exit handler already declared
: 177      0936 1
: 178      0937 1  !+
: 179      0938 1  !- The following are the BASIC error codes used in this module.
: 180      0939 1  !-
: 181      0940 1
: 182      0941 1  EXTERNAL LITERAL
: 183      0942 1      BAS$$K_IO_CHAALR : UNSIGNED (8),      ! I/O channel already open
: 184      0943 1      BAS$$K_SYNRERR : UNSIGNED (8),      ! Syntax error
: 185      0944 1      BAS$$K_FATSYSIO : UNSIGNED (8),      ! Fatal system I/O error
: 186      0945 1      BAS$$K_FILATTNOT : UNSIGNED (8),     ! File attributes not matched
: 187      0946 1      BAS$$K_ILLUSA : UNSIGNED (8),        ! Illegal Usage
: 188      0947 1      BAS$$K_MAXMEMEXC : UNSIGNED (8);     ! Maximum memory exceeded
: 189      0948 1
: 190      0949 1  EXTERNAL LITERAL
: 191      0950 1      OTS$_FATINTERR;          ! OTS Fatal internal error
: 192      0951 1
```

```

194 0952 1 GLOBAL ROUTINE BASS$OPEN_DEFLT          ! Default open
195 0953 1   : NOVALUE CALL_CCB =
196 0954 1
197 0955 1  +-+
198 0956 1  FUNCTIONAL DESCRIPTION:
199 0957 1
200 0958 1      Do a default open. This routine is called when an input or
201 0959 1      output statement is done to channel 0 (= a negative LUN) and
202 0960 1      it is not open. This routine opens the LUN as a terminal format
203 0961 1      file.
204 0962 1
205 0963 1  FORMAL PARAMETERS:
206 0964 1
207 0965 1      NONE
208 0966 1
209 0967 1  IMPLICIT INPUTS:
210 0968 1
211 0969 1      The CCB, which is passed in a register.
212 0970 1
213 0971 1  IMPLICIT OUTPUTS:
214 0972 1
215 0973 1      A lot of fields in the LUB.
216 0974 1
217 0975 1  ROUTINE VALUE:
218 0976 1  COMPLETION CODES:
219 0977 1
220 0978 1      NONE
221 0979 1
222 0980 1  SIDE EFFECTS:
223 0981 1
224 0982 1      Either opens a file, thus permitting use of channel 0
225 0983 1      by BASIC I/O statements, or calls BASS$STOP_IO, thus not
226 0984 1      returning to its caller.
227 0985 1
228 0986 1  --
229 0987 1
230 0988 2  BEGIN
231 0989 2
232 0990 2  EXTERNAL REGISTER
233 0991 2      CCB : REF BLOCK [0, BYTE];
234 0992 2
235 0993 2  LOCAL
236 0994 2      OPEN STATUS,          ! RMS status returned by $OPEN or $CREATE
237 0995 2      CONNECT STATUS,       ! RMS status returned by $CONNECT
238 0996 2      FAB_BLOCK : BLOCK [FAB$C_BLN, BYTE], ! local FAB
239 0997 2      FAB : REF BLOCK [FAB$C_BLN, BYTE], ! pointer to FAB
240 0998 2      NAM_BLOCK : BLOCK [NAM$C_BLN, BYTE], ! local NAM block
241 0999 2      XAB_BLOCK : BLOCK [XAB$C_FHCLN, BYTE], ! local XABFHC block
242 1000 2      FILE_NAME : BLOCK [NAM$C_MAXRSS, BYTE]; ! text for file name
243 1001 2
244 1002 2  +-+
245 1003 2  In BASIC, only channel 0 can go through default open. This maps
246 1004 2  into one LUN for INPUT, LINPUT and INPUT LINE, and another LUN
247 1005 2  for PRINT and PRINT USING. In the code below, we first do the
248 1006 2  common setups, and then a SELECTONE statement separates the
249 1007 2  setups for the two LUNs.
250 1008 2  -

```

```

251 1009 2 FAB = FAB_BLOCK;
252 1010 2 CH$FILL (0, FAB$C_BLN, .FAB); ! clea the FAB
253 1011 2 FAB [FAB$B_BID] = FAB$C_BID; ! this is a FAB
254 1012 2 FAB [FAB$B_BLN] = FAB$C_BLN; ! length of a FAB
255 1013 2 FAB [FAB$B_RFM] = FAB$C_VAR; ! Variable length records
256 1014 2 FAB [FAB$V_CR] = 1; ! Assume LF before record and CR after
257 1015 2 FAB [FAB$V_SQO] = 1; ! Only sequential operations
258 1016 2
259 1017 2 + Set up the LUB.
260 1018 2 -
261 1019 2 CCB [LUB$A_FAB] = .FAB; ! Store FAB pointer in the LUB
262 1020 2 CCB [LUB$V_FORMATTED] = 1; ! This file must be formatted
263 1021 2 CCB [LUB$L_LOG_RECNO] = 1; ! We are on record number 1
264 1022 2 CCB [LUB$B_ORGAN] = LUB$K_ORG_TERM1; ! terminal organization
265 1023 2 CCB [LUB$B_RAT] = FAB [FAB$B_RAT]; ! store FAB record attribute
266 1024 2
267 1025 2 + Set up the right margin and default right margin.
268 1026 2 -
269 1027 2 CCB [LUB$W_D_MARGIN] = LUB$K_D_MARGIN;
270 1028 2 CCB [LUB$W_R_MARGIN] = .CCB [LUB$W_D_MARGIN];
271 1029 2 CCB [LUB$V_NOMARGIN] = 0;
272 1030 2 CCB [LUB$V_UNIT_0] = 1; ! This is BASIC channel 0
273 1031 2 CCB [LUB$V_TERM_FOR] = 1; ! File is in terminal format
274 1032 2
275 1033 2 + Set up the RAB
276 1034 2 -
277 1035 2 CCB [RAB$B_BID] = RAB$C_BID; ! This is a RAB
278 1036 2 CCB [RAB$B_BLN] = RAB$C_BLN; ! Length of a RAB
279 1037 2 CCB [RAB$V_LOC] = 0; ! Do not do locate-mode GETs
280 1038 2 CCB [RAB$L_FAB] = .FAB; ! Store pointer to FAB
281 1039 2
282 1040 2 + Set up the NAM block
283 1041 2 -
284 1042 2 CH$FILL (0, NAM$C_BLN, NAM_BLOCK); ! Clear the NAM block
285 1043 2 NAM_BLOCK [NAM$B_BID] = NAM$C_BID; ! This is a NAM block
286 1044 2 NAM_BLOCK [NAM$B_BLN] = NAM$C_BLN; ! Length of a NAM block
287 1045 2
288 1046 2 + Set up file name pointers and lengths
289 1047 2 -
290 1048 2 NAM_BLOCK [NAM$L_RSA] = NAM_BLOCK [NAM$L_ESA] = FILE_NAME;
291 1049 2 NAM_BLOCK [NAM$B_RSS] = NAM_BLOCK [NAM$B_ESS] = NAM$C_MAXRSS;
292 1050 2 FAB [FAB$L_NAM] = NAM_BLOCK; ! Store pointer in FAB
293 1051 2
294 1052 2 + Initialize the FHC XAB.
295 1053 2 -
296 1054 2 CH$FILL (0, XAB$C_FHLEN, XAB_BLOCK); ! Clear XAB
297 1055 2 XAB_BLOCK [XAB$B_OD] = XAB$C_FHC; ! This is an FHC XAB
298 1056 2 XAB_BLOCK [XAB$B_BLN] = XAB$C_FHLEN; ! Length of an FHC XAB
299 1057 2 FAB [FAB$L_XAB] = XAB_BLOCK; ! Store pointer in FAB
300 1058 2
301 1059 2 + The remainder of the initialization is done differently depending
302 1060 2 on whether this is an input-type default open or an output-type
303 1061 2 default open.
304 1062 2 -
305 1063 2
306 1064 2 SELECTONE (.CCB [LUB$W_LUN]) OF
307 1065 2 SET

```



```

308 1066 [LUB$K_LUN_INPU] : ! INPUT statement
309 1067 BEGIN
310 1068
311 1069
312 1070
313 1071
314 1072
315 1073
316 1074
317 1075
318 1076
319 1077
320 1078
321 1079
322 1080
323 1081
324 1082
325 1083
326 1084
327 1085
328 1086
329 1087
330 1088
331 1089
332 1090
333 1091
334 1092
335 1093
336 1094
337 1095
338 1096
339 1097
340 1098
341 1099
342 1100
343 1101
344 1102
345 1103
346 1104
347 1105
348 1106
349 1107
350 1108
351 1109
352 1110
353 1111
354 1112
355 1113
356 1114
357 1115
358 1116
359 1117
360 1118
361 1119
362 1120
363 1121
364 1122

    Set up a file name of SYSS$INPUT:SYSINPUT.DAT

    FAB [FAB$B_DNS] = %CHARCOUNT ('SYSINPUT.DAT');
    FAB [FAB$L_DNA] = UPLIT ('SYSINPUT.DAT');
    FAB [FAB$B_FNS] = %CHARCOUNT ('SYSS$INPUT:');
    FAB [FAB$L_FNA] = UPLIT ('SYSS$INPUT:');
    CCB [LUB$B_RSL] = %CHARCOUNT ('SYSS$INPUT:SYSINPUT.DAT');
    CCB [LUB$A_RSN] = UPLIT ('SYSS$INPUT:SYSINPUT.DAT');
    CCB [RAB$V_PMT] = 1; ! Use prompt buffer on read

    Point the RAB to the Prompt buffer.
    It is allocated dynamically.
    It must be deallocated when the file is closed.

    BEGIN

    LOCAL
    GET_VM_RESULT;

    IF ( NOT (GET_VM_RESULT = LIB$GET_VM (%REF (LUB$K_PBUF_SIZ), CCB [RAB$L_PBF])))
    THEN
    BAS$$STOP_IO (BAS$K_MAXMEMEXC);

    END;
    CCB [RAB$B_PSZ] = 0; ! Will be filled in as needed
    FAB [FAB$V_GET] = 1; ! only allow reading from this LUN
    FAB [FAB$V_SHRGET] = 1; ! allow others to read also
    CCB [LUB$V_READ_ONLY] = 1; ! We will only read from this LUN
    CCB [LUB$V_OLD_FILE] = 1; ! File must already exist
    END;

    Default OPEN for BASIC PRINT statement.

    [LUB$K_LUN_BPRI] : ! PRINT statement
    BEGIN

    Set the file name to SYSS$OUTPUT:SYSOUTPUT.DAT

    FAB [FAB$B_DNS] = %CHARCOUNT ('SYSOUTPUT.DAT');
    FAB [FAB$L_DNA] = UPLIT ('SYSOUTPUT.DAT');
    FAB [FAB$B_FNS] = %CHARCOUNT ('SYSS$OUTPUT:');
    FAB [FAB$L_FNA] = UPLIT ('SYSS$OUTPUT:');
    CCB [LUB$B_RSL] = %CHARCOUNT ('SYSS$OUTPUT:SYSOUTPUT.DAT');
    CCB [LUB$A_RSN] = UPLIT ('SYSS$OUTPUT:SYSOUTPUT.DAT');
    FAB [FAB$V_PUT] = 1; ! Only allow PUTs to this LUN
    FAB [FAB$V_NIL] = 1; ! no others may access this file
    END;

    [OTHERWISE] :
    LIB$STOP (OTSS$_FATINTERR);
    YES;

```

BASSOPEN\_DEFLT  
1-041

: 365

1123 2 !<BLF/PAGE>

M 7  
16-Sep-1984 00:57:05  
14-Sep-1984 11:56:23

VAX-11 Bliss-32 V4.0-742  
[BASRTL.SRC]BASOPENDE.B32;1

Page 8  
(J)

```

367 1124 2  !+
368 1125 2  !- Now open the file.
369 1126 2  !-
370 1127 2  OPEN_STATUS = (IF (.CCB [LUB$V_OLD_FILE]) THEN $OPEN (FAB = .FAB) ELSE $CREATE (FAB = .FAB));
371 1128 2  !+
372 1129 2  !- If the OPEN succeeded, check for a terminal format file on a terminal device,
373 1130 2  !- and change to PRN format if so. This is so that the terminal is forcible.
374 1131 2  !-
375 1132 2  !-
376 1133 2  IF (.OPEN_STATUS)
377 1134 2  THEN
378 1135 2  BEGIN
379 1136 2  IF ((.FAB [FAB$L_DEV] AND DEV$M_TRM) NEQ 0)
380 1137 4  THEN
381 1138 3  BEGIN
382 1139 4  !+
383 1140 4  !- only reset to PRN if this is the output side of channel 0.
384 1141 4  !- an attempt to do so on a file opened for input results in an error from RMS.
385 1142 4  !-
386 1143 4  !-
387 1144 4  !-
388 1145 4  !-
389 1146 5  IF (.CCB [LUB$W_LUN] EQL LUB$K_LUN_BPRI)
390 1147 4  THEN
391 1148 5  BEGIN
392 1149 5  !+
393 1150 5  !- Close and re-open the FAB, since we do not have $MODIFY.
394 1151 5  !-
395 1152 5  !-
396 1153 5  !-
397 1154 5  IF ( NOT $CLOSE (FAB = .FAB)) THEN BASS$STOP_IO (BASS$K_IOERR_REC);
398 1155 5  !+
399 1156 5  !- Turn off CR and turn on PRN.
400 1157 5  !-
401 1158 5  !-
402 1159 5  FAB [FAB$V_CR] = 0;
403 1160 5  FAB [FAB$V_PRN] = 1;
404 1161 5  CCB [RAB$L_RHB] = CCB [LUB$W_BAS_VFC];
405 1162 5  FAB [FAB$B_RFM] = FAB$C_VFC;
406 1163 5  !+
407 1164 5  !- Re-open the user's file.
408 1165 5  !-
409 1166 5  OPEN_STATUS = $OPEN (FAB = .FAB);
410 1167 4  END;
411 1168 4  !+
412 1169 4  !- indicate a terminal device is forcible : (both input & output sides)
413 1170 4  !-
414 1171 4  !-
415 1172 4  CCB [LUB$V_FORCIBLE] = .
416 1173 4  END;
417 1174 3  !-
418 1175 2  END;
419 1176 2  !-
420 1177 2  IF (.OPEN_STATUS) THEN CONNECT_STATUS = $CONNECT (RAB = .CCB);
421 1178 2  !-
422 1179 2  !+
423 1180 2  !- Store away the Directory ID in case CLOSE needs to delete the file.

```

```

424 1181 2 | Also save the IFI.
425 1182 2 | -
426 1183 2 | CH$MOVE (NAM$S_DID, NAM_BLOCK [NAM$W_DID], CCB [LUB$W_DID]);
427 1184 2 | CCB [LUB$W_IFI] = .FAB [FAB$W_IFI];
428 1185 2 | +
429 1186 2 | If we have an expanded name string or a resultant name string, point
430 1187 2 | the LUB to it instead of the user-supplied name, to improve error
431 1188 2 | messages.
432 1189 2 | -
433 1190 2 |
434 1191 2 | IF (.NAM_BLOCK [NAM$B_RSL] NEQA 0)
435 1192 2 | THEN
436 1193 2 | BEGIN
437 1194 2 | CCB [LUB$A_RSN] = .NAM_BLOCK [NAM$L_RSA];
438 1195 2 | CCB [LUB$B_RSL] = .NAM_BLOCK [NAM$B_RSL];
439 1196 2 | END
440 1197 2 | ELSE
441 1198 2 |
442 1199 2 | IF (.NAM_BLOCK [NAM$B_ESL] NEQA 0)
443 1200 2 | THEN
444 1201 2 | BEGIN
445 1202 2 | CCB [LUB$A_RSN] = .NAM_BLOCK [NAM$L_ESA];
446 1203 2 | CCB [LUB$B_RSL] = .NAM_BLOCK [NAM$B_ESL];
447 1204 2 | END;
448 1205 2 |
449 1206 2 | +
450 1207 2 | If OPEN or CREATE got an error, give an appropriate error message.
451 1208 2 | -
452 1209 2 |
453 1210 2 | IF ( NOT .OPEN_STATUS) THEN BAS$$STOP_IO (BAS$K_IOERR_OPE);
454 1211 2 |
455 1212 2 | +
456 1213 2 | If CONNECT got an error, give an appropriate error message.
457 1214 2 | -
458 1215 2 |
459 1216 2 | IF ( NOT .CONNECT_STATUS) THEN BAS$$STOP_IO (BAS$K_IOERR_CON);
460 1217 2 |
461 1218 2 | +
462 1219 2 | If the device opened is a terminal, set the TERM_DEV bit in the LUB
463 1220 2 | and set the default margin based on the width of the terminal, which is
464 1221 2 | returned in the BLS field of the FAB.
465 1222 2 | -
466 1223 2 |
467 1224 2 | IF ((.FAB [FAB$L_DEV] AND DEV$M_TRM) NEQ 0)
468 1225 2 | THEN
469 1226 2 | BEGIN
470 1227 2 | CCB [LUB$V_TERM_DEV] = 1;
471 1228 2 | CCB [LUB$W_D_MARGIN] = .FAB [FAB$W_BLS];
472 1229 2 | CCB [LUB$W_R_MARGIN] = 0;
473 1230 2 | CCB [LUB$V_NOMARGIN] = 1;
474 1231 2 | END;
475 1232 2 |
476 1233 2 | +
477 1234 2 | If the file just opened was already in existence, perform
478 1235 2 | consistency checks between the file's attributes and the
479 1236 2 | default parameters.
480 1237 2 | -

```

```

481 1238
482 1239     IF (.CCB [LUB$V_OLD_FILE])
483 1240     THEN
484 1241         BEGIN
485 1242     +
486 1243     - Organization check: must be sequential.
487 1244
488 1245
489 1246         IF (.FAB [FAB$B_ORG] NEQ FAB$C_SEQ) THEN BASS$STOP_IO (BASS$K_FILATTNOT);
490 1247
491 1248     +
492 1249     - If the file is in PRN format, record the fact.
493 1250
494 1251
495 1252         IF (.FAB [FAB$V_PRN]) THEN CCB [LUB$V_PRN] = 1;
496 1253
497 1254     +
498 1255     - Compute the record size and store it in the LUB.
499 1256
500 1257         CCB [LUB$W_RBUF_SIZE] = MAXU (.FAB [FAB$W_MRS], .XAB_BLOCK [XAB$W_LRL], BASS$K_DEF_RECLE);
501 1258         END                                     ! end of old file processing
502 1259     ELSE
503 1260         BEGIN
504 1261     +
505 1262     - The following processing is done only if this is a new file.
506 1263
507 1264     +
508 1265     - Set the record buffer size to the default.
509 1266
510 1267         CCB [LUB$W_RBUF_SIZE] = BASS$K_DEF_RECLE;
511 1268         END;                                     ! end of new file processing
512 1269
513 1270     +
514 1271     - Don't permit the 'undefined' record format, since it can be used
515 1272     only with block I/O.
516 1273
517 1274
518 1275         SELECTONE (.FAB [FAB$B_RFM]) OF
519 1276         SET
520 1277
521 1278         [FAB$C_FIX, FAB$C_VAR, FAB$C_VFC] :      ! ok, do nothing.
522 1279         ;
523 1280
524 1281         [OTHERWISE] :
525 1282         BASS$STOP_IO (BASS$K_ILLUSA);
526 1283         TES;
527 1284
528 1285     +
529 1286     - Allocate a record buffer.
530 1287
531 1288         BEGIN
532 1289
533 1290         LOCAL
534 1291         GET_VM_RESULT;
535 1292
536 1293         GET_VM_RESULT = LIB$GET_VM (%REF (.CCB [LUB$W_RBUF_SIZE]), CCB [LUB$A_RBUF_ADR]);
537 1294

```

```

538      1295      3      IF ( NOT .GET_VM_RESULT) THEN BAS$$STOP_IO (BAS$K_MAXMEMEXC);
539      1296      3
540      1297      3      END;
541      1298      2      +
542      1299      2      Allocate dynamic storage for the file name so that the name can be
543      1300      2      used later for error diagnostics. Point the LUB to the new location.
544      1301      2      Indicate that the space pointed to must be deallocated when the file
545      1302      2      is closed.
546      1303      2      -
547      1304      3      BEGIN
548      1305      3
549      1306      3      LOCAL
550      1307      3      GET_VM_RESULT,
551      1308      3      OLD_ADDRESS;
552      1309      3
553      1310      3      OLD_ADDRESS = .CCB [LUB$A_RSN];
554      1311      3      GET_VM_RESULT = LIB$GET_VM (%REF (.CCB [LUB$B_RSL]), CCB [LUB$A_RSN]);
555      1312      3
556      1313      3      IF ( NOT .GET_VM_RESULT) THEN LIB$STOP (.GET_VM_RESULT);
557      1314      3
558      1315      3      CH$MOVE (.CCB [LUB$B_RSL], .OLD_ADDRESS, .CCB [LUB$A_RSN]);
559      1316      3      CCB [LUB$V_VIRT_RSN] = 1;
560      1317      2      END;
561      1318      2      +
562      1319      2      Set those RAB fields that seldom change.
563      1320      2      -
564      1321      2      CCB [RAB$L_UBF] = .CCB [LUB$A_RBUF_ADR];
565      1322      2      CCB [RAB$W_USZ] = .CCB [LUB$W_RBUF_SIZE];
566      1323      2      CCB [LUB$A_UBF] = .CCB [LUB$A_RBUF_ADR];
567      1324      2      +
568      1325      2      Clear LUB$A_FAB to indicate that the FAB is no longer present.
569      1326      2      -
570      1327      2      CCB [LUB$A_FAB] = 0;
571      1328      2      +
572      1329      2      Indicate that the file is now open for BASIC.
573      1330      2      -
574      1331      2      CCB [LUB$B_LANGUAGE] = LUB$K_LANG_BAS;
575      1332      2      CCB [LUB$V_OPENED] = 1;
576      1333      2      +
577      1334      2      Declare the BASIC exit handler to purge I/O buffers and close the
578      1335      2      file when the image exits.
579      1336      2      -
580      1337      2
581      1338      2      IF ( NOT BAS$$L_XIT_LOCK) THEN BAS$$DECL_EXITH ();
582      1339      2
583      1340      2      RETURN;
584      1341      1      END;

```

! of routine BAS\$\$OPEN\_DEFLT

```

.TITLE BAS$$OPEN_DEFLT
.IDENT \1-041\
.PSELT _BAS$CODE,NOWRT, SHR, PIC,2

```

```

4E 49 53 54 41 44 2E 54 55 50 4E 49 53 59 53 0000 P.AAA: .ASCII \SYSINPUT.DAT\
00 00 3A 54 55 50 4E 49 24 53 59 53 0000C P.AAB: .ASCII \SYS$INPUT:\<0><0>
59 53 3A 54 55 50 4E 49 24 53 59 53 00018 P.AAC: .ASCII \SYS$INPUT:SYSINPUT.DAT\<0><0>

```

00	00	54	41	44	2E	54	55	50	54	41	44	2E	54	55	50	00027
						54	55	50	54	55	4F	53	59	53	00030	P.AAD: .ASCII \SYSOUTPUT.DAT\<0><0><0>
															00	0003F
4F	53	59	00	3A	54	55	50	54	55	4F	24	53	59	53	00040	P.AAE: .ASCII \SYS\$OUTPUT:\<0>
						54	55	54	55	4F	24	53	59	53	0004C	P.AAF: .ASCII \SYS\$OUTPUT:SYSOUTPUT.DAT\
						54	41	44	2E	54	55	50	54	55	0005B	

```

.EXTRN LIB$STOP, BASS$STOP IO
.EXTRN LIB$GET_VM, BASS$DECL_EXITH
.EXTRN BASS$L_XIT_LOCK
.EXTRN BASSK_IO CHAALR
.EXTRN BASSK_SYNERR, BASSK_FATSYSIO_
.EXTRN BASSK_FILATTNOT
.EXTRN BASSK_ILLUSA, BASSK_MAXMEMEXC
.EXTRN OTSS_FATINTERR, SYS$OPEN
.EXTRN SYS$CREATE, SYS$CLOSE
.EXTRN SYS$CONNECT

```

07FC 00000

```

.ENTRY BASS$OPEN_DEFLT, Save R2,R3,R4,R5,R6,R7,R8,-; 0952
R9,R10
MOVAB -480(SP), SP
MOVAB FAB_BLOCK, FAB
MOVCS #0, (SP), #0, #80, (FAB)
MOVW #20483, (FAB)
MOVAB #2, 31(FAB)
MOVAB 30(FAB), R9
BISB2 #2, (R9)
BISB2 #64, 4(FAB)
MOVL FAB, -24(CCB)
MOVAB -4(CCB), R7
BISB2 #1, 1(R7)
MOVL #1, -32(CCB)
MOVAB #4, -60(CCB)
MOVAB R9, -10(CCB)
MOVZBW #72, -42(CCB)
MOVW -42(CCB), -44(CCB)
BICB2 #2, -95(CCB)
MOVAB -2(CCB), R8
BISB2 #144, (R8)
MOVW #17409, (CCB)
BICB2 #1, 6(CCB)
MOVL FAB, 60(CCB)
MOVCS #0, (SP), #0, #96, NAM_BLOCK
MOVW #24578, NAM_BLOCK
MOVAB FILE_NAME, R0
MOVL R0, NAM_BLOCK+12
MOVL R0, NAM_BLOCK+4
MNEGB #1, NAM_BLOCK+10
MNEGB #1, NAM_BLOCK+2
MOVAB NAM_BLOCK, 40(FAB)
MOVCS #0, (SP), #0, #44, XAB_BLOCK
MOVW #11293, XAB_BLOCK
MOVAB XAB_BLOCK, 36(FAB)
CVTBL -58(CCB), R0

```

						5E	FE20	CE	9E	00002		
						56	80	AD	9E	00007		
0050	8F				00	6E		00	2C	0000B		
								66		00012		
						66	5003	8F	80	00013		
				1F		A6		02	90	00018		
						59		1E	A6	9E	0001C	
						69		02	88	00020		
				04		A6		40	8F	88	00023	
				E8		AB			56	D0	00028	
						57		FC	AB	9E	0002C	
				01		A7			01	88	00030	
				E0		AB			01	D0	00034	
				C4		AB			04	90	00038	
				F6		AB			59	90	0003C	
				D6		AB		48	8F	9B	00040	
				D4		AB		D6	AB	B0	00045	
				A1		AB			02	8A	0004A	
						58		FE	AB	9E	0004E	
						68		90	8F	88	00052	
						6B		4401	8F	B0	00056	
				06		AB			01	8A	0005B	
				3C		AB			56	D0	0005F	
0060	8F				00	6E			00	2C	00063	
								FF50	CD		0006A	
						FF50		6002	8F	B0	0006D	
								04	AE	9E	00074	
						FF5C			50	D0	00078	
						FF54			50	D0	0007D	
						FF5A			01	8E	00082	
						FF52			01	8E	00087	
						28			01	8E	00087	
						A6		FF50	CD	9E	0008C	
						6E			00	2C	00092	
									FF24	CD	00097	
						FF24			2C1D	8F	B0	0009A
						24			FF24	CD	9E	000A1
						50			C6	AB	32	000A7

FFF9	8F		50	B1	000AB	CMPW	RO, #-7	1067
			4F	12	000B0	BNEQ	2\$	
30	A6	FEE6	CF	9E	000B2	MOVAB	P.AAA, 48(FAB)	1073
34	A6	OCOA	8F	B0	000B8	MOVW	#3082, 52(FAB)	1074
2C	A6	FEE6	CF	9E	000BE	MOVAB	P.AAB, 44(FAB)	1075
F7	AB		16	90	000C4	MOVB	#22, -9(CCB)	1076
F8	AB	FEE8	CF	9E	000C8	MOVAB	P.AAC, -8(CCB)	1077
07	AB		40	8F	000CE	BISB2	#64, 7(CCB)	1078
			30	AB	9F	PUSHAB	48(CCB)	1089
04	AE		50	3F	9A	MOVZBL	#80, 4(SP)	
			04	AE	9F	PUSHAB	4(SP)	
00000000G	00		02	FB	000DE	CALLS	#2, LIB\$GET_VM	
	0B		50	E8	000E5	BLBS	GET_VM_RESULT, 1\$	
	7E	00G	8F	9A	000E8	MOVZBL	#BASSK-MAXMEMEXC, -(SP)	1091
00000000G	00		01	FB	000EC	CALLS	#1, BASS\$STOP_IO	
			34	AB	94	CLRB	52(CCB)	1094
16	A6	0202	8F	A8	000F6	BISW2	#514, 22(FAB)	1096
	67		0C	88	000FC	BISB2	#12, (R7)	1098
			38	11	000FF	BRB	4\$	1064
FFF8	8F		50	B1	00101	CMPW	RO, #-8	1104
			24	12	00106	BNEQ	3\$	
30	A6	FECO	CF	9E	00108	MOVAB	P.AAD, 48(FAB)	1110
34	A6	ODOB	8F	B0	0010E	MOVW	#3339, 52(FAB)	1111
2C	A6	FEC4	CF	9E	00114	MOVAB	P.AAE, 44(FAB)	1112
F7	AB		18	90	0011A	MOVB	#24, -9(CCB)	1113
F8	AB	FEC6	CF	9E	0011E	MOVAB	P.AAF, -8(CCB)	1114
16	A6	2001	8F	A8	00124	BISW2	#8193, 22(FAB)	1116
			0D	11	0012A	BRB	4\$	1064
		00000000G	8F	DD	0012C	PUSHL	#OTSS FATINTERR	1120
00000000G	00		01	FB	00132	CALLS	#1, LIB\$STOP	
0B	67		03	E1	00139	BBC	#3, (R7), 5\$	1127
			56	DD	0013D	PUSHL	FAB	
00000000G	00		01	FB	0013F	CALLS	#1, SYSS\$OPEN	
			09	11	00146	BRB	6\$	
			56	DD	00148	PUSHL	FAB	
00000000G	00		01	FB	0014A	CALLS	#1, SYSS\$CREATE	
	5A		50	D0	00151	MOVL	RO, OPEN STATUS	
	51		5A	E9	00154	BLBC	OPEN STATUS, 10\$	1133
3D	40		02	E1	00157	BBC	#2, 84(FAB), 9\$	1137
	FFF8	C6	AB	B1	0015C	CMPW	-58(CCB), #-8	1146
			31	12	00162	BNEQ	8\$	
			56	DD	00164	PUSHL	FAB	1154
00000000G	00		01	FB	00166	CALLS	#1, SYSS\$CLOSE	
	0A		50	E8	0016D	BLBS	RO, 7\$	
	7E		01	CE	00170	MNEGL	#1, -(SP)	
00000000G	00		01	FB	00173	CALLS	#1, BASS\$STOP_IO	
	69		02	8A	0017A	BICB2	#2, (R9)	1159
	69		04	88	0017D	BISB2	#4, (R9)	1160
	2C	DA	AB	9E	00180	MOVAB	-38(R11), 44(CCB)	1161
	1F		03	90	00185	MOVB	#3, 31(FAB)	1162
			56	DD	00189	PUSHL	FAB	1166
00000000G	00		01	FB	0018B	CALLS	#1, SYSS\$OPEN	
	5A		50	D0	00192	MOVL	RO, OPEN STATUS	
	68	40	8F	88	00195	BISB2	#64, (R8)	1172
	0C		5A	E9	00199	BLBC	OPEN STATUS, 10\$	1177
			5B	DD	0019C	PUSHL	CCB	
00000000G	00		01	FB	0019E	CALLS	#1, SYSS\$CONNECT	



FO	AB	FF7A	6E		50	DO	001A5		MOVL	R0, CONNECT STATUS		
		D0	CD		06	28	001A8	10\$:	MOVC3	#6, NAM_BLOCK+42, -16(CCB)		1183
			AB	02	A6	B0	001AF		MOVW	2(FAB), -48(CCB)		1184
			50	FF53	CD	9A	001B4		MOVZBL	NAM_BLOCK+3, R0		1191
		F8	AB	FF54	08	13	001B9		BEQL	11\$		
					CD	DO	001BB		MOVL	NAM_BLOCK+4, -8(CCB)		1194
			50	FF5B	0D	11	001C1		BRB	12\$		1195
					CD	9A	001C3	11\$:	MOVZBL	NAM_BLOCK+11, R0		1199
		F8	AB	FF5C	0A	13	001C8		BEQL	13\$		
		F7	AB		CD	DO	001CA		MOVL	NAM_BLOCK+12, -8(CCB)		1202
			0A		50	90	001D0	12\$:	MOVB	R0, -9(CCB)		1203
			7E		5A	E8	001D4	13\$:	BLBS	OPEN_STATUS, 14\$		1210
		00000000G	00		02	CE	001D7		MNEGL	#2, -(SP)		
			0A		01	FB	001DA		CALLS	#1, BAS\$\$STOP IO		
			7E		6E	E8	001E1	14\$:	BLBS	CONNECT STATUS, 15\$		1216
		00000000G	00		03	CE	001E4		MNEGL	#3, -(SP)		
	OF	40	A6		01	FB	001E7		CALLS	#1, BAS\$\$STOP IO		
			68		02	E1	001EE	15\$:	BBC	#2, 64(FAB), T6\$		1224
		D6	AB	3C	20	88	001F3		BISB2	#32, (R8)		1227
				D4	A6	B0	001F6		MOVW	60(FAB), -42(CCB)		1228
		A1	AB		AB	B4	001FB		CLRW	-44(CCB)		1229
	39		67		02	88	001FE		BISB2	#2, -95(CCB)		1230
				1D	03	E1	00202	16\$:	BBC	#3, (R7), 21\$		1239
			7E		A6	95	00206		TSTB	29(FAB)		1246
		00000000G	00	00G	0B	13	00209		BEQL	17\$		
			69		8F	9A	0020B		MOVZBL	#BASSK_FILATT:UT, -(SP)		
		A1	AB		01	FB	0020F		CALLS	#1, BAS\$\$STOP IO		
	04		50		02	E1	00216	17\$:	BBC	#2, (R9), 18\$		1252
			50		01	88	0021A		BISB2	#1, -95(CCB)		
				36	A6	3C	0021E	18\$:	MOVZWL	54(FAB), R0		1257
				FF2E	CD	B1	00222		CMPW	XAB_BLOCK+10, R0		
			50		05	1B	00227		BLEQU	19\$		
		0084	8F	FF2E	CD	3C	00229		MOVZWL	XAB_BLOCK+10, R0		
					50	B1	0022E	19\$:	CMPW	R0, #132		
			50		04	1E	00233		BGEQU	20\$		
		D2	AB	84	8F	9A	00235		MOVZBL	#132, R0		
					50	B0	00239	20\$:	MOVW	R0, -46(CCB)		
					05	11	0023D		BRB	22\$		1239
		D2	AB	84	8F	9B	0023F	21\$:	MOVZBW	#132, -46(CCB)		1267
			50	1F	A6	9A	00244	22\$:	MOVZBL	31(FAB), R0		1275
					05	15	00248		BLEQ	23\$		1278
					50	91	0024A		CMPB	R0, #3		
					0B	1B	0024D		BLEQU	24\$		
		00000000G	00	00G	8F	9A	0024F	23\$:	MOVZBL	#BASSK_ILLUSA, -(SP)		1282
					01	FB	00253		CALLS	#1, BAS\$\$STOP IO		
				EC	AB	9F	0025A	24\$:	PUSHAB	-20(CCB)		1293
		04	AE	D2	AB	3C	0025D		MOVZWL	-46(CCB), 4(SP)		
				04	AE	9F	00262		PUSHAB	4(SP)		
		00000000G	00		02	FB	00265		CALLS	#2, LIB\$GET_VM		
			0B		50	E8	0026C		BLBS	GET_VM_RESULT, 25\$		1295
			7E	00G	8F	9A	0026F		MOVZBL	#BASSK_MAXMEMEXC, -(SP)		
		00000000G	00		01	FB	00273		CALLS	#1, BAS\$\$STOP IO		
			52	F8	AB	DO	0027A	25\$:	MOVL	-8(CCB), OLD_ADDRESS		1310
				F8	AB	9F	0027E		PUSHAB	-8(CCB)		1311
		04	AE	F7	AB	9A	00281		MOVZBL	-9(CCB), 4(SP)		
				04	AE	9F	00286		PUSHAB	4(SP)		
		00000000G	00		02	FB	00289		CALLS	#2, LIB\$GET_VM		

```

09          50 E8 00290          BLBS GET_VM_RESULT, 26$          ; 1313
          50 DD 00293          PUSHL GET_VM_RESULT          ;
          01 FB 00295          CALLS #1, LIB$STOP          ;
00000000G 00          AB 9A 0029C 26$: MOVZBL -9(CCB), R0          ; 1315
          50 28 002A0          MOV3 R0, (OLD_ADDRESS), @-8(CCB)          ;
          62          01 88 002A5          BISB2 #1, (R8)          ; 1316
          68          AB DO 002A8          MOVL -20(CCB), 36(CCB)          ; 1321
          24          AB D2 002AD          MOVW -46(CCB), 32(CCB)          ; 1322
          20          AB EC 002B2          MOVL -20(CCB), -100(CCB)          ; 1323
          9C          AB          EB 002B7          CLRL -24(CCB)          ; 1327
          D8          AB          01 90 002BA          MOVB #1, -40(CCB)          ; 1331
          67          01 88 002BE          BISB2 #1, (R7)          ; 1332
          50 00000000G 00 9E 002C1          MOVAB BASS$L_XIT_LOCK, R0          ; 1338
          07          50 E8 002C8          BLBS R0, 27$          ;
00000000G 00          00 FB 002CB          CALLS #0, BASS$DECL_EXITH          ;
          04 002D2 27$:          RET          ; 1341

```

; Routine Size: 723 bytes, Routine Base: \_BASS\$CODE + 0064

```

: 585          1342 1
: 586          1343 1 END
: 587          1344 1
: 588          1345 0 ELUDOM

```

! end of module BASS\$OPEN\_DEFLT

PSECT SUMMARY

Name	Bytes	Attributes
_BASS\$CODE	823	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded		
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	66	581	00:01.1

COMMAND QUALIFIERS

```

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

```

BAS\$\$OPEN\_DEFLT  
1-041

<sup>1</sup>8  
16-Sep-1984 00:57:05

VAX-11 Bliss-32 V4.0-742

Page 17

: Size: 723 code + 100 data bytes  
: Run Time: 00:20.1  
: Elapsed Time: 00:47.0  
: Lines/CPU Min: 4006  
: Lexemes/CPU-Min: 28838  
: Memory Used: 270 pages  
: Compilation Complete

