

|              |     |              |     |            |     |            |     |            |  |              |
|--------------|-----|--------------|-----|------------|-----|------------|-----|------------|--|--------------|
| BBBBBBBBBBBB |     | AAAAAAAAA    |     | SSSSSSSSSS |     | RRRRRRRRRR |     | TTTTTTTTTT |  | LLL          |
| BBBBBBBBBBBB |     | AAAAAAAAA    |     | SSSSSSSSSS |     | RRRRRRRRRR |     | TTTTTTTTTT |  | LLL          |
| BBBBBBBBBBBB |     | AAAAAAAAA    |     | SSSSSSSSSS |     | RRRRRRRRRR |     | TTTTTTTTTT |  | LLL          |
| BBB          | BBB | AAA          | AAA | SSS        |     | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAA          | AAA | SSS        |     | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAA          | AAA | SSS        |     | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAA          | AAA | SSS        |     | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAA          | AAA | SSS        |     | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAA          | AAA | SSS        |     | RRR        | RRR | TTT        |  | LLL          |
| BBBBBBBBBBBB |     | AAA          | AAA | SSSSSSSS   |     | RRRRRRRRRR |     | TTT        |  | LLL          |
| BBBBBBBBBBBB |     | AAA          | AAA | SSSSSSSS   |     | RRRRRRRRRR |     | TTT        |  | LLL          |
| BBBBBBBBBBBB |     | AAA          | AAA | SSSSSSSS   |     | RRRRRRRRRR |     | TTT        |  | LLL          |
| BBB          | BBB | AAAAAAAAAAAA |     |            | SSS | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAAAAAAAAAAA |     |            | SSS | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAAAAAAAAAAA |     |            | SSS | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAA          | AAA |            | SSS | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAA          | AAA |            | SSS | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAA          | AAA |            | SSS | RRR        | RRR | TTT        |  | LLL          |
| BBB          | BBB | AAA          | AAA |            | SSS | RRR        | RRR | TTT        |  | LLL          |
| BBBBBBBBBBBB |     | AAA          | AAA | SSSSSSSS   |     | RRR        | RRR | TTT        |  | LLLLLLLLLLLL |
| BBBBBBBBBBBB |     | AAA          | AAA | SSSSSSSS   |     | RRR        | RRR | TTT        |  | LLLLLLLLLLLL |
| BBBBBBBBBBBB |     | AAA          | AAA | SSSSSSSS   |     | RRR        | RRR | TTT        |  | LLLLLLLLLLLL |

```
BBBBBBBBB      AAAAAA      SSSSSSSS      IIIIII      NN      NN      IIIIII      TTTTTTTTTT
BBBBBBBBB      AAAAAA      SSSSSSSS      IIIIII      NN      NN      IIIIII      TTTTTTTTTT
BB      BB      AA      AA      SS      II      NN      NN      II      TT
BB      BB      AA      AA      SS      II      NN      NN      II      TT
BB      BB      AA      AA      SS      II      NNNN      NN      II      TT
BBBBBBBBB      AA      AA      SSSSSS      II      NN      NN      II      TT
BBBBBBBBB      AA      AA      SSSSSS      II      NN      NN      II      TT
BB      BB      AAAAAAAAAA      SS      II      NN      NN      II      TT
BB      BB      AAAAAAAAAA      SS      II      NN      NN      II      TT
BB      BB      AA      AA      SS      II      NN      NN      II      TT
BBBBBBBBB      AA      AA      SSSSSSSS      IIIIII      NN      NN      IIIIII      TT
BBBBBBBBB      AA      AA      SSSSSSSS      IIIIII      NN      NN      IIIIII      TT
                                         ....
                                         ....
                                         ....
                                         ....

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLL      IIIIII      SSSSSSSS
```

```
1 0001 0 MODULE BASSINIT (
2 0002 0 IDENT = '1-021'
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 Frame Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 These routines set up and tear down frames for BASIC-PLUS-2.
36 0036 1 Frames are used for main routines, external functions,
37 0037 1 external subroutines, internal functions (both DEFs and DEF*s)
38 0038 1 internal subroutines (GOSUBs) and condition handlers.
39 0039 1
40 0040 1 ENVIRONMENT: VAX-11 user mode
41 0041 1
42 0042 1 AUTHOR: John Sauter, CREATION DATE: 10-Oct-78
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1 1-001 - Original.
47 0047 1 1-002 - Clear numeric array elements when allocating them. JBS 31-JAN-1979
48 0048 1 1-003 - Correct a typo in the check for proper number of arguments
49 0049 1 to a subroutine. JBS 02-FEB-1979
50 0050 1 1-004 - Correct some typos in setting up arrays. JBS 05-FEB-1979
51 0051 1 1-005 - Check scale factors and long/double flags in the previous
52 0052 1 major frame, not just the previous frame. JBS 08-FEB-1979
53 0053 1 1-006 - Remove BSFSB_IN_L_FCD. JBS 09-FEB-1979
54 0054 1 1-007 - Use an auxiliary variable when nulling the result string
55 0055 1 to avoid a bug in the BLISS compiler when referring to
56 0056 1 AP directly as a BUILTIN. JBS 12-FEB-1979
57 0057 1 1-008 - Allocate two kinds of strings: dynamic and fixed. JBS 20-MAR-1979
```

BASSINIT  
1-021

F 6  
16-Sep-1984 00:38:23  
14-Sep-1984 11:55:08

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

Page 2  
(1)

```

58 0058 1 1-009 - Do not imply that R11 points to a frame. JBS 08-MAY-1979
59 0059 1 1-010 - Change OTSS$ and LIB$ to STR$. JBS 21-MAY-1979
60 0060 1 1-011 - Use right shifts instead of divides. JBS 11-JUN-1979
61 0061 1 1-012 - Check for correct code in arg list. JBS 03-AUG-1979
62 0062 1 1-013 - Change BASSK_WROMATPAC to BASSK_DIFUSELON. JBS 19-SEP-1979
63 0063 1 1-014 - Add support for run-time dimensioned arrays. PLL 12-May-1982
64 0064 1 1-015 - Check frame version number before executing the new run-time
65 0065 1 array code. PLL 17-May-1982
66 0066 1 1-016 - Check for two frames using the same integer and float types
67 0067 1 should only be done for Basic V1 programs. PLL 20-May-1982
68 0068 1 1-017 - Add only 123 on the stack for V2 variables. Otherwise all the
69 0069 1 BSF$MAJOR FRAME offsets will be wrong. PLL 3-Jun-1982
70 0070 1 1-018 - Set up BSF$A_USER_HAND from R11, not FP. PLL 4-Jun-1982
71 0071 1 1-019 - The compiler allocates contiguous space for array descriptors
72 0072 1 and run-time array descriptors. The start address for the
73 0073 1 rt arrays can be calculated from the start address of all array
74 0074 1 descriptors and the offset for rt descriptors (passed by the
75 0075 1 compiler). PLL 10-Aug-1982
76 0076 1 1-020 - Only set BSF$A_RTA_DESC if this is a V2 program. PLL 11-Aug-1982
77 0077 1 1-021 - Allow result parameter passed to a STRING FUNCTION to be a static
78 0078 1 string (so other lang's w/o dynamic strings can call such functions.
79 0079 1 DG 14-Feb-1984
80 0080 1 --
81 0081 1
82 0082 1
83 0083 1 !<BLF/PAGE>
```

```

85 0084 1 |
86 0085 1 | SWITCHES:
87 0086 1 |
88 0087 1 |
89 0088 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
90 0089 1 |
91 0090 1 |
92 0091 1 | LINKAGES:
93 0092 1 |
94 0093 1 |
95 0094 1 | LINKAGE
96 0095 1 |     BASSINIT_LINK = JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2) :
97 0096 1 |     GLOBAL (BSF$A_MAJOR_STG = 11, BSF$A_MINOR_STG = 10, BSF$A_TEMP_STG = 9) !
98 0097 1 |     NOPRESERVE (8, 7, 6, 5, 4, 3, 2, 1, 0);
99 0098 1 |
100 0099 1 |
101 0100 1 | TABLE OF CONTENTS:
102 0101 1 |
103 0102 1 |
104 0103 1 | FORWARD ROUTINE
105 0104 1 |     BASSINIT_R8 : NOVALUE BASSINIT_LINK; ! start major frame
106 0105 1 |
107 0106 1 |
108 0107 1 | INCLUDE FILES:
109 0108 1 |
110 0109 1 |
111 0110 1 | LIBRARY 'RTLSTARLE'; ! symbols for strings
112 0111 1 |
113 0112 1 | REQUIRE 'RTLIN:RTLPSECT'; ! macros for defining psects
114 0207 1 |
115 0208 1 | REQUIRE 'RTLIN:BASFRAME'; ! Define frame structure
116 0411 1 |
117 0412 1 | REQUIRE 'RTLIN:BASINARG'; ! Define argument list
118 0496 1 |
119 0497 1 |
120 0498 1 | MACROS:
121 0499 1 |
122 0500 1 |     NONE
123 0501 1 |
124 0502 1 | EQUATED SYMBOLS:
125 0503 1 |
126 0504 1 |     NONE
127 0505 1 |
128 0506 1 | PSECTS:
129 0507 1 |
130 0508 1 | DECLARE_PSECTS (BAS); ! declare psects for BASS facility
131 0509 1 |
132 0510 1 | OWN STORAGE:
133 0511 1 |
134 0512 1 |     NONE
135 0513 1 |
136 0514 1 | EXTERNAL REFERENCES:
137 0515 1 |
138 0516 1 |
139 0517 1 | EXTERNAL ROUTINE
140 0518 1 |     STR$FREE1 DX, ! frees a dynamic string
141 0519 1 |     BASS$SIGNAL : NOVALUE, ! signals error
```

BASSINIT  
1-021

H 6  
16-Sep-1984 00:38:23  
14-Sep-1984 11:55:08

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

Page 4  
(2)

```
: 142      0520 1      BASSHANDLER;                      ! handles signals
: 143      0521 1
: 144      0522 1
: 145      0523 1      ! The following are the error codes used in this module.
: 146      0524 1
: 147      0525 1
: 148      0526 1      EXTERNAL LITERAL
: 149      0527 1      BASSK_TOOFEWARG : UNSIGNED (8),      ! Too few arguments
: 150      0528 1      BASSK_TOOMANARG : UNSIGNED (8),      ! Too many arguments
: 151      0529 1      BASSK_SCAFACINT : UNSIGNED (8),      ! Scale factor interlock
: 152      0530 1      BASSK_DIFUSELON : UNSIGNED (8),      ! /LONG, /DOUBLE problems
: 153      0531 1      BASSK_ARGDONMAT : UNSIGNED (8),      ! Arguments don't match
: 154      0532 1      BASSK_NOTIMP : UNSIGNED (8);         ! Not implemented
: 155      0533 1
```

```
157 0534 1 GLOBAL ROUTINE BASSINIT_R8 (
158 0535 1     ARGLIST,
159 0536 1     DATA_RELOC,
160 0537 1     CODE_RELOC
161 0538 1 ) : NOVALUE BASSINIT_LINK =
162 0539 1
163 0540 1 ++
164 0541 1 FUNCTIONAL DESCRIPTION:
165 0542 1
166 0543 1     Set up a frame for a BASIC-PLUS-2 major procedure. The frame
167 0544 1     is allocated on the stack, and R11 and R9 are set up to point
168 0545 1     to it. The argument list tells how to do the allocation.
169 0546 1
170 0547 1 FORMAL PARAMETERS:
171 0548 1
172 0549 1     ARGLIST.rx.v List of information needed to set up the
173 0550 1     frame. See BASIC-PLUS-2/VAX Description
174 0551 1     of Generated Code for details.
175 0552 1     DATA_RELOC.ra.v Address of this procedure's data. Data offsets
176 0553 1     in the argument list are based on this value.
177 0554 1     CODE_RELOC.ra.v Address of this procedure's code. Offsets in
178 0555 1     the PC delta table are based on this value.
179 0556 1
180 0557 1 IMPLICIT INPUTS:
181 0558 1
182 0559 1     Some information from the previous frame, if it is a
183 0560 1     BASIC frame.
184 0561 1
185 0562 1 IMPLICIT OUTPUTS:
186 0563 1
187 0564 1     The values of R11 and R9, which point to the automatic
188 0565 1     storage and the temporary storage, respectively.
189 0566 1
190 0567 1 ROUTINE VALUE:
191 0568 1
192 0569 1     NONE
193 0570 1
194 0571 1 COMPLETION CODES:
195 0572 1
196 0573 1     NONE
197 0574 1
198 0575 1 SIDE EFFECTS:
199 0576 1
200 0577 1     Leaves lots of things on the stack for use by the compiled
201 0578 1     BASIC-PLUS-2 code. These things will be removed by BASSEND_R8.
202 0579 1
203 0580 1 --
204 0581 1
205 0582 2 BEGIN
206 0583 2
207 0584 2 EXTERNAL REGISTER
208 0585 2     BSF$A_MAJOR_STG : REF BLOCK [, BYTE] FIELD (BSF$MAJOR_FRAME),
209 0586 2     BSF$A_MINOR_STG,
210 0587 2     BSF$A_TEMP_STG;
211 0588 2
212 0589 2 BUILTIN
213 0590 2     AP,
```

```
214 0591 2      FP,  
215 0592 2      SP,  
216 0593 2      CVFLD;          ! convert 32-bit integer to double floating  
217 0594 2  
218 0595 2      MAP  
219 0596 2      ARGLIST : REF BLOCK [0, BYTE] FIELD (BASSINIT_ARGS), ! arg list  
220 0597 2      AP : REF VECTOR;          ! caller's arg list  
221 0598 2  
222 0599 2  
223 0600 2      !+ Define local variables as registers. We cannot have any stack locals  
224 0601 2      !- since we manipulate the stack pointer in this routine.  
225 0602 2  
226 0603 2  
227 0604 2      REGISTER  
228 0605 2      RETURN_ADDRESS,          ! address to return to  
229 0606 2      FMP : REF BLOCK [0, BYTE] FIELD (BSF$FCD), ! pointer to FCD  
230 0607 2      PREV_FMP : REF BLOCK [0, BYTE] FIELD (BSF$FCD), ! a previous frame  
231 0608 2      ARRAY_DESC : REF BLOCK [0, BYTE], ! pointer to build array descriptors  
232 0609 2      ARRAY_INDEX, ! index for array modification  
233 0610 2      FCD_LEN; ! length of major frame  
234 0611 2      ! (diff for V1 and V2)  
235 0612 2  
236 0613 2      !+ Save return address because we are going to fool with the stack  
237 0614 2      !-  
238 0615 2      RETURN_ADDRESS = ..SP;  
239 0616 2  
240 0617 2      !+ Allocate frame control data.  
241 0618 2      !-  
242 0619 2      FMP = .FP;  
243 0620 2  
244 0621 2      CASE .ARGLIST [BAS$B_IN_V_FCD] FROM 1 TO 2 OF  
245 0622 2      SET  
246 0623 2      [1] :  
247 0624 2      FCD_LEN = BSF$K_LENFCDMAJ;  
248 0625 2  
249 0626 2      [2] :  
250 0627 2      FCD_LEN = BSF$K_LENFCDMAJ2;  
251 0628 2  
252 0629 2      [OUTRANGE] :  
253 0630 2      BAS$$$SIGNAL (BAS$K_NOTIMP);  
254 0631 2      TES;  
255 0632 2  
256 0633 2      SP = .FMP - .FCD_LEN;  
257 0634 2  
258 0635 2  
259 0636 2  
260 0637 2      !+  
261 0638 2      LOAD Rn (R11)  
262 0639 2      Notice that first the frame is allocated, then an arbitrary amount of  
263 0640 2      storage which is easily addressed by byte offsets. This amount of storage  
264 0641 2      has been reduced for V2 by the amount that the frame was increased so that  
265 0642 2      offsets from FP (BSF$FCD defs) and from R11 (BSF$MAJOR_FRAME) will continue  
266 0643 2      to work for both versions 1 and 2.  
267 0644 2      !-  
268 0645 2      CASE .ARGLIST [BAS$B_IN_V_FCD] FROM 1 TO 2 OF  
269 0646 2      SET  
270 0647 2      [1]:
```

```
271 0648 BSFSA_MAJOR_STG = .SP - 127;
272 0649 [2]:
273 0650 BSFSA_MAJOR_STG = .SP - 123;
274 0651 [OUTRANGE]:
275 0652 BAS$$SIGNAL (BAS$K_NOTIMP);
276 0653 TES;
277 0654 BSFSA_MINOR_STG = 0;
278 0655
279 0656
280 0657 Init BSFSA_USER_HAND.
281 0658 It is initialized to 0 normally (ON ERROR GOTO 0), but if the
282 0659 first statement in the program is ON ERROR GOTO <line number>
283 0660 or ON ERROR GO BACK, it is initialized to 1 (ON ERROR GO BACK)
284 0661 to prevent a "window" in which error handling is ON ERROR GOTO 0
285 0662 no matter what the user wants.
286 0663
287 0664 IF ((.ARGLIST [BAS$W_IN_FLAGS] AND BSF$M_FCD_OEGO) NEQ 0)
288 0665 THEN
289 0666 .BSFSA_MAJOR_STG + 127 = 1
290 0667 ELSE
291 0668 .BSFSA_MAJOR_STG + 127 = 0;
292 0669
293 0670
294 0671 Init parts of the frame control data.
295 0672
296 0673 FMP [BSFSA_MARK] = 0;
297 0674 FMP [BSFSA_BASE_R11] = .BSFSA_MAJOR_STG;
298 0675 FMP [BSFSA_BASE_R10] = .BSFSA_MINOR_STG;
299 0676 FMP [BSF$B_LEN_FCD] = .FCD_LEN;
300 0677 FMP [BSF$B_PROC_CODE] = .ARGLIST [BAS$B_IN_PROC_C];
301 0678 FMP [BSF$W_FCD_FLAGS] = .ARGLIST [BAS$W_IN_FLAGS];
302 0679 FMP [BSFSA_PROC_ID] = .ARGLIST [BAS$L_IN_PROC_I] + .DATA_RELOC;
303 0680 FMP [BSFSA_INIT_ARG] = .ARGLIST;
304 0681 FMP [BSF$L_INIT_REL] = .DATA_RELOC;
305 0682
306 0683 Allocate numeric scalars. They are all initialized to zero.
307 0684
308 0685
309 0686 INCR COUNTER FROM 1 TO .ARGLIST [BAS$L_IN_LEN_SC] DO
310 0687 BEGIN
311 0688 SP = .SP - %UPVAL;
312 0689 .SP = 0;
313 0690 END;
314 0691
315 0692
316 0693 Copy formals.
317 0694
318 0695
319 0696 DECR COUNTER FROM MIN (.ARGLIST [BAS$B_IN_NO_FML], ((.AP [0]) AND 255)) TO 1 DO
320 0697 BEGIN
321 0698 SP = .SP - %UPVAL;
322 0699 .SP = .AP [.COUNTER];
323 0700 END;
324 0701
325 0702
326 0703 Allocate and initialize descriptors.
327 0704
```

```
328 0705 2 SP = .SP - .ARGLIST [BAS$L_IN_LEN_DT];
329 0706 2
330 0707 2 + Set ARRAY_DESC to point to the space allocated.
331 0708 2
332 0709 2 - ARRAY_DESC = .SP;
333 0710 2
334 0711 2 +
335 0712 2 Calculate the start address of the run-time array descriptors
336 0713 2 from the start address of all array descriptors and the offset
337 0714 2 to the rt descriptors.
338 0715 2 -
339 0716 2 IF (.ARGLIST [BAS$B_IN_V_FCD] GTR 1) AND !make sure rt arrays supported
340 0717 2 (.ARGLIST [BAS$L_IN_LEN_RT_DT] NEQ 0)
341 0718 2 THEN
342 0719 2 FMP [BSF$A_RT_DESC] = .SP + (.ARGLIST [BAS$L_IN_RT_DT_TMT] -
343 0720 2 .ARGLIST [BAS$L_IN_DT_TMT]);
344 0721 2
345 0722 2 +
346 0723 2 Load the space from the template and then modify it based
347 0724 2 on the modification table.
348 0725 2 -
349 0726 2
350 0727 2 INCR COUNTER FROM 0 TO ((.ARGLIST [BAS$L_IN_LEN_DT]^2) - 1) DO
351 0728 2 BEGIN
352 0729 2 ARRAY_DESC [.COUNTER*%UPVAL, 0, %BPVAL, 0] = .((.ARGLIST [BAS$L_IN_DT_TMT]) + !
353 0730 2 .DATA_RELOC + (.COUNTER*%UPVAL));
354 0731 2 END;
355 0732 2
356 0733 2 +
357 0734 2 Now modify the descriptors. These are usually array descriptors.
358 0735 2 -
359 0736 2
360 0737 2 INCR COUNTER FROM 0 TO (.ARGLIST [BAS$L_IN_LEN_DM] - 1) DO
361 0738 2 BEGIN
362 0739 2 ARRAY_INDEX = .((.ARGLIST [BAS$L_IN_DT_MOD]) + .DATA_RELOC + (.COUNTER*%UPVAL));
363 0740 2 BSF$A_MAJOR_STG [.ARRAY_INDEX, 0, %BPVAL, 0] = !
364 0741 2 = .BSF$A_MAJOR_STG [.ARRAY_INDEX, 0, %BPVAL, 0] + .BSF$A_MAJOR_STG;
365 0742 2 END;
366 0743 2
367 0744 2 +
368 0745 2 Allocate dynamic string descriptors.
369 0746 2 -
370 0747 2
371 0748 2 INCR COUNTER FROM 1 TO .ARGLIST [BAS$W_IN_NO_DST] DO
372 0749 2 BEGIN
373 0750 2 SP = .SP - %UPVAL;
374 0751 2 .SP = 0; ! Pointer 0 implies not allocated.
375 0752 2 SP = .SP - %UPVAL;
376 0753 2 BLOCK [.SP, DSC$B_CLASS; 0, BYTE] = DSC$K_CLASS_D; ! dynamic
377 0754 2 BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T; ! text
378 0755 2 BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0; ! length = 0
379 0756 2 END;
380 0757 2
381 0758 2 FMP [BSF$A_STR_DESC] = .SP;
382 0759 2 +
383 0760 2 Allocate fixed string templates.
384 0761 2 -
```

```
385 0762 2
386 0763 2 INCR COUNTER FROM 1 TO .ARGLIST [BAS$W_IN_NO_FST] DO
387 0764 2 BEGIN
388 0765 2 SP = .SP - %UPVAL;
389 0766 2 .SP = 0; ! Pointer 0 implies not allocated.
390 0767 2 SP = .SP - %UPVAL;
391 0768 2 BLOCK [.SP, DSC$B_CLASS; 0, BYTE] = DSC$K_CLASS_S; ! fixed
392 0769 2 BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T; ! text
393 0770 2 BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0; ! length = 0
394 0771 2 END;
395 0772 2
396 0773 2 !+
397 0774 2 Allocate numeric array elements. They are all initialized to zero.
398 0775 2 !-
399 0776 2
400 0777 2 INCR COUNTER FROM 1 TO (.ARGLIST [BAS$L_IN_LEN_NA]^2) DO
401 0778 2 BEGIN
402 0779 2 SP = .SP - %UPVAL;
403 0780 2 .SP = 0;
404 0781 2 END;
405 0782 2
406 0783 2 !+
407 0784 2 Allocate temporary cells.
408 0785 2 !-
409 0786 2
410 0787 2 IF ((.ARGLIST [BAS$L_IN_NO_TST] NEQ 0) OR (.ARGLIST [BAS$L_IN_NO_NMT] NEQ 0))
411 0788 2 THEN
412 0789 2 BEGIN
413 0790 2 !+
414 0791 2 We must set up R9. First allocate string temporaries.
415 0792 2 !-
416 0793 2
417 0794 2 INCR COUNTER FROM 1 TO .ARGLIST [BAS$L_IN_NO_TST] DO
418 0795 2 BEGIN
419 0796 2 SP = .SP - %UPVAL;
420 0797 2 .SP = 0; ! Pointer 0 implies not allocated.
421 0798 2 SP = .SP - %UPVAL;
422 0799 2 BLOCK [.SP, DSC$B_CLASS; 0, BYTE] = DSC$K_CLASS_D; ! dynamic
423 0800 2 BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T; ! text
424 0801 2 BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0; ! length = 0
425 0802 2 END;
426 0803 2
427 0804 2 !+
428 0805 2 Point R9 to the last string descriptor allocated.
429 0806 2 !-
430 0807 2 BSF$A_TEMP_STG = .SP;
431 0808 2 !+
432 0809 2 Now allocate numeric temporaries.
433 0810 2 !-
434 0811 2 SP = .SP - .ARGLIST [BAS$L_IN_NO_NMT];
435 0812 2 END;
436 0813 2
437 0814 2 !+
438 0815 2 Store the value of R9, whether or not loaded, in the frame.
439 0816 2 !-
440 0817 2 FMP [BSF$A_BASE_R9] = .BSF$A_TEMP_STG;
441 0818 2 !+
```

```

442 0819 2 ! Set up major frame information.
443 0820 2 !-
444 0821 2
445 0822 2 CASE .ARGLIST [BASSB_IN_S_V_DB] FROM -6 TO 0 OF
446 0823 2 SET
447 0824 2
448 0825 2 [0] :
449 0826 2 CVTLD (%REF (1), FMP [BSF$D_SCALE_DOU]);
450 0827 2
451 0828 2 [-1] :
452 0829 2 CVTLD (%REF (10), FMP [BSF$D_SCALE_DOU]);
453 0830 2
454 0831 2 [-2] :
455 0832 2 CVTLD (%REF (100), FMP [BSF$D_SCALE_DOU]);
456 0833 2
457 0834 2 [-3] :
458 0835 2 CVTLD (%REF (1000), FMP [BSF$D_SCALE_DOU]);
459 0836 2
460 0837 2 [-4] :
461 0838 2 CVTLD (%REF (10000), FMP [BSF$D_SCALE_DOU]);
462 0839 2
463 0840 2 [-5] :
464 0841 2 CVTLD (%REF (100000), FMP [BSF$D_SCALE_DOU]);
465 0842 2
466 0843 2 [-6] :
467 0844 2 CVTLD (%REF (1000000), FMP [BSF$D_SCALE_DOU]);
468 0845 2
469 0846 2 [OUTRANGE] :
470 0847 2 CVTLD (%REF (0), FMP [BSF$D_SCALE_DOU]);
471 0848 2 TES;
472 0849 2
473 0850 2 FMP [BSF$B_SCA_V_DOU] = .ARGLIST [BASSB_IN_S_V_DB];
474 0851 2 FMP [BSF$B_SCA_V_PAC] = .ARGLIST [BASSB_IN_S_V_PK];
475 0852 2 FMP [BSF$A_CUR_DATA] = .ARGLIST [BASSL_IN_BEG_DA] + .DATA_RELOC;
476 0853 2 FMP [BSF$A_END_DATA] = .ARGLIST [BASSL_IN_END_DA] + .DATA_RELOC;
477 0854 2 FMP [BSF$A_BASE_PC] = .CODE_RELOC;
478 0855 2 !+
479 0856 2 ! Complete frame.
480 0857 2 !-
481 0858 2 FMP [BSF$A_BASE_SP] = .SP;
482 0859 2 FMP [BSF$A_HAND[ER]] = BASSHANDLER;
483 0860 2 !+
484 0861 2 ! First consistency checks.
485 0862 2 !-
486 0863 2
487 0864 2 IF (((.AP [0]) AND 255) NEQ .ARGLIST [BASSB_IN_NO_FML])
488 0865 2 THEN
489 0866 2 !+
490 0867 2 ! The number of arguments is incorrect.
491 0868 2 !-
492 0869 2 BEGIN
493 0870 2
494 0871 2 IF (((.AP [0]) AND 255) GTRU .ARGLIST [BASSB_IN_NO_FML])
495 0872 2 THEN
496 0873 2 BEGIN
497 0874 2 !+
498 0875 2 ! Main programs are permitted more arguments than they are declared
```

```
499 0876 4 ! with, to allow old BASIC programs to work with later versions of
500 0877 4 ! the command language interpreter.
501 0878 4 !-
502 0879 4
503 0880 4 IF (.FMP [BSF$B_PROC_CODE] NEQ BSF$K_PROC_MAIN) THEN BAS$$SIGNAL (BAS$K_TOOMANARG);
504 0881 4
505 0882 4 END
506 0883 4 ELSE
507 0884 4 BAS$$SIGNAL (BAS$K_TOOFEWARG);
508 0885 4
509 0886 4 END;
510 0887 4
511 0888 4 IF (((.FMP [BSF$W_FCD_FLAGS]) AND (BSF$M_FCD_RSTR)) NEQ 0)
512 0889 4 THEN
513 0890 4 BEGIN
514 0891 4
515 0892 4 LOCAL
516 0893 4 STR_DESC_ADDR : REF BLOCK [8, BYTE];
517 0894 4
518 0895 4 !+
519 0896 4 This procedure has been marked by the compiler as returning a
520 0897 4 string result. Be sure that there is at least one formal, and
521 0898 4 that it is a dynamic or static string descriptor. If so, null its value.
522 0899 4 !-
523 0900 4
524 0901 4 IF (.ARGLIST [BAS$B_IN_NO_FML] LSSU 1) THEN BAS$$SIGNAL (BAS$K_TOOFEWARG);
525 0902 4
526 0903 4 STR_DESC_ADDR = AP [1];
527 0904 4 STR_DESC_ADDR = ..STR_DESC_ADDR; ! Avoid BLISS compiler bug
528 0905 4
529 0906 4 IF (((.STR_DESC_ADDR [DSC$B_CLASS] NEQU DSC$K_CLASS_D) AND
530 0907 4 (.STR_DESC_ADDR [DSC$B_CLASS] NEQU DSC$K_CLASS_S)) OR
531 0908 4 (.STR_DESC_ADDR [DSC$B_DTYPE] NEQU DSC$K_DTYPE_T))
532 0909 4 THEN
533 0910 4 BAS$$SIGNAL (BAS$K_ARGDONMAT);
534 0911 4
535 0912 4 !+
536 0913 4 Null the string. This insures that, if the procedure does not reference
537 0914 4 the string, the function will have the value of the null string.
538 0915 4 !-
539 0916 4 IF .STR_DESC_ADDR [DSC$B_CLASS] EQLU DSC$K_CLASS_D
540 0917 4 THEN
541 0918 4 STR$FREE1_DX (.STR_DESC_ADDR)
542 0919 4 ELSE
543 0920 4 CH$FILL ('X'0', .STR_DESC_ADDR [DSC$W_LENGTH], .STR_DESC_ADDR [DSC$A_POINTER]);
544 0921 4
545 0922 4 END;
546 0923 4
547 0924 4 !+
548 0925 4 Second consistency checks. If the previous frame is a BASIC frame
549 0926 4 verify that it was compiled with the same options as this one.
550 0927 4 !-
551 0928 4 PREV_FMP = .FMP [BSF$A_SAVED_FP];
552 0929 4
553 0930 4 IF (.PREV_FMP [BSF$A_HANDLER] EQLA BAS$HANDLER)
554 0931 4 THEN
555 0932 4 !+
```

```

556 0933 2 ! The previous frame is a BASIC frame.
557 0934 2 !-
558 0935 2 BEGIN
559 0936 2 !+
560 0937 2 Make sure we are pointing to the major frame. This will be different
561 0938 2 from the previous frame if the call came from a DEF, for example.
562 0939 2 !-
563 0940 2 PREV_FMP = .PREV_FMP [BSF$A_BASE_R11] + %FIELDEXPAND (BSF$FRAME_BASE, 0);
564 0941 2
565 0942 2 IF ((.FMP [BSF$B_SCA_V_PAC] NEQ .PREV_FMP [BSF$B_SCA_V_PAC]) OR !
566 0943 2 (.FMP [BSF$B_SCA_V_DOU] NEQ .PREV_FMP [BSF$B_SCA_V_DOU]))
567 0944 2 THEN
568 0945 2 BAS$$SIGNAL (BAS$K_SCAFACINT);
569 0946 2 END;
570 0947 2 !+
571 0948 2 Put the return address back on the stack so we can return to the
572 0949 2 caller.
573 0950 2 !-
574 0951 2 SP = .SP - %UPVAL;
575 0952 2 .SP = .RETURN_ADDRESS;
576 0953 2 RETURN;
577 0954 2 END;
```

! of BASS\$INIT\_R8

.TITLE BASS\$INIT  
.IDENT \1-021\.EXTRN STR\$FREE1 DX, BAS\$\$SIGNAL  
.EXTRN BASS\$HANDLER, BAS\$K\_TOOFEWARG  
.EXTRN BAS\$K\_TOOMANARG  
.EXTRN BAS\$K\_SCAFACINT  
.EXTRN BAS\$K\_DIFUSELON  
.EXTRN BAS\$K\_ARGDONMAT  
.EXTRN BAS\$K\_NOTIMP

.PSECT \_BAS\$CODE, NOWRT, SHR, PIC, 2

|           |      |                  |                              |  |      |
|-----------|------|------------------|------------------------------|--|------|
| 58        | 51   | DO 00000         | BASS\$INIT R8::              |  |      |
| 54        | 50   | DO 00003         | MOVL R1, R8                  |  | 0534 |
| 56        | 6E   | DO 00006         | MOVL R0, R4                  |  |      |
| 57        | 5D   | DO 00009         | MOVL (SP), RETURN_ADDRESS    |  | 0615 |
| 01        | 01   | A4 8F 0000C      | MOVL FP, FMP                 |  | 0620 |
| 0017      | 04   | 0011 00011 1\$:  | CASEB 4(ARGLIST), #1, #1     |  | 0622 |
|           |      |                  | .WORD 2\$-1\$,-              |  |      |
|           |      |                  | 3\$-1\$                      |  |      |
| 7E        | 00G  | 8F 9A 00015      | MOVZBL #BAS\$K_NOTIMP, -(SP) |  | 0631 |
| 00000000G | 00   | 01 FB 00019      | CALLS #1, BAS\$\$SIGNAL      |  |      |
|           |      | 0A 11 00020      | BRB 4\$                      |  |      |
| 53        | 44   | 8F 9A 00022 2\$: | MOVZBL #68, FCD_LEN          |  | 0625 |
|           |      | 04 11 00026      | BRB 4\$                      |  |      |
| 53        | 48   | 8F 9A 00028 3\$: | MOVZBL #72, FCD_LEN          |  | 0628 |
| 5E        |      | 53 C3 0002C 4\$: | SUBL3 FCD_LEN, FMP, SP       |  | 0634 |
| 01        | 04   | A4 8F 00030      | CASEB 4(ARGLIST), #1, #1     |  | 0645 |
| 0017      | 0011 | 00035 5\$:       | .WORD 6\$-5\$,-              |  |      |
|           |      |                  | 7\$-5\$                      |  |      |
| 7E        | 00G  | 8F 9A 00039      | MOVZBL #BAS\$K_NOTIMP, -(SP) |  | 0652 |
| 00000000G | 00   | 01 FB 0003D      | CALLS #1, BAS\$\$SIGNAL      |  |      |

|    |    |    |    |    |    |       |        |                                      |      |  |
|----|----|----|----|----|----|-------|--------|--------------------------------------|------|--|
|    |    | 5B | 81 | 0A | 11 | 00044 | BRB    | 8\$                                  |      |  |
|    |    |    |    | AE | 9E | 00046 | MOVAB  | -127(SP), BSF\$A_MAJOR_STG           | 0648 |  |
|    |    | 5B | 85 | 04 | 11 | 0004A | BRB    | 8\$                                  |      |  |
|    |    |    |    | AE | 9E | 0004C | MOVAB  | -123(SP), BSF\$A_MAJOR_STG           | 0650 |  |
| 06 |    | 64 |    | 5A | D4 | 00050 | CLRL   | BSF\$A_MINOR_STG                     | 0654 |  |
|    | 7F | AB |    | 3C | E1 | 00052 | BBC    | #60, (ARGLIST), 9\$                  | 0664 |  |
|    |    |    |    | 01 | D0 | 00056 | MOVL   | #1, 127(BSF\$A_MAJOR_STG)            | 0666 |  |
|    |    |    |    | 03 | 11 | 0005A | BRB    | 10\$                                 |      |  |
|    |    |    | 7F | AB | D4 | 0005C | CLRL   | 127(BSF\$A_MAJOR_STG)                | 0668 |  |
|    |    |    | FC | A7 | D4 | 0005F | CLRL   | -4(FMP)                              | 0673 |  |
|    | F0 | A7 |    | 5A | 7D | 00062 | MOVQ   | BSF\$A_MINOR_STG, -16(FMP)           | 0675 |  |
|    | E4 | A7 |    | 53 | 90 | 00066 | MOVB   | FCD_LEN, -28(FMP)                    | 0676 |  |
|    | E5 | A7 | 05 | A4 | 90 | 0006A | MOV8   | 5(ARGLIST), -27(FMP)                 | 0677 |  |
|    | E6 | A7 | 06 | A4 | B0 | 0006F | MOVW   | 6(ARGLIST), -26(FMP)                 | 0678 |  |
|    | E8 | A7 | 08 | B4 | 48 | 9E    | MOVAB  | 28(ARGLIST)[DATA_RELOC], -24(FMP)    | 0679 |  |
|    | D8 | A7 |    | 54 | D0 | 0007A | MOVL   | ARGLIST, -40(FMP)                    | 0680 |  |
|    | DC | A7 |    | 58 | D0 | 0007E | MOVL   | DATA_RELOC, -36(FMP)                 | 0681 |  |
|    |    |    |    | 50 | D4 | 00082 | CLRL   | COUNTER                              | 0686 |  |
|    |    |    |    | 05 | 11 | 00084 | BRB    | 12\$                                 |      |  |
|    |    | 5E |    | 04 | C2 | 00086 | SUBL2  | #4, SP                               | 0688 |  |
|    |    |    |    | 6E | D4 | 00089 | CLRL   | (SP)                                 | 0689 |  |
| F6 |    | 50 | 10 | A4 | F3 | 0008B | AOBLEQ | 16(ARGLIST), COUNTER, 11\$           | 0686 |  |
|    |    | 50 | 14 | A4 | 9A | 00090 | MOVZBL | 20(ARGLIST), R0                      | 0696 |  |
|    |    | 50 |    | 6C | 91 | 00094 | CMPB   | (AP), R0                             |      |  |
|    |    |    |    | 03 | 1E | 00097 | BGEQU  | 13\$                                 |      |  |
|    |    | 50 |    | 6C | 9A | 00099 | MOVZBL | (AP), R0                             |      |  |
|    |    |    |    | 50 | D6 | 0009C | INCL   | COUNTER                              |      |  |
|    |    |    |    | 07 | 11 | 0009E | BRB    | 15\$                                 |      |  |
|    |    | 5E |    | 04 | C2 | 000A0 | SUBL2  | #4, SP                               | 0698 |  |
|    |    | 6E |    | 6C | 40 | D0    | MOVL   | (AP)[COUNTER], (SP)                  | 0699 |  |
|    |    | F6 |    | 50 | F5 | 000A7 | SOBGTR | COUNTER, 14\$                        | 0696 |  |
|    |    | 5E | 18 | A4 | C2 | 000AA | SUBL2  | 24(ARGLIST), SP                      | 0705 |  |
|    |    | 50 |    | 5E | D0 | 000AE | MOVL   | SP, ARRAY_DESC                       | 0709 |  |
|    |    | 01 | 04 | A4 | 91 | 000B1 | CMPB   | 4(ARGLIST), #1                       | 0716 |  |
|    |    |    |    | 10 | 1B | 000B5 | BLEQU  | 16\$                                 |      |  |
|    |    |    | 40 | A4 | D5 | 000B7 | TSTL   | 64(ARGLIST)                          | 0717 |  |
|    |    |    |    | 0B | 13 | 000BA | BEQL   | 16\$                                 |      |  |
|    |    |    | 1C | A4 | C3 | 000BC | SUBL3  | 28(ARGLIST), 68(ARGLIST), R1         | 0720 |  |
| B8 | 51 | 44 |    | 5E | C1 | 000C2 | ADDL3  | SP, R1, -72(FMP)                     | 0719 |  |
|    | A7 |    |    | 8F | 78 | 000C7 | ASHL   | #-2, 24(ARGLIST), R1                 | 0727 |  |
|    | 51 | 18 | FE | 01 | CE | 000CD | MNEGL  | #1, COUNTER                          | 0729 |  |
|    |    |    |    | 0A | 11 | 000D0 | BRB    | 18\$                                 |      |  |
|    | 53 | 58 | 1C | A4 | C1 | 000D2 | ADDL3  | 28(ARGLIST), DATA_RELOC, R3          | 0730 |  |
|    |    | 60 |    | 45 | D0 | 000D7 | MOVL   | (R3)[COUNTER], (ARRAY_DESC)[COUNTER] | 0729 |  |
|    | F2 | 55 |    | 51 | F2 | 000DC | AOBLSS | R1, COUNTER, 17\$                    | 0727 |  |
|    |    | 51 |    | 01 | CE | 000E0 | MNEGL  | #1, COUNTER                          | 0739 |  |
|    |    |    |    | 0F | 11 | 000E3 | BRB    | 20\$                                 |      |  |
|    | 53 | 58 | 24 | A4 | C1 | 000E5 | ADDL3  | 36(ARGLIST), DATA_RELOC, R3          |      |  |
|    |    | 50 |    | 63 | 41 | D0    | MOVL   | (R3)[COUNTER], ARRAY_INDEX           |      |  |
|    |    |    |    | 60 | 4B | 9F    | PUSHAB | (ARRAY_INDEX)[BSF\$A_MAJOR_STG]      | 0741 |  |
|    |    |    |    | 5B | C0 | 000F1 | ADDL2  | BSF\$A_MAJOR_STG, 2(SP)+             |      |  |
|    | EC | 9E | 20 | A4 | F2 | 000F4 | AOBLSS | 32(ARGLIST), COUNTER, 19\$           | 0737 |  |
|    |    | 51 | 28 | A4 | 3C | 000F9 | MOVZWL | 40(ARGLIST), R1                      | 0748 |  |
|    |    |    |    | 50 | D4 | 000FD | CLRL   | COUNTER                              |      |  |
|    |    |    |    | 0F | 11 | 000FF | BRB    | 22\$                                 |      |  |
|    |    | 5E |    | 04 | C2 | 00101 | SUBL2  | #4, SP                               | 0750 |  |
|    |    |    |    | 6E | D4 | 00104 | CLRL   | (SP)                                 | 0751 |  |

|      |      |    |          |      |    |       |       |        |                            |  |      |
|------|------|----|----------|------|----|-------|-------|--------|----------------------------|--|------|
|      |      | 5E |          | 04   | C2 | 00106 |       | SUBL2  | #4, SP                     |  | 0752 |
|      |      | 6E | 020E0000 | 8F   | D0 | 00109 |       | MOVL   | #34471936, (SP)            |  | 0755 |
| ED   |      | 50 |          | 51   | F3 | 00110 | 22\$: | AOBLEQ | R1, COUNTER, 21\$          |  | 0748 |
|      | E0   | A7 |          | 5E   | D0 | 00114 |       | MOVL   | SP, -32(FMP)               |  | 0758 |
|      |      | 51 | 2A       | A4   | 3C | 00118 |       | MOVZWL | 42(ARGLIST), R1            |  | 0763 |
|      |      |    |          | 50   | D4 | 0011C |       | CLRL   | COUNTER                    |  |      |
|      |      |    |          | 0F   | 11 | 0011E |       | BRB    | 24\$                       |  |      |
|      |      | 5E |          | 04   | C2 | 00120 | 23\$: | SUBL2  | #4, SP                     |  | 0765 |
|      |      |    |          | 6E   | D4 | 00123 |       | CLRL   | (SP)                       |  | 0766 |
|      |      | 5E |          | 04   | C2 | 00125 |       | SUBL2  | #4, SP                     |  | 0767 |
|      |      | 6E | 010E0000 | 8F   | D0 | 00128 |       | MOVL   | #17694720, (SP)            |  | 0770 |
| ED   |      | 50 |          | 51   | F3 | 0012F | 24\$: | AOBLEQ | R1, COUNTER, 23\$          |  | 0763 |
| 50   | 2C   | A4 | FE       | 8F   | 78 | 00133 |       | ASHL   | #-2, 44(ARGLIST), R0       |  | 0777 |
|      |      |    |          | 51   | D4 | 00139 |       | CLRL   | COUNTER                    |  |      |
|      |      |    |          | 05   | 11 | 0013B |       | BRB    | 26\$                       |  |      |
|      |      | 5E |          | 04   | C2 | 0013D | 25\$: | SUBL2  | #4, SP                     |  | 0779 |
|      |      |    |          | 6E   | D4 | 00140 |       | CLRL   | (SP)                       |  | 0780 |
| F7   |      | 51 |          | 50   | F3 | 00142 | 26\$: | AOBLEQ | R0, COUNTER, 25\$          |  | 0777 |
|      |      |    | 30       | A4   | D5 | 00146 |       | TSTL   | 48(ARGLIST)                |  | 0787 |
|      |      |    |          | 05   | 12 | 00149 |       | BNEQ   | 27\$                       |  |      |
|      |      |    | 34       | A4   | D5 | 0014B |       | TSTL   | 52(ARGLIST)                |  |      |
|      |      |    |          | 1F   | 13 | 0014E |       | BEQL   | 30\$                       |  |      |
|      |      |    |          | 50   | D4 | 00150 | 27\$: | CLRL   | COUNTER                    |  | 0794 |
|      |      |    |          | 0F   | 11 | 00152 |       | BRB    | 29\$                       |  |      |
|      |      | 5E |          | 04   | C2 | 00154 | 28\$: | SUBL2  | #4, SP                     |  | 0796 |
|      |      |    |          | 6E   | D4 | 00157 |       | CLRL   | (SP)                       |  | 0797 |
|      |      | 5E |          | 04   | C2 | 00159 |       | SUBL2  | #4, SP                     |  | 0798 |
|      |      | 6E | 020E0000 | 8F   | D0 | 0015C |       | MOVL   | #34471936, (SP)            |  | 0801 |
| EC   |      | 50 | 30       | A4   | F3 | 00163 | 29\$: | AOBLEQ | 48(ARGLIST), COUNTER, 28\$ |  | 0794 |
|      |      | 59 |          | 5E   | D0 | 00168 |       | MOVL   | SP, BSFSA TEMP STG         |  | 0807 |
|      |      | 5E | 34       | A4   | C2 | 0016B |       | SUBL2  | 52(ARGLIST), SP            |  | 0811 |
|      | EC   | A7 |          | 59   | D0 | 0016F | 30\$: | MOVL   | BSFSA TEMP STG, -20(FMP)   |  | 0817 |
|      |      | 50 | D0       | A7   | 9E | 00173 |       | MOVAB  | -48(FMP), R0               |  | 0847 |
|      | 06   | 8F | OD       | A4   | 8F | 00177 |       | CASEB  | 13(ARGLIST), #-6, #6       |  | 0822 |
| 0026 | 002F |    |          |      |    |       | 31\$: | .WORD  | 38\$-31\$,-                |  |      |
| 0013 | 0018 |    |          | 0041 |    | 0017D |       |        | 37\$-31\$,-                |  |      |
|      |      |    |          | 001D |    | 00185 |       |        | 36\$-31\$,-                |  |      |
|      |      |    |          |      |    |       |       |        | 35\$-31\$,-                |  |      |
|      |      |    |          |      |    |       |       |        | 34\$-31\$,-                |  |      |
|      |      |    |          |      |    |       |       |        | 33\$-31\$,-                |  |      |
|      |      |    |          |      |    |       |       |        | 32\$-31\$                  |  |      |
|      |      | 60 |          | 00   | 6E | 0018B |       | CVTLD  | #0, (R0)                   |  | 0847 |
|      |      |    |          | 35   | 11 | 0018E |       | BRB    | 39\$                       |  |      |
|      |      | 60 |          | 01   | 6E | 00190 | 32\$: | CVTLD  | #1, (R0)                   |  | 0826 |
|      |      |    |          | 30   | 11 | 00193 |       | BRB    | 39\$                       |  |      |
|      |      | 60 |          | 0A   | 6E | 00195 | 33\$: | CVTLD  | #10, (R0)                  |  | 0829 |
|      |      |    |          | 2B   | 11 | 00198 |       | BRB    | 39\$                       |  |      |
|      |      | 60 | 00000064 | 8F   | 6E | 0019A | 34\$: | CVTLD  | #100, (R0)                 |  | 0832 |
|      |      |    |          | 22   | 11 | 001A1 |       | BRB    | 39\$                       |  |      |
|      |      | 60 | 000003E8 | 8F   | 6E | 001A3 | 35\$: | CVTLD  | #1000, (R0)                |  | 0835 |
|      |      |    |          | 19   | 11 | 001AA |       | BRB    | 39\$                       |  |      |
|      |      | 60 | 00002710 | 8F   | 6E | 001AC | 36\$: | CVTLD  | #10000, (R0)               |  | 0838 |
|      |      |    |          | 10   | 11 | 001B3 |       | BRB    | 39\$                       |  |      |
|      |      | 60 | 000186A0 | 8F   | 6E | 001B5 | 37\$: | CVTLD  | #100000, (R0)              |  | 0841 |
|      |      |    |          | 07   | 11 | 001BC |       | BRB    | 39\$                       |  |      |
|      |      | 60 | 000F4240 | 8F   | 6E | 001BE | 38\$: | CVTLD  | #1000000, (R0)             |  | 0844 |
| CC   |      | A7 | OC       | A4   | B0 | 001C5 | 39\$: | MOVW   | 12(ARGLIST), -52(FMP)      |  | 0851 |

|    |           |    |           |          |       |       |        |                                    |      |
|----|-----------|----|-----------|----------|-------|-------|--------|------------------------------------|------|
|    | C4        | A7 | 38        | B448     | 9E    | 001CA | MOVAB  | @56(ARGLIST)[DATA_RELOC], -60(FMP) | 0852 |
|    | C8        | A7 | 3C        | B448     | 9E    | 001D0 | MOVAB  | @60(ARGLIST)[DATA_RELOC], -56(FMP) | 0853 |
|    | C0        | A7 |           | 52       | D0    | 001D6 | MOVL   | CODE_RELOC, -64(FMP)               | 0854 |
|    | F8        | A7 |           | 5E       | D0    | 001DA | MOVL   | SP, -8(FMP)                        | 0858 |
|    | 14        | A4 | 00000000G | 00       | 9E    | 001DE | MOVAB  | BASSHANDLER, (FMP)                 | 0859 |
|    |           |    |           | 6C       | 91    | 001E5 | CMPB   | (AP), 20(ARGLIST)                  | 0864 |
|    |           |    |           | 19       | 13    | 001E9 | BEQL   | 42\$                               |      |
|    |           |    |           | 0C       | 1B    | 001EB | BLEQU  | 40\$                               | 0871 |
|    |           | 01 | E5        | A7       | 91    | 001ED | CMPB   | -27(FMP), #1                       | 0880 |
|    |           |    |           | 11       | 13    | 001F1 | BEQL   | 42\$                               |      |
|    |           | 7E | 00G       | 8F       | 9A    | 001F3 | MOVZBL | #BASSK_TOOMANARG, -(SP)            |      |
|    |           |    |           | 04       | 11    | 001F7 | BRB    | 41\$                               |      |
|    |           | 7E | 00G       | 8F       | 9A    | 001F9 | MOVZBL | #BASSK_TOOFEWARG, -(SP)            | 0884 |
|    | 00000000G | 00 |           | 01       | FB    | 001FD | CALLS  | #1, BASS\$SIGNAL                   |      |
| 4C | E6        | A7 |           | 0D       | E1    | 00204 | BBC    | #13, -26(FMP), 48\$                | 0888 |
|    |           |    | 14        | A4       | 95    | 00209 | TSTB   | 20(ARGLIST)                        | 0901 |
|    |           |    |           | 0B       | 12    | 0020C | BNEQ   | 43\$                               |      |
|    |           | 7E | 00G       | 8F       | 9A    | 0020E | MOVZBL | #BASSK_TOOFEWARG, -(SP)            |      |
|    | 00000000G | 00 |           | 01       | FB    | 00212 | CALLS  | #1, BASS\$SIGNAL                   |      |
|    |           | 52 | 04        | AC       | 9E    | 00219 | MOVAB  | 4(AP), STR_DESC_ADDR               | 0903 |
|    |           | 52 |           | 62       | D0    | 0021D | MOVL   | (STR_DESC_ADDR), STR_DESC_ADDR     | 0904 |
|    |           | 02 | 03        | A2       | 91    | 00220 | CMPB   | 3(STR_DESC_ADDR), #2               | 0906 |
|    |           |    |           | 06       | 13    | 00224 | BEQL   | 44\$                               |      |
|    |           | 01 | 03        | A2       | 91    | 00226 | CMPB   | 3(STR_DESC_ADDR), #1               | 0907 |
|    |           |    |           | 06       | 12    | 0022A | BNEQ   | 45\$                               |      |
|    |           | 0E | 02        | A2       | 91    | 0022C | CMPB   | 2(STR_DESC_ADDR), #14              | 0908 |
|    |           |    |           | 0B       | 13    | 00230 | BEQL   | 46\$                               |      |
|    |           | 7E | 00G       | 8F       | 9A    | 00232 | MOVZBL | #BASSK_ARGDONMAT, -(SP)            | 0910 |
|    | 00000000G | 00 |           | 01       | FB    | 00236 | CALLS  | #1, BASS\$SIGNAL                   |      |
|    |           | 02 | 03        | A2       | 91    | 0023D | CMPB   | 3(STR_DESC_ADDR), #2               | 0916 |
|    |           |    |           | 0B       | 12    | 00241 | BNEQ   | 47\$                               |      |
|    |           |    |           | 52       | DD    | 00243 | PUSHL  | STR_DESC_ADDR                      | 0918 |
|    | 00000000G | 00 |           | 01       | FB    | 00245 | CALLS  | #1, -STR\$FREE1_DX                 |      |
|    |           |    |           | 07       | 11    | 0024C | BRB    | 48\$                               |      |
| 62 | 00        | 6E |           | 00       | 2C    | 0024E | MOVCS  | #0, (SP), #0, (STR_DESC_ADDR), -   | 0920 |
|    |           |    | 04        | B2       |       | 00253 |        | @4(STR_DESC_ADDR)                  |      |
|    |           | 50 | 0C        | A7       | D0    | 00255 | MOVL   | 12(FMP), PREV_FMP                  | 0928 |
|    |           | 51 | 00000000G | 00       | 9E    | 00259 | MOVAB  | BASSHANDLER, R1                    | 0930 |
|    |           | 51 |           | 60       | D1    | 00260 | CMPL   | (PREV_FMP), R1                     |      |
|    |           |    |           | 22       | 12    | 00263 | BNEQ   | 50\$                               |      |
|    | 50        | F4 | A0        | 000000C3 | 8F    | C1    | ADDL3  | #195, -12(PREV_FMP), PREV_FMP      | 0940 |
|    |           | CC | A0        | CC       | A7    | 91    | CMPB   | -52(FMP), -52(PREV_FMP)            | 0942 |
|    |           |    |           | 07       | 12    | 00273 | BNEQ   | 49\$                               |      |
|    |           | CD | A0        | CD       | A7    | 91    | CMPB   | -51(FMP), -51(PREV_FMP)            | 0943 |
|    |           |    |           | 0B       | 13    | 0027A | BEQL   | 50\$                               |      |
|    |           | 7E | 00G       | 8F       | 9A    | 0027C | MOVZBL | #BASSK_SCAFACINT, -(SP)            | 0945 |
|    | 00000000G | 00 |           | 01       | FB    | 00280 | CALLS  | #1, BASS\$SIGNAL                   |      |
|    |           | 5E |           | 04       | C2    | 00287 | SUBL2  | #4, SP                             | 0951 |
|    |           | 6E |           | 56       | D0    | 0028A | MOVL   | RETURN_ADDRESS, (SP)               | 0952 |
|    |           |    |           | 05       | 0028D | RSB   |        |                                    | 0954 |

; Routine Size: 654 bytes, Routine Base: \_BASSCODE + 0000

: 578 0955 1  
: 579 0956 1 END  
: 580 0957 1

BAS\$INIT  
1-021

G 7  
16-Sep-1984 00:38:23  
14-Sep-1984 11:55:08

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

Page 16  
(3)

: 581 0958 0 ELUDOM

PSECT SUMMARY

| Name       | Bytes | Attributes                                              |
|------------|-------|---------------------------------------------------------|
| _BAS\$CODE | 654   | NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2) |

Library Statistics

| File                                | -----<br>Total | Symbols<br>Loaded | -----<br>Percent | Pages<br>Mapped | Processing<br>Time |
|-------------------------------------|----------------|-------------------|------------------|-----------------|--------------------|
| _\$255\$DUA28:[SYSLIB]STARLET.L32;1 | 9776           | 7                 | 0                | 581             | 00:01.1            |

COMMAND QUALIFIERS

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

: Size: 654 code + 0 data bytes  
: Run Time: 00:17.2  
: Elapsed Time: 00:37.1  
: Lines/CPU Min: 3345  
: Lexemes/CPU-Min: 20423  
: Memory Used: 235 pages  
: Compilation Complete

0024 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY