



```

BBBBBBBB      AAAAAA      SSSSSSSS      IIIIII      NN      NN      IIIIII      000000      NN      NN      EEEEEEEEE
BBBBBBBB      AAAAAA      SSSSSSSS      IIIIII      NN      NN      IIIIII      000000      NN      NN      EEEEEEEEE
BB      BB      AA      AA      SS      II      NN      NN      II      00      00      NN      NN      EE
BB      BB      AA      AA      SS      II      NN      NN      II      00      00      NN      NN      EE
BB      BB      AA      AA      SS      II      NNNN      NN      II      00      00      NNNN      NN      EE
BBBBBBBB      AA      AA      SSSSSS      II      NN      NN      NN      II      00      00      NN      NN      EEEEEEE
BBBBBBBB      AA      AA      SSSSSS      II      NN      NN      NN      II      00      00      NN      NN      EEEEEEE
BB      BB      AAAAAAAAAA      SS      II      NN      NN      NN      II      00      00      NN      NN      EE
BB      BB      AAAAAAAAAA      SS      II      NN      NN      NN      II      00      00      NN      NN      EE
BB      BB      AA      AA      SS      II      NN      NN      NN      II      00      00      NN      NN      EE
BB      BB      AA      AA      SS      II      NN      NN      NN      II      00      00      NN      NN      EE
BBBBBBBB      AA      AA      SSSSSSSS      IIIIII      NN      NN      IIIIII      000000      NN      NN      EEEEEEEEE
BBBBBBBB      AA      AA      SSSSSSSS      IIIIII      NN      NN      IIIIII      000000      NN      NN      EEEEEEEEE

```

```

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 BASSINIT_ONER (
2 0002 0     IDENT = '1-003'
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 ++
32 0032 1 FACILITY: BASIC-PLUS-2 Frame Support
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1     These routines set up and tear down frames for BASIC-PLUS-2.
37 0037 1     Frames are used for main routines, external functions,
38 0038 1     external subroutines, internal functions (both DEFs and DEF*s)
39 0039 1     internal subroutines (GOSUBs) and condition handlers.
40 0040 1
41 0041 1 ENVIRONMENT: VAX-11 user mode
42 0042 1
43 0043 1 AUTHOR: John Sauter, CREATION DATE: 10-Oct-78
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1     1-001 - Original.
48 0048 1     1-002 - Use BSF$ instead of BASS$ for stack frame prefix. JBS 08-FEB-1979
49 0049 1     1-003 - Set the IV bit in the PSW if requested. JBS 11-SEP-1979
50 0050 1 --
51 0051 1
52 0052 1 !<BLF/PAGE>
    
```

! File: BASINIONE.B32



```

112 0489 1 GLOBAL ROUTINE BASSINIT_ONERR (
113 0490 1     OLD_FMP,
114 0491 1     NEW_PC
115 0492 1 ) =
116 0493 1
117 0494 1 !++
118 0495 1 ! FUNCTIONAL DESCRIPTION:
119 0496 1
120 0497 1     Set up a frame for a BASIC-PLUS-2 condition handler.
121 0498 1     The frame is allocated on the stack, and R9 is left pointing
122 0499 1     to its temporary storage. R10 and R11 are set up from the
123 0500 1     frame which declared the error handler.
124 0501 1
125 0502 1 ! FORMAL PARAMETERS:
126 0503 1
127 0504 1     OLD_FMP.ra.v     Address of the frame of the establisher of
128 0505 1     the error handler.
129 0506 1     NEW_PC.ra.v      Address of the first line of the condition
130 0507 1     handler.
131 0508 1
132 0509 1 ! IMPLICIT INPUTS:
133 0510 1
134 0511 1     NONE
135 0512 1
136 0513 1 ! IMPLICIT OUTPUTS:
137 0514 1
138 0515 1     The value of R9, which points to the temporary storage,
139 0516 1     and of R10 and R11, which point to the variables of the
140 0517 1     establisher.
141 0518 1
142 0519 1 ! ROUTINE VALUE:
143 0520 1
144 0521 1     The "value" of the routine is determined by how the condition
145 0522 1     handler terminates. See the BASS$ERROR module for the
146 0523 1     termination routines and what value they cause to be returned.
147 0524 1
148 0525 1 ! COMPLETION CODES:
149 0526 1
150 0527 1     NONE
151 0528 1
152 0529 1 ! SIDE EFFECTS:
153 0530 1
154 0531 1     Leaves lots of things on the stack for use by the compiled
155 0532 1     BASIC-PLUS-2 code. This routine calls the compiled code, and
156 0533 1     return to this routine's caller will be made when the compiled
157 0534 1     code does a RESUME, ON ERROR GOTO 0 or ON ERROR GO BACK.
158 0535 1     See the BASS$ERROR module for details.
159 0536 1
160 0537 1 !--
161 0538 1
162 0539 2     BEGIN
163 0540 2
164 0541 2     MAP
165 0542 2         OLD_FMP : REF BLOCK [0, BYTE] FIELD (BSF$FCD);
166 0543 2
167 0544 2     BUILTIN
168 0545 2         FP,

```

```

: 169      0546      SP,
: 170      0547      BI$PSW;
: 171      0548
: 172      0549      +
: 173      0550      + Define local variables as registers. We cannot have any stack
: 174      0551      + locals since we manipulate the stack pointer in this routine.
: 175      0552      -
: 176      0553
: 177      0554      REGISTER
: 178      0555      FMP : REF BLOCK [0, BYTE] FIELD (BSF$FCD),      ! pointer to FCD
: 179      0556      ARGLIST : REF BLOCK [0, BYTE] FIELD (BASS$INIT_ARGS);      ! points to establisher's arg list
: 180      0557
: 181      0558      +
: 182      0559      + The following registers are passed to the compiled code.
: 183      0560      -
: 184      0561
: 185      0562      GLOBAL REGISTER
: 186      0563      BSF$A_MAJOR_STG = 11,
: 187      0564      BSF$A_MINOR_STG = 10,
: 188      0565      BSF$A_TEMP_STG = 9;
: 189      0566
: 190      0567      +
: 191      0568      + Allocate frame control data.
: 192      0569      -
: 193      0570      FMP = .FMP;
: 194      0571      SP = .FMP - BSF$K_LENFCDONE;
: 195      0572      +
: 196      0573      + Set up new temporary storage.
: 197      0574      -
: 198      0575      ARGLIST = .OLD_FMP [BSF$A_INIT_ARG];
: 199      0576
: 200      0577      IF ((.ARGLIST [BAS$L_IN_NO_TST] NEQ 0) OR (.ARGLIST [BAS$L_IN_NO_NMT] NEQ 0))
: 201      0578      THEN
: 202      0579      BEGIN
: 203      0580      +
: 204      0581      + We must set up R9. First allocate string temporaries.
: 205      0582      -
: 206      0583
: 207      0584      INCR COUNTER FROM 1 TO .ARGLIST [BAS$L_IN_NO_TST] DO
: 208      0585      BEGIN
: 209      0586      SP = .SP - %UPVAL;
: 210      0587      .SP = 0;      ! Pointer 0 implies not allocated.
: 211      0588      SP = .SP - %UPVAL;
: 212      0589      BLOCK [.SP, DSC$B_CLASS; 0, BYTE] = DSC$K_CLASS_D;
: 213      0590      BLOCK [.SP, DSC$B_DTYPE; 0, BYTE] = DSC$K_DTYPE_T;
: 214      0591      BLOCK [.SP, DSC$W_LENGTH; 0, BYTE] = 0;
: 215      0592      END;
: 216      0593
: 217      0594      +
: 218      0595      + Point R9 to the last string descriptor allocated.
: 219      0596      -
: 220      0597      BSF$A_TEMP_STG = .SP;
: 221      0598      +
: 222      0599      + Now allocate numeric temporaries.
: 223      0600      -
: 224      0601      SP = .SP - .ARGLIST [BAS$L_IN_NO_NMT];
: 225      0602      END;

```

```

226 0603
227 0604
228 0605
229 0606
230 0607
231 0608
232 0609
233 0610
234 0611
235 0612
236 0613
237 0614
238 0615
239 0616
240 0617
241 0618
242 0619
243 0620
244 0621
245 0622
246 0623
247 0624
248 0625
249 0626
250 0627
251 0628
252 0629
253 0630
254 0631
255 0632
256 0633
257 0634
258 0635
259 0636
260 0637
261 0638
262 0639
263 0640
264 0641
265 0642
266 0643
267 0644
268 0645
269 0646
270 0647
271 0648
272 0649

+ Initialize the parts of the FCD relavent to a condition handler.
-
FMP [BSFSA_MARK] = 0;
FMP [BSFSA_BASE_SP] = .SP;
FMP [BSFSA_BASE_R11] = (BSFSA_MAJOR_STG = .OLD_FMP [BSFSA_BASE_R11]);
FMP [BSFSA_BASE_R10] = (BSFSA_MINOR_STG = .OLD_FMP [BSFSA_BASE_R10]);
FMP [BSFSA_BASE_R9] = .BSFSA_TEMP_STG;
+
- The "PROCEDURE ID" is the address of the start of the condition handler.
-
FMP [BSFSA_PROC_ID] = .NEW_PC;
+
- Copy the frame flags from the old frame.
-
FMP [BSF$W_FCD_FLAGS] = .OLD_FMP [BSF$W_FCD_FLAGS];
+
- Set the frame ID to be "CONDITION HANDLER". This frame ID is checked
for by the RESUME, ON ERROR GOTO 0 and ON ERROR GO BACK routines.
-
FMP [BSF$B_PROC_CODE] = BSF$K_PROC_ONER;
+
- Set the frame length field.
-
FMP [BSF$B_LEN_FCD] = BSF$K_LENFCDONE;
+
- Set the integer interrupt enable bit in the PSW if requested.
-
IF ((.FMP [BSF$W_FCD_FLAGS] AND BSF$M_FCD_IV) NEQ 0) THEN BISPSW (%REF (PSW$M_IV));
+
- Set up the exception handler. This also marks the frame as a
BASIC frame.
-
FMP [BSFSA_HANDLER] = BASSHANDLER;
+
- Branch to the compiled code.
-
BASSCOND_JSB (.NEW_PC);
+
- The routine we "call" above will cut back the stack, and so never
return here, but we must return a value to satisfy BLISS.
-
RETURN (0);
END;

! of BASSINIT_ONER

.TITLE BASSINIT_ONER
.IDENT \1-003\

.EXTRN BASSHANDLER

.PSECT _BASSCODE,NOWRT, SHR, PIC,2

.ENTRY BASSINIT_ONERR, Save R2,R3,R4,R5,R6,R7,R8,- ; 0489

OFFC 0000

```

	50		5D	D0	00002	MOVL	R9,R10,R11	0570
	5E	E0	A0	9E	00005	MOVAB	FP,FMP	0571
	52	04	AC	D0	00009	MOVL	-32(R0),SP	0575
	51	D8	A2	D0	0000D	MOVL	OLD FMP,R2	
		30	A1	D5	00011	MOVL	-40(R2),ARGLIST	
			05	12	00014	TSTL	48(ARGLIST)	0577
		34	A1	D5	00016	BNEQ	1\$	
			1F	13	00019	TSTL	52(ARGLIST)	
			53	D4	0001B	BEQL	4\$	
			0F	11	0001D	CLRL	COUNTER	0584
	5E		04	C2	0001F	BRB	3\$	
			6E	D4	00022	SUBL2	#4,SP	0586
			04	C2	00024	CLRL	(SP)	0587
	5E		04	C2	00024	SUBL2	#4,SP	0588
	6E	020E0000	8F	D0	00027	MOVL	#34471936,(SP)	0591
EC	53		A1	F3	0002E	AOBLEQ	48(ARGLIST),COUNTER,2\$	0584
	59		5E	D0	00033	MOVL	SP,BSF\$A TEMP STG	0597
	5E		A1	C2	00036	SUBL2	52(ARGLIST),SP	0601
			FC	A0	0003A	CLRL	-4(FMP)	0607
	F8	A0	5E	D0	0003D	MOVL	SP,-8(FMP)	0608
			A2	7D	00041	MOVQ	-16(R2),BSF\$A MINOR STG	0610
	F0	A0	5A	7D	00045	MOVQ	BSF\$A MINOR STG,-16(FMP)	
	EC	A0	59	D0	00049	MOVL	BSF\$A TEMP STG,-20(FMP)	0611
	E8	A0	08	AC	0004D	MOVL	NEW PC,-24(FMP)	0615
	E6	A0	E6	A2	00052	MOVW	-26(R2),-26(FMP)	0619
	E4	A0	0720	8F	00057	MOVW	#1824,-28(FMP)	0628
02	E6	A0	0B	E1	0005D	BBC	#11,-26(FMP),5\$	0633
			20	B8	00062	BISPSW	#32	
	60	00000000G	00	9E	00064	MOVAB	BAS\$HANDLER,(FMP)	0639
			08	BC	0006B	JSB	@NEW_PC	0643
			50	D4	0006E	CLRL	R0	0648
			04	00	00070	RET		0649

: Routine Size: 113 bytes, Routine Base: \_BAS\$CODE + 0000

```

: 273      0650  1
: 274      0651  1 END
: 275      0652  1
: 276      0653  0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
_BAS\$CODE	113	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

BASS\$INIT\_ONER  
1-003

C 6  
16-Sep-1984 00:37:57  
14-Sep-1984 11:55:08

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

Page 7  
(3)

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
:_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	6	0	581	00:01.1

COMMAND QUALIFIERS

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

: Size: 113 code + 0 data bytes  
: Run Time: 00:07.2  
: Elapsed Time: 00:17.4  
: Lines/CPU Min: 5449  
: Lexemes/CPU-Min: 21137  
: Memory Used: 85 pages  
: Compilation Complete

