

BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAA		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      SSSSSSSSS  FFFFFFFF  000000  RRRRRRRR  MM      MM      AAAAAA  TTTTTTTTTT
BBBBBBBBB      AAAAAA      SSSSSSSSS  FFFFFFFF  000000  RRRRRRRR  MM      MM      AAAAAA  TTTTTTTTTT
BB      BB      AA      AA      SS      FF      00      00      RR      RR      MMMM  MMMM  AA      AA      TT
BB      BB      AA      AA      SS      FF      00      00      RR      RR      MMMM  MMMM  AA      AA      TT
BB      BB      AA      AA      SS      FF      00      00      RR      RR      MM  MM  MM  AA      AA      TT
BBBBBBBBB      AA      AA      SSSSSS  FFFFFFFF  00      00      RRRRRRRR  MM      MM  AA      AA      TT
BBBBBBBBB      AA      AA      SSSSSS  FFFFFFFF  00      00      RRRRRRRR  MM      MM  AA      AA      TT
BB      BB      AAAAAAAAAA      SS      FF      00      00      RR      RR      MM      MM  AAAAAAAAAA  TT
BB      BB      AAAAAAAAAA      SS      FF      00      00      RR      RR      MM      MM  AAAAAAAAAA  TT
BB      BB      AA      AA      SS      FF      00      00      RR      RR      MM      MM  AA      AA      TT
BB      BB      AA      AA      SS      FF      00      00      RR      RR      MM      MM  AA      AA      TT
BBBBBBBBB      AA      AA      SSSSSSSS  FF      000000  RR      RR      MM      MM  AA      AA      TT
BBBBBBBBB      AA      AA      SSSSSSSS  FF      000000  RR      RR      MM      MM  AA      AA      TT
                                     ....
                                     ....
                                     ....
                                     ....

LL      I I I I I      SSSSSSSSS
LL      I I I I I      SSSSSSSSS
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  I I I I I      SSSSSSSS
LLLLLLLLLL  I I I I I      SSSSSSSS
```

```

1 0001 0 MODULE BASS$FORMAT (
2 0002 0 IDENT = '1-006' ! Format a number by BASIC FORMATS function
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 support library
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 Create a string by formatting a number according to the semantics of
37 0037 1 the PRINT USING statement.
38 0038 1
39 0039 1 ENVIRONMENT: User mode, AST level or not or mixed
40 0040 1
41 0041 1 AUTHOR: R. Will, CREATION DATE: 15-May-79
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 R. Will, 15-May-79: VERSION 01
46 0046 1 01 - original
47 0047 1 1-002 - Always call format interpreter with destination string. RW 18-Jul-79
48 0048 1 1-003 - String cleanup. Don't use $STR$ macros. RW 29-Oct-79
49 0049 1 1-004 - For BASIC V2.0, enable FORMATS to be able to take any type of
50 0050 1 elements, not just numeric. For this enhancement we need a new
51 0051 1 routine : BASS$FORMAT_I. FM 11-JUL-81
52 0052 1 1-005 - Add entry points for g and h floating. PL 3-Sep-81
53 0053 1 1-006 - Add entry point for packed decimal. PLL 19-Jan-82
54 0054 1 --
55 0055 1 !<BLF/PAGE>

```

```
57 0056 1 |
58 0057 1 | SWITCHES:
59 0058 1 |
60 0059 1 |
61 0060 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
62 0061 1 |
63 0062 1 |
64 0063 1 | LINKAGES:
65 0064 1 |
66 0065 1 |
67 0066 1 | REQUIRE 'RTLIN:BASLNK';           ! Get linkage for scaling
68 0143 1 |
69 0144 1 |
70 0145 1 | TABLE OF CONTENTS:
71 0146 1 |
72 0147 1 |
73 0148 1 | FORWARD ROUTINE
74 0149 1 |     BASSFORMAT_F : NOVALUE,       ! Format a floating number using PRINT
75 0150 1 |                                     ! USING semantics
76 0151 1 |     BASSFORMAT_D : NOVALUE,       ! Format a double floating number
77 0152 1 |                                     ! using PRINT USING semantics
78 0153 1 |     BASSFORMAT_T : NOVALUE,       ! Format a string using PRINT
79 0154 1 |                                     ! USING semantics
80 0155 1 |     BASSFORMAT_G : NOVALUE,       ! Format a g float number
81 0156 1 |                                     ! using PRINT USING semantics
82 0157 1 |     BASSFORMAT_H : NOVALUE,       ! Format an h float number
83 0158 1 |                                     ! using PRINT USING semantics
84 0159 1 |     BASSFORMAT_P : NOVALUE;       ! Format a decimal number
85 0160 1 |                                     ! using PRINT USING semantics
86 0161 1 |
87 0162 1 |
88 0163 1 | INCLUDE FILES:
89 0164 1 |
90 0165 1 |
91 0166 1 | REQUIRE 'RTLIN:RTLPSECT';         ! Declare PSECTS code
92 0261 1 | REQUIRE 'RTLIN:BASFRAME';         ! Define offsets in BASIC frame
93 0464 1 | LIBRARY 'RTLSTARLE';             ! STARLET library for macros and symbol
94 0465 1 |
95 0466 1 |
96 0467 1 | MACROS: NONE
97 0468 1 |
98 0469 1 |
99 0470 1 |
100 0471 1 | EQUATED SYMBOLS: NONE
101 0472 1 |
102 0473 1 |
103 0474 1 |
104 0475 1 | PSECT DECLARATIONS
105 0476 1 |
106 0477 1 |
107 0478 1 | DECLARE_PSECTS (BAS);
108 0479 1 |
109 0480 1 |
110 0481 1 | OWN STORAGE: NONE
111 0482 1 |
112 0483 1 |
113 0484 1 |
```

BASS\$FORMAT
1-006

N 11
16-Sep-1984 00:32:29
14-Sep-1984 11:54:59

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

Page 3
(2)

: 114
: 115
: 116
: 117
: 118

0485 1 ! EXTERNAL REFERENCES:
0486 1 !
0487 1 !
0488 1 EXTERNAL ROUTINE
0489 1 BASS\$FORMAT_INT : NOVALUE;

! BASIC format interpreter

```
120 0490 1 GLOBAL ROUTINE BASS$FORMAT_F (
121 0491 1     DEST_DESC,
122 0492 1     VALUE,
123 0493 1     FORMAT_DESC)
124 0494 1 : NOVALUE =
125 0495 1
126 0496 1 ++
127 0497 1 FUNCTIONAL DESCRIPTION:
128 0498 1     Format a floating number according to the format in the format
129 0499 1     string and the rules of PRINT USING. To be sure that PRINT USING
130 0500 1     and FORMATS are compatible, all formatting will be done by one
131 0501 1     routine, BASS$FORMAT_INT.
132 0502 1
133 0503 1 FORMAL PARAMETERS:
134 0504 1
135 0505 1     DEST_DESC.wt.dx      pointer to descriptor of output string
136 0506 1     VALUE.rf.v          value of floating number to be formatted
137 0507 1     FORMAT_DESC.rt.dx pointer to descriptor of format string
138 0508 1
139 0509 1 IMPLICIT INPUTS:
140 0510 1
141 0511 1     NONE
142 0512 1
143 0513 1 IMPLICIT OUTPUTS:
144 0514 1
145 0515 1     NONE
146 0516 1
147 0517 1 ROUTINE VALUE:
148 0518 1 COMPLETION CODES:
149 0519 1
150 0520 1     NONE
151 0521 1
152 0522 1 SIDE EFFECTS:
153 0523 1
154 0524 1     This routine may allocate a temporary dynamic string and may cause
155 0525 1     any of STR$GET1's errors to be signalled. This routine also may
156 0526 1     call STR$COPY which may signal errors. This routine also calls
157 0527 1     the BASIC format interpreter which may allocate space to the
158 0528 1     destination string.
159 0529 1
160 0530 1 --
161 0531 1
162 0532 2 BEGIN
163 0533 2
164 0534 2 LOCAL
165 0535 2     RET_TEMP;
166 0536 2     ! Not used by this routine, but required by
167 0537 2     ! format interpreter
168 0538 2
169 0539 2 MAP
170 0540 2     FORMAT_DESC: REF BLOCK [8, BYTE];
171 0541 2
172 0542 2 MAP
173 0543 2     DEST_DESC: REF BLOCK [8, BYTE];
174 0544 2
175 0545 2 BASS$FORMAT_INT (
176 0546 2     VALUE,
177 0546 2     FORMAT_DESC [0,0,0,0],
178 0546 2     ! call formatter
179 0546 2     ! addr of value
180 0546 2     ! descr of format str
```

BASS\$FORMAT
1-006

C 12
16-Sep-1984 00:32:29
14-Sep-1984 11:54:59

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BAS\$FORMAT.B32;1

Page 5
(3)

```
: 177      0547  2      DSC$K_DTYPE_F
: 178      0548  2      DEST_DESC [0,0,0,0],
: 179      0549  2      RET_TEMP    );
: 180      0550  2
: 181      0551  2      RETURN;
: 182      0552  2
: 183      0553  1      END;
```

```
: floating value
: return string
: dummy arg
```

!End of BASS\$FORMAT_F

```
.TITLE BASS$FORMAT
.IDENT \1-006\
```

```
.EXTRN BASS$$FORMAT_INT
```

```
.PSECT _BASS$CODE,NOWRT, SHR, PIC,2
```

```
.ENTRY BASS$FORMAT_F, Save nothing
```

```
SUBL2 #4, SP
```

```
PUSHL SP
```

```
PUSHL DEST_DESC
```

```
PUSHL #10
```

```
PUSHL FORMAT_DESC
```

```
PUSHAB VALUE
```

```
CALLS #5, BASS$$FORMAT_INT
```

```
RET
```

: 0490

: 0548

: 0544

: 0548

: 0553

```
SE          0000 00000
          04 C2 00002
          5E DD 00005
          04 AC DD 00007
          0A DD 0000A
          0C AC DD 0000C
          08 AC 9F 0000F
00000000G 00 05 FB 00012
          04 00019
```

; Routine Size: 26 bytes, Routine Base: _BASS\$CODE + 0000

```
185 0554 1 GLOBAL ROUTINE BASS$FORMAT_D (
186 0555 1
187 0556 1     DEST_DESC,
188 0557 1     VALUE1,
189 0558 1     VALUE2,
190 0559 1     FORMAT_DESC)
191 0560 1 : NOVALUE =
192 0561 1
193 0562 1 ++
194 0563 1 FUNCTIONAL DESCRIPTION:
195 0564 1     Format a double precision number according to the format in the format
196 0565 1     string and the rules of PRINT USING. To be sure that PRINT USING
197 0566 1     and FORMATS$ are compatible, all formatting will be done by one
198 0567 1     routine, BASS$FORMAT_INT.
199 0568 1
200 0569 1 FORMAL PARAMETERS:
201 0570 1     DEST_DESC.wt.dx     pointer to descriptor of output string
202 0571 1     VALUE.rd.v        value of double number to be formatted
203 0572 1     FORMAT_DESC.rt.dx pointer to descriptor of format string
204 0573 1
205 0574 1 IMPLICIT INPUTS:
206 0575 1     Scale for double in the callers BASIC frame
207 0576 1
208 0577 1 IMPLICIT OUTPUTS:
209 0578 1     NONE
210 0579 1
211 0580 1 ROUTINE VALUE:
212 0581 1 COMPLETION CODES:
213 0582 1     NONE
214 0583 1
215 0584 1 SIDE EFFECTS:
216 0585 1     This routine calls the BASIC format interpreter which may allocate
217 0586 1     space to the destination string.
218 0587 1
219 0588 1 --
220 0589 1
221 0590 1 BEGIN
222 0591 1
223 0592 1 LOCAL
224 0593 1     RET_TEMP;
225 0594 2
226 0595 2
227 0596 2
228 0597 2     RET_TEMP;
229 0598 2
230 0599 2     MAP
231 0600 2     FORMAT_DESC: REF BLOCK [8, BYTE],
232 0601 2     DEST_DESC: REF BLOCK [8, BYTE];
233 0602 2
234 0603 2 BASS$FORMAT_INT (
235 0604 2     VALUE1,
236 0605 2     FORMAT_DESC [0,0,0,0],
237 0606 2     DSC$K_DTYPE_D,
238 0607 2     DEST_DESC [0,0,0,0],
239 0608 2     RET_TEMP,
240 0609 2     SBASS$SCALE );
241 0610 2
```

Format double like print using
Pointer to dest string desc
1st longword of double value
2nd longword of double value
Pointer to format str desc

call formatter
addr of value
descr of format str
floating value
return string
dummy arg
scale

BASS\$FORMAT
1-006

E 12
16-Sep-1984 00:32:29
14-Sep-1984 11:54:59

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BAS\$FORMAT.B32;1

Page 7
(4)

: 242
: 243
: 244
0611 2 RETURN;
0612 2
0613 1 END;

!End of BASS\$FORMAT_D

OFFC 00000
5E 04 C2 00002
51 5D D0 00005
50 0C A1 D0 00008
00000000G 00 00 16 0000C
04 50 DD 00012
04 04 AE 9F 00014
04 AC DD 00017
10 0B DD 0001A
08 AC DD 0001C
00000000G 00 08 AC 9F 0001F
06 FB 00022
04 00029

.EXTRN BASS\$\$SCALE_L_R1
.ENTRY BASS\$FORMAT_D, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0554
R10,R11
SUBL2 #4, SP
MOVL FP, FMP 0608
MOVL 12(FMP), R0
JSB BASS\$\$SCALE_L_R1
PUSHL R0
PUSHAB RET_TEMP 0607
PUSHL DEST_DESC
PUSHL #11
PUSHL FORMAT_DESC
PUSHAB VALUE1 0603
CALLS #6, BASS\$FORMAT_INT 0607
RET 0613

; Routine Size: 42 bytes, Routine Base: _BAS\$CODE + 001A

```
: 246      0614 1 GLOBAL ROUTINE BASS$FORMAT_T (      | Format a string with print using
: 247      0615 1                                | Pointer to dest string desc
: 248      0616 1                                | Pointer to string element's descriptor.
: 249      0617 1                                | Pointer to format str desc
: 250      0618 1                                | NOVALOE =
: 251      0619 1 ++
: 252      0620 1 FUNCTIONAL DESCRIPTION:
: 253      0621 1
: 254      0622 1     Format a string according to the format in the format
: 255      0623 1     string and the rules of PRINT USING. To be sure that PRINT USING
: 256      0624 1     and FORMAT$ are compatible, all formatting will be done by one
: 257      0625 1     routine, BASS$FORMAT_INT.
: 258      0626 1
: 259      0627 1 FORMAL PARAMETERS:
: 260      0628 1
: 261      0629 1     DEST_DESC.wt.dx      pointer to descriptor of output string
: 262      0630 1     ELEM_DESC.rt.dx      pointer to descriptor of element passed
: 263      0631 1     FORMAT_DESC.rt.dx    pointer to descriptor of format string
: 264      0632 1
: 265      0633 1 IMPLICIT INPUTS:
: 266      0634 1
: 267      0635 1     NONE
: 268      0636 1
: 269      0637 1 IMPLICIT OUTPUTS:
: 270      0638 1
: 271      0639 1     NONE
: 272      0640 1
: 273      0641 1 ROUTINE VALUE:
: 274      0642 1 COMPLETION CODES:
: 275      0643 1
: 276      0644 1     NONE
: 277      0645 1
: 278      0646 1 SIDE EFFECTS:
: 279      0647 1
: 280      0648 1     This routine may allocate a temporary dynamic string and may cause
: 281      0649 1     any of STR$GET1's errors to be signalled. This routine also may
: 282      0650 1     call STR$COPY which may signal errors. This routine also calls
: 283      0651 1     the BASIC format interpreter which may allocate space to the
: 284      0652 1     destination string.
: 285      0653 1
: 286      0654 1 --
: 287      0655 1
: 288      0656 2 BEGIN
: 289      0657 2
: 290      0658 2 LOCAL
: 291      0659 2     RET_TEMP;      ! Not used by this routine, but required by
: 292      0660 2                        ! format interpreter
: 293      0661 2
: 294      0662 2 MAP
: 295      0663 2     FORMAT_DESC: REF BLOCK [8, BYTE];
: 296      0664 2
: 297      0665 2 MAP
: 298      0666 2     DEST_DESC: REF BLOCK [8, BYTE];
: 299      0667 2
: 300      0668 2 MAP
: 301      0669 2     ELEM_DESC: REF BLOCK [8, BYTE];
: 302      0670 2
```

BASS\$FORMAT
1-006

G 12
16-Sep-1984 00:32:29
14-Sep-1984 11:54:59

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BAS\$FORMAT.B32;1

Page 9
(5)

```
: 303      0671  2  BASS$FORMAT INT (
: 304      0672  2      ELEM_DESC [0,0,0,0],
: 305      0673  2      FORMAT_DESC [0,0,0,0],
: 306      0674  2      DSC$K_DTYPE T,
: 307      0675  2      DEST_DESC [0,0,0,0],
: 308      0676  2      RET_TEMP
: 309      0677  2      );
: 310      0678  2  RETURN;
: 311      0679  2
: 312      0680  1  END;
```

```
: call formatter
: pointer to element's desc.
: descr of format str
: string
: return string
: dummy arg
```

!End of BASS\$FORMAT_F

```

          5E          0000 00000
          04 C2 00002
          5E DD 00005
          04 AC DD 00007
          0E DD 0000A
          08 AC 7D 0000C
00000000G 7E 00 05 FB 00010
          04 00017
```

```
.ENTRY BASS$FORMAT_T, Save nothing
SUBL2 #4, SP
PUSHL SP
PUSHL DEST_DESC
PUSHL #14
MOVQ ELEM_DESC, -(SP)
CALLS #5, BASS$FORMAT_INT
RET
```

```
: 0614
:
: 0675
:
:
:
: 0680
```

; Routine Size: 24 bytes, Routine Base: _BASS\$CODE + 0044

```
314 0681 1 GLOBAL ROUTINE BASSFORMAT_G (      ! Format g float like print using
315 0682 1                                     ! Pointer to dest string desc
316 0683 1                                     ! 1st longword of g float value
317 0684 1                                     ! 2nd longword of g float value
318 0685 1                                     ! Pointer to format str desc
319 0686 1                                     : NOVALUE =
320 0687 1
321 0688 1 ++
322 0689 1 FUNCTIONAL DESCRIPTION:
323 0690 1     Format a g floating number according to the format in the format
324 0691 1     string and the rules of PRINT USING. To be sure that PRINT USING
325 0692 1     and FORMATS are compatible, all formatting will be done by one
326 0693 1     routine, BASS$FORMAT_INT.
327 0694 1
328 0695 1 FORMAL PARAMETERS:
329 0696 1
330 0697 1     DEST_DESC.wt.dx      pointer to descriptor of output string
331 0698 1     VALUE.rg.v         value of g float number to be formatted
332 0699 1     FORMAT_DESC.rt.dx  pointer to descriptor of format string
333 0700 1
334 0701 1 IMPLICIT INPUTS:
335 0702 1
336 0703 1     NONE
337 0704 1
338 0705 1 IMPLICIT OUTPUTS:
339 0706 1
340 0707 1     NONE
341 0708 1
342 0709 1 ROUTINE VALUE:
343 0710 1 COMPLETION CODES:
344 0711 1
345 0712 1     NONE
346 0713 1
347 0714 1 SIDE EFFECTS:
348 0715 1
349 0716 1     This routine calls the BASIC format interpreter which may allocate
350 0717 1     space to the destination string.
351 0718 1
352 0719 1 --
353 0720 1
354 0721 2 BEGIN
355 0722 2
356 0723 2 LOCAL
357 0724 2     RET_TEMP;      ! Not used by this routine, but required by
358 0725 2                   ! format interpreter
359 0726 2
360 0727 2 MAP
361 0728 2     FORMAT_DESC: REF BLOCK [8, BYTE],
362 0729 2     DEST_DESC: REF BLOCK [8, BYTE];
363 0730 2
364 0731 2 BASS$FORMAT_INT (      ! call formatter
365 0732 2     VALUE1,           ! addr of value
366 0733 2     FORMAT_DESC [0,0,0,0], ! descr of format str
367 0734 2     DSC$K_DTYPE_G,      ! g floating value
368 0735 2     DEST_DESC [0,0,0,0], ! return string
369 0736 2     RET_TEMP,           ! dummy arg
370 0737 2     0);               ! scale
```

BASS\$FORMAT
1-006

I 12
16-Sep-1984 00:32:29
14-Sep-1984 11:54:59

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BAS\$FORMAT.B32;1

Page 11
(6)

: 371 0738 2 RETURN;
: 372 0739 2
: 373 0740 1 END;

!End of BASS\$FORMAT_G

5E 0000 00000
04 04 C2 00002
7E D4 00005
04 AE 9F 00007
04 AC DD 0000A
1B DD 0000D
10 AC DD 0000F
08 AC 9F 00012
06 FB 00015
04 0001C

00000000G 00

.ENTRY BASS\$FORMAT_G, Save nothing
SUBL2 #4, SP
CLRL -(SP)
PUSHAB RET_TEMP
PUSHL DEST_DESC
PUSHL #27
PUSHL FORMAT_DESC
PUSHAB VALUE1
CALLS #6, BASS\$FORMAT_INT
RET

: 0681
: 0734
: 0730
: 0734
: 0740

; Routine Size: 29 bytes, Routine Base: _BASS\$CODE + 005C

```
375 0741 1 GLOBAL ROUTINE BASS$FORMAT_H (      ! Format h float like print using
376 0742 1                                     ! Pointer to dest string desc
377 0743 1                                     ! 1st longword of h float value
378 0744 1                                     ! 2nd longword of h float value
379 0745 1                                     ! 3rd longword of h float value
380 0746 1                                     ! 4th longword of h float value
381 0747 1                                     ! Pointer to format str desc
382 0748 1                                     : NOVALUE =
383 0749 1
384 0750 1 ++
385 0751 1 FUNCTIONAL DESCRIPTION:
386 0752 1     Format an h floating number according to the format in the format
387 0753 1     string and the rules of PRINT USING. To be sure that PRINT USING
388 0754 1     and FORMATS are compatible, all formatting will be done by one
389 0755 1     routine, BASS$FORMAT_INT.
390 0756 1
391 0757 1 FORMAL PARAMETERS:
392 0758 1
393 0759 1     DEST_DESC.wt.dx      pointer to descriptor of output string
394 0760 1     VALUE.rh.v       value of h float number to be formatted
395 0761 1     FORMAT_DESC.rt.dx pointer to descriptor of format string
396 0762 1
397 0763 1 IMPLICIT INPUTS:
398 0764 1
399 0765 1     NONE
400 0766 1
401 0767 1 IMPLICIT OUTPUTS:
402 0768 1
403 0769 1     NONE
404 0770 1
405 0771 1 ROUTINE VALUE:
406 0772 1 COMPLETION CODES:
407 0773 1
408 0774 1     NONE
409 0775 1
410 0776 1 SIDE EFFECTS:
411 0777 1
412 0778 1     This routine calls the BASIC format interpreter which may allocate
413 0779 1     space to the destination string.
414 0780 1
415 0781 1 --
416 0782 1
417 0783 2 BEGIN
418 0784 2
419 0785 2 LOCAL
420 0786 2     RET_TEMP;      ! Not used by this routine, but required by
421 0787 2                     ! format interpreter
422 0788 2
423 0789 2 MAP
424 0790 2     FORMAT_DESC: REF BLOCK [8, BYTE],
425 0791 2     DEST_DESC: REF BLOCK [8, BYTE];
426 0792 2
427 0793 2 BASS$FORMAT_INT (      ! call formatter
428 0794 2     VALUE1,             ! addr of value
429 0795 2     FORMAT_DESC [0,0,0,0], ! descr of format str
430 0796 2     DSC$K_BTYPE_H,       ! h floating value
431 0797 2     DEST_DESC [0,0,0,0], ! return string
                                ! dummy arg
```

BASS\$FORMAT
1-006

K 12
16-Sep-1984 00:32:29
14-Sep-1984 11:54:59

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BAS\$FORMAT.B32;1

Page 13
(7)

```
: 432      0798 2      0);
: 433      0799 2
: 434      0800 2      RETURN;
: 435      0801 2
: 436      0802 1      END;
```

! scale

!End of BASS\$FORMAT_H

```
                    5E          0000 00000
                                04 C2 00002
                                7E D4 00005
                                04 AE 9F 00007
                                04 AC DD 0000A
                                1C DD 0000D
                                18 AC DD 0000F
                                08 AC 9F 00012
                                06 FB 00015
                                04 0001C
000000000G 00
```

```
.ENTRY BASS$FORMAT_H, Save nothing
SUBL2 #4, SP
CLRL -(SP)
PUSHAB RET_TEMP
PUSHL DEST_DESC
PUSHL #28
PUSHL FORMAT_DESC
PUSHAB VALUE1
CALLS #6, BASS$FORMAT_INT
RET
```

```
: 0741
: 0796
:
:
: 0792
: 0796
: 0802
```

; Routine Size: 29 bytes, Routine Base: _BASS\$CODE + 0079

```

438 0803 1 GLOBAL ROUTINE BASS$FORMAT_P (
439 0804 1     DEST_DESC,
440 0805 1     VALUE,
441 0806 1     FORMAT_DESC)
442 0807 1     : NOVALUE =
443 0808 1
444 0809 1 ++
445 0810 1 FUNCTIONAL DESCRIPTION:
446 0811 1     Format a decimal number according to the format in the format
447 0812 1     string and the rules of PRINT USING. To be sure that PRINT USING
448 0813 1     and FORMATS are compatible, all formatting will be done by one
449 0814 1     routine, BASS$FORMAT_INT.
450 0815 1
451 0816 1 FORMAL PARAMETERS:
452 0817 1
453 0818 1     DEST_DESC.wt.dx      pointer to descriptor of output string
454 0819 1     VALUE.rp.dsd       pointer to descriptor of decimal number
455 0820 1     FORMAT_DESC.rt.dx pointer to descriptor of format string
456 0821 1
457 0822 1 IMPLICIT INPUTS:
458 0823 1
459 0824 1     NONE
460 0825 1
461 0826 1 IMPLICIT OUTPUTS:
462 0827 1
463 0828 1     NONE
464 0829 1
465 0830 1 ROUTINE VALUE:
466 0831 1 COMPLETION CODES:
467 0832 1
468 0833 1     NONE
469 0834 1
470 0835 1 SIDE EFFECTS:
471 0836 1
472 0837 1     This routine may allocate a temporary dynamic string and may cause
473 0838 1     any of STR$GET1's errors to be signalled. This routine also may
474 0839 1     call STR$COPY which may signal errors. This routine also calls
475 0840 1     the BASIC format interpreter which may allocate space to the
476 0841 1     destination string.
477 0842 1
478 0843 1 --
479 0844 1
480 0845 2 BEGIN
481 0846 2
482 0847 2 LOCAL
483 0848 2     RET_TEMP;
484 0849 2     ! Not used by this routine, but required by
485 0850 2     ! format interpreter
486 0851 2
487 0852 2 MAP
488 0853 2     FORMAT_DESC: REF BLOCK [8, BYTE],
489 0854 2     VALUE : REF BLOCK [12, BYTE];
490 0855 2
491 0856 2 MAP
492 0857 2     DEST_DESC: REF BLOCK [8, BYTE];
493 0858 2
494 0859 2 BASS$FORMAT_INT (
495 0859 2     .VALUE,
496 0859 2     ! call formatter
497 0859 2     ! addr of value

```

BASS\$FORMAT
1-006

M 12
16-Sep-1984 00:32:29 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:54:59 [BASRTL.SRC]BASS\$FORMAT.B32;1

Page 15
(8)

```
: 495      0860  2      FORMAT_DESC [0,0,0,0],      ! descr of format str
: 496      0861  2      DSC$K_DTYPE_P              ! decimal value
: 497      0862  2      DEST_DESC [0,0,0,0],          ! return string
: 498      0863  2      RET_TEMP                      ! dummy arg
: 499      0864  2
: 500      0865  2      RETURN;
: 501      0866  2
: 502      0867  1      END;                          !End of BASS$FORMAT_P
```

```
                                0000 00000
                                04 C2 00002
                                5E DD 00005
                                04 AC DD 00007
                                15 DD 0000A
                                08 AC 7D 0000C
                                05 FB 00010
                                04 00017
                                .ENTRY BASS$FORMAT_P, Save nothing
                                SUBL2 #4, SP
                                PUSH  SP
                                PUSH  DEST_DESC
                                PUSH  #21
                                MOVQ  VALUE, -(SP)
                                CALLS #5, BASS$FORMAT_INT
                                RET
```

; Routine Size: 24 bytes, Routine Base: _BAS\$CODE + 0096

BAS\$FORMAT
1-006

N 12
16-Sep-1984 00:32:29
14-Sep-1984 11:54:59

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

Page 16
(9)

: 504 0868 1 END
: 505 0869 0 ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
_BAS\$CODE	174	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	6	0	581	00:01.0

COMMAND QUALIFIERS

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

: Size: 174 code + 0 data bytes
: Run Time: 00:08.0
: Elapsed Time: 00:19.4
: Lines/CPU Min: 6493
: Lexemes/CPU-Min: 18074
: Memory Used: 53 pages
: Compilation Complete

0023 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

