


```

BBBBBBBB      AAAAAA      SSSSSSSS      SSSSSSSS      TTTTTTTTTT      RRRRRRRR      IIIIII      NN      NN      GGGGGGGG
BBBBBBBB      AAAAAA      SSSSSSSS      SSSSSSSS      TTTTTTTTTT      RRRRRRRR      IIIIII      NN      NN      GGGGGGGG
BB      BB      AA      AA      SS      SS      SS      SS      TT      TT      RR      RR      II      NN      NN      GG      GG
BB      BB      AA      AA      SS      SS      SS      SS      TT      TT      RR      RR      II      NN      NN      GG      GG
BB      BB      AA      AA      SS      SS      SS      SS      TT      TT      RR      RR      II      NN      NN      GG      GG
BBBBBBBB      AA      AA      SSSSSS      SSSSSS      TT      TT      RR      RR      II      NN      NN      GG      GG
BBBBBBBB      AA      AA      SSSSSS      SSSSSS      TT      TT      RR      RR      II      NN      NN      GG      GG
BB      BB      AAAAAAAAAA      SS      SS      TT      TT      RR      RR      II      NN      NN      GG      GG
BB      BB      AAAAAAAAAA      SS      SS      TT      TT      RR      RR      II      NN      NN      GG      GG
BB      BB      AA      AA      SS      SS      TT      TT      RR      RR      II      NN      NN      GG      GG
BB      BB      AA      AA      SS      SS      TT      TT      RR      RR      II      NN      NN      GG      GG
BBBBBBBB      AA      AA      SSSSSSSS      SSSSSSSS      TT      TT      RR      RR      IIIIII      NN      NN      GGGGGG
BBBBBBBB      AA      AA      SSSSSSSS      SSSSSSSS      TT      TT      RR      RR      IIIIII      NN      NN      GGGGGG

```

```

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$STRING (
2 0002 0 IDENT = '1-002'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: String support library
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This routine fills a string with an input number (defaults to
36 0036 1 the string's current length) of an input character (defaults to
37 0037 1 space).
38 0038 1
39 0039 1 ENVIRONMENT: User mode, AST level or not or mixed
40 0040 1
41 0041 1 AUTHOR: R. Will, CREATION DATE: 13-Mar-79
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 R. Will, 13-Mar-79: VERSION 01
46 0046 1 01 - original
47 0047 1 1-002 - Use STR$K FILL CHAR. JBS 15-APR-1979
48 0048 1 1-003 CALL STR$CUPL_CHAR DELETE WHEN COMPILER CALLS STR$> RW 7_nov-79
49 0049 1 --
50 0050 1
51 0051 1 !<BLF/PAGE>

```

! Duplicate a character in a string
! File: BASSTRING.B32

```
53 0052 1 |  
54 0053 1 | SWITCHES:  
55 0054 1 |  
56 0055 1 |  
57 0056 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);  
58 0057 1 |  
59 0058 1 |  
60 0059 1 | LINKAGES:  
61 0060 1 |  
62 0061 1 |  
63 0062 1 |  
64 0063 1 | TABLE OF CONTENTS:  
65 0064 1 |  
66 0065 1 |  
67 0066 1 | FORWARD ROUTINE  
68 0067 1 |   BAS$STRING : NOVALUE;           ! Fill a string with a character  
69 0068 1 |  
70 0069 1 |  
71 0070 1 | INCLUDE FILES:  
72 0071 1 |  
73 0072 1 |  
74 0073 1 | REQUIRE 'RTLIN:RTLPSECT';       ! Use to declare PSECTs  
75 0168 1 |  
76 0169 1 |  
77 0170 1 |  
78 0171 1 | MACROS: NONE  
79 0172 1 |  
80 0173 1 |  
81 0174 1 | EQUATED SYMBOLS: NONE  
82 0175 1 |  
83 0176 1 |  
84 0177 1 | PSECT DECLARATIONS  
85 0178 1 |  
86 0179 1 | DECLARE_PSECTS (BAS);  
87 0180 1 |  
88 0181 1 | OWN STORAGE: NONE  
89 0182 1 |  
90 0183 1 |  
91 0184 1 | EXTERNAL REFERENCES:  
92 0185 1 |  
93 0186 1 |  
94 0187 1 | EXTERNAL ROUTINE  
95 0188 1 |   STR$DUPL_CHAR;  
96 0189 1 |
```

```

: 98      0190 1 GLOBAL ROUTINE BAS$STRING (DEST_DESC,      ! Pointer to dest str desc
: 99      0191 1      LENGTH,                                ! Number of characters
100      0192 1      INPUT_CHAR)                          ! Character to duplicate
101      0193 1      : NOVALUE =
102      0194 1
103      0195 1
104      0196 1  !++
105      0197 1  FUNCTIONAL DESCRIPTION:
106      0198 1  This routine writes LENGTH characters of CHAR into the string pointer
107      0199 1  to by DEST_DESC.  If the destination is a fixed length string, and
108      0200 1  LENGTH is greater than the length of the string, only as many CHARs as
109      0201 1  will fit are copied.  If destination is fixed length and LENGTH is
110      0202 1  less than the destination string length then LENGTH CHARs are copied
111      0203 1  and the destination is padded with blanks.  If the destination is
112      0204 1  a dynamic string, after execution of this routine the destination will
113      0205 1  have a length of LENGTH.
114      0206 1  FORMAL PARAMETERS:
115      0207 1
116      0208 1      DEST_DESC.wt.dx      pointer to destination string descriptor
117      0209 1      LENGTH.rl.v         value of number of characters to duplicate
118      0210 1      INPUT_CHAR.rb.v     value of ASCII character to duplicate
119      0211 1
120      0212 1  IMPLICIT INPUTS:
121      0213 1
122      0214 1      NONE
123      0215 1
124      0216 1  IMPLICIT OUTPUTS:
125      0217 1
126      0218 1      NONE
127      0219 1
128      0220 1  ROUTINE VALUE:
129      0221 1  COMPLETION CODES:
130      0222 1
131      0223 1      NONE
132      0224 1
133      0225 1  SIDE EFFECTS:
134      0226 1
135      0227 1      NONE
136      0228 1
137      0229 1  --
138      0230 1
139      0231 2  BEGIN
140      0232 2
141      0233 2  LOCAL
142      0234 2      CHAR : BYTE;
143      0235 2
144      0236 2  BUILTIN
145      0237 2      ACTUALCOUNT;
146      0238 2
147      0239 2  MAP
148      0240 2      DEST_DESC : REF BLOCK [8,BYTE];
149      0241 2
150      0242 2  IF ACTUALCOUNT () EQLU 3 THEN CHAR = .INPUT_CHAR ELSE CHAR = ' ';
151      0243 2
152      0244 2  STR$DUPL_CHAR (DEST_DESC [0,0,0,0], LENGTH, CHAR);
153      0245 1  END;                                !End of bas$STRING

```

```

                                .TITLE  BAS$STRING
                                .IDENT  \1-002\
                                .EXTRN  STR$DUPL_CHAR
                                .PSECT  _BAS$CODE,NOWRT, SHR, PIC,2
                                .ENTRY  BAS$STRING, Save nothing
                                SUBL2   #4, SP
                                CMPB   (AP), #3
                                BNEQ   1$
                                MOVB   INPUT_CHAR, CHAR
                                BRB    2$
                                MOVB   #32, CHAR
                                PUSHL  SP
                                PUSHAB LENGTH
                                PUSHL  DEST_DESC
                                CALLS  #3, STR$DUPL_CHAR
                                RET
                                : 0190
                                : 0242
                                : 0244
                                : 0245

```

: Routine Size: 35 bytes, Routine Base: _BAS\$CODE + 0000

```

: 154      0246  1
: 155      0247  1 END
: 156      0248  1
: 157      0249  0 ELUDOM

```

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
_BAS\$CODE	35	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

COMMAND QUALIFIERS

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

```

: Size:      35 code + 0 data bytes
: Run Time:  00:01.9
: Elapsed Time: 00:03.5
: Lines/CPU Min: 7781
: Lexemes/CPU-Min: 20687
: Memory Used: 22 pages

```

BAS\$STRING
1-002

H 3
16-Sep-1984 01:16:34

VAX-11 Bliss-32 V4.0-742

Page 5

: Compilation Complete

