



W32 App Builder Bar Code Library Programmer's Guide

I. Introduction

The CET BASIC Bar Code Library provides the subroutines necessary to generate and print bar codes from a CET program. This is accomplished by using the CALL statement Bbarcode() and the new EXTERN statement in a BASIC program. With the Bar Code Library, you can:

Generate the following bar codes

- UPC-A and E
- EAN/JAN 8 and 13
- 2 & 5 digit supplementals
- Interleaved 2 of 5
- Code 39 and Extended 39
- Codabar
- Code 128 & UCC-128
- MSI Plessey
- Code 93 and Extended 93
- Zip+4 Postnet

and print with the following printers

- Postscript
- Epson 9 pin
- IBM Proprinter 9 pin
- HP Laserjet at 150 dpi
- HP Laserjet at 300 dpi
- Epson 24 pin
- Toshiba 24 pin
- Okidata Microline Standard
- IBM Proprinter 24 pin
- HP Deskjet 150 dpi
- HP Deskjet 300 dpi
- HP Paintjet 180 dpi
- HP Laserjet 150 dpi Compatibility Mode
- HP Laserjet 300 dpi Compatibility Mode

II. The EXTERN Statement

With the release of the Bar Code Library, a new BASIC statement has been added to CET BASIC. This statement allows a variable used in a library subroutine, such as the Bar Code subroutines, to be set in a basic program. As an example, the variable BAR.WIDTH.COUNT% is used in the Bar Code subroutines and would be set in a basic program by first declaring the variable as an EXTERN and then setting the variable to an appropriate value.

```
EXTERN BAR.WIDTH.COUNT%  
BAR.WIDTH.COUNT%=1
```

III. The New OPEN Statement Option - RAWMODE

A new option - RAWMODE - has been added to the OPEN statement to provide a mode which does not modify the stream of characters going to the printer channel. A printer channel opened for bar code output should use this new option. This eliminates the NL/CR substitution for CR's which occurs when a printer channel is opened without this option. It is suggested that when both bar code and normal text are to be printed from the same program, a separate channel be opened for each type of output. The bar code channel would be opened using OPEN #1: "PRINTER", OUTPUT SEQUENTIAL, RAWMODE and the text channel using OPEN #2: "PRINTER", OUTPUT SEQUENTIAL.

IV. The Bbarcode CALL Statement

The Bbarcode routine is used to generate all of the printer control codes necessary to print a Bar Code using the printer specified in the statement variables.

```
CALL Bbarcode(ADDR OF(BCSTG$), BAR.TYPE%, BSTRG$, BTEXT$, PRTNUM%, ~
             OFFSET%, HEIGHT%, CHKSUM%, PASSES%, ADDR OF(RETCODE%))
```

Where:

BCSTG\$ is the Bar Code returned from the subroutine.
BAR.TYPE% specifies the Bar Code type to be produced by the subroutine.
BSTRG\$ is the string to be converted into a Bar Code (No spaces unless they are to be included in the Bar Code).
BTEXT\$ is the character string to be printed under the Bar Code.
PRTNUM% the number used to describe the Bar Code printer.
OFFSET% the number of characters to indent from the left margin. (use an offset of 0 for relative positioning on Laserjet)
HEIGHT% Bar Code height in 1/10 inch units or print head passes.
CHKSUM% set to 1 to enable optional checksum where applicable.
PASSES% number of overstrike passes to print (ignored with laser).
RETCODE% return code from routine - see RETCODE Section below.

A) BCSTG\$ Variable

This variable contains the bar code generated by the routine from the characters sent to it in the BSTRG\$ variable and the text sent to the subroutine in the BTEXT\$ variable. The text is always printed under the bar code exactly as it is sent in the BTEXT\$ variable with the exception of the underscore character ("_"). If a checksum is generated for the bar code, the routine replaces the first occurrence of the "_" character with the checksum character.

B) BAR.TYPE% Variable

The following numbers are used to specify the Bar Code type to be produced by the routine.

1	UPC-A*
2	UPC-E*
3	EAN/JAN-13*
4	EAN/JAN-8*
5	3 of 9 (Code 39)
6	Extended 3 of 9
7	Interleaved 2 of 5
8	Code 128
9	Codabar
10	Zip+4 Postal Codes (POSTNET Codes)
11	MSI Plessey
12	Code 93
13	Extended 93
14	UCC-128
15	HIBC

* with/without 2 or 5 digit supplementals

C) BSTRG\$ Variable

The BSTRG\$ variable contains the characters to be encoded into a bar code. It is beyond the scope of this guide to explain the different rules that apply to each type of bar code and where they are typically used. The following comments about the formatting of a UPC-A bar code should help the programmer get the feel of the flexibility of the bar code routine.

A typical string for a UPC-A bar code might be

```
BSTRG$="01234567890"      BTEXT$="0 12345 67890"
```

To print the checksum for this code, add the underscore character to the BTEXT\$ variable.

```
BSTRG$="01234567890"      BTEXT$="0 12345 67890_"
```

To encode a 2 digit supplemental of 12, you would add the supplemental to the end of the normal code.

```
BSTRG$="0123456789012"    BTEXT$="0 12345 67890 12"
```

To encode a 5 digit supplemental of 44556, format as follows

```
BSTRG$="0123456789044556" BTEXT$="0 12345 67890 44556"
```

The bar code routines will automatically switch to the proper supplemental based on the length of the input string. If the string is the wrong length, or if a letter or space is passed in, the function will return a value that indicates a bar code generation error has occurred.

D) BTEXT\$ Variable

The BTEXT\$ variable contains the text that is to be printed below the bar code. The exceptions to this rule are the Zip+4 codes and when using a PostScript printer. In both cases the text in this variable is ignored.

Passing no text causes the print head to stay at the end of the last bar code line when using a dot matrix printer. For lasers, the cursor position is restored to the original location before the bar code was sent to the printer.

E) PRTNUM% Variable

This variable contains the number which indicates the type of printer being used to print the bar code. The supported printers are listed below.

0	PostScript	11	HP Deskjet 150 dpi
1	Epson 9 pin	12	HP Deskjet 300 dpi
2	IBM Proprinter 9 pin	13	HP Paintjet 180 dpi
3	HP Laserjet at 150 dpi		
4	HP Laserjet at 300 dpi		
5	Epson 24 pin		
6	Toshiba 24 pin		
7	Okidata Microline Standard		
8	IBM Proprinter 24 pin		
9	HP Laserjet 150 dpi Compatibility Mode		
10	HP Laserjet 300 dpi Compatibility Mode		

F) HEIGHT% Variable

The bar code height parameter varies from printer to printer. For laser printers, height is specified in one tenth inch units. For most 9 and 24 pin dot matrix printers, the height is specified in one ninth inch units (one print head pass) for faster speed.

G) RETCODE% Variable

If the RETCODE% variable returns with a value other than zero, an error has occurred during the generation of the requested bar code. The following return codes are defined.

1	Return string is not allocated
2	Bar code type is not between 1 and 14
3	Length of BSTRG\$ is zero or greater than 30
4	Length of BTEXT\$ is greater than 30
5	PRTNUM% is greater than available printer types
6	OFFSET% is greater than 250
7	HEIGHT% is greater the 10 (1 inch) or less than 1
8	CHKSUM% is not zero or one
9	PASSES% is not between 1 and 5
10	Wrong number of characters or incorrect characters for bar code type
11	Buffer exceeds 32K, print smaller bar codes

V. EXTERN Variables

The following EXTERN variables are used in the Bar Code Library and may be set using the EXTERN Statement.

A) BAR.WIDTH.COUNT%

Default value of one. This variable defines the number of bar codes to print on each line. The valid range for this variable is from 1 to 6. Results are unpredictable if set outside this range. As an example, for two bar codes on the same line the EXTERN variable BAR.WIDTH.COUNT% is set to two. Before printing, two calls are made to Bbarcode() to generate the required bar codes. Between calls the variable OFFSET% is modified to position the bar codes at different offsets on the line.

B) DOTS%

Default value of two. This variable defines the dot width for each vertical bar or space (element) of the bar code. Under normal circumstances, a value of 1 is a high density bar code, 2 is medium density and 3 is low density. If high density is selected be sure the bar code reader can read it. The valid range for this variable is from 1 to 6. Results are unpredictable if set outside this range.

C) LANDSCAPE%

Default value of zero. This variable defines whether the bar code should be generated in landscape mode on the HP Laserjet. Note that the printer must be configured for landscape mode prior to printing the output from the bar code routine. The valid range for this variable is zero or non-zero.

D) CM.LPI%

Default value of 6 lines per inch. This variable defines the line per inch of the printer. The bar code routine uses this information to adjust the vertical spacing so that it matches the lines per inch of the printer. If zero lines per inch is specified then no adjustment is made. The valid range for this variable is from 0 to 20.

E) CM.CPI%

Default value of 10 cpi. This variable defines the number of characters per inch of the printer. The bar code routine uses this information to properly center the text that is printed underneath the bar codes. It is also used to properly position the bar codes and text when using the landscape mode of the HP Laserjet. The valid range for this variable is from 2 to 20.

F) CM.RATIO%

Default value is 3 to 1 ratio. This variable defines the wide to narrow ratio for Code 39, Extended Code 39 and Interleaved 2 of 5 bar codes. The available ratios are 2 to 1, 2.5 to 1, and 3 to 1. Note that when you go into the 2.5 to 1 ratio, the narrow bar width is double that of the DOTS% variable described above. To set the ratio, set this variable to 1 for 2 to 1, 2 for 2.5 to 1 and 3 for 3 to 1. This variable is used for all bar codes on a line when the return string is generated.

G) CM.CHECKSUM%

This variable is used as a storage for the first checksum value calculated in the bar code routine. This variable is 0 if there is no checksum.

H) CM.CHECKSUM2%

This variable is used as a storage for the second checksum value calculated in the bar code routine. This variable is 0 if there is no second checksum.

I) CM.128TYPE%

Default value is 0. This variable is used to force the code subset mode that is used when printing Code 128 bar codes. The available values for this variable are:

- 0 Automatic subset switching (All 128 characters)
- 1 Subset A (upper case/control characters)
- 2 Subset B (upper case/lower case characters)
- 3 Subset C (double density numbers)

J) CM.MODE128%

This variable is a storage area for the starting subset mode for the last Code 128 bar code that was generated. This variable is set to 1, 2, or 3 as described for the CM.128TYPE% variable. This is set after a call to the bar code routine when the bar code type is set to Code 128 or UCC-128.

K) CM.UPCE.MODE%

Default value of 0. This variable determines the method that the bar code routine uses for generating UPC-E bar codes. Setting this variable to the default will require 11 characters to be entered to print the 6 digit bar code. Setting the variable to 1 uses a 6 digit input string using a System of 0. When the variable is 2, System 1 will be used with a 6 digit input string.

VI. Sample Program

```
REM **** TEST PROGRAM TO PRINT BAR CODES

REM ** DECLARE EXTERN'S
  EXTERN BAR.WIDTH.COUNT%
  EXTERN DOTS%
  EXTERN LANDSCAPE%
  EXTERN CM.LPI%
  EXTERN CM.CPI%
  EXTERN CM.RATIO%

REM *** INITIALIZE EXTERNAL VARIABLES
  BAR.WIDTH.COUNT%=1
  DOTS%=3
  LANDSCAPE%=0
  CM.LPI%=6
  CM.CPI%=10
  CM.RATIO%=3

REM *** INITIALIZE CALL VARIABLES
  OFFSET%=20  \REM Print 20 characters from left margin
  HEIGHT%=4   \REM 0.4 inches high bar code
  CHKSUM%=0   \REM No Checksum
  PASSES%=1   \REM One pass printing
  PRTNUM%=12  \REM Using Deskjet at 300dpi
  BAR.TYPE%=5 \REM Code 39

INPUT BSTRG$
INPUT BTEXT$
GOSUB JUMP
END

JUMP:
  RETCODE%=-1

CALL Bbarcode(ADDROF(BCSTG$), BAR.TYPE%, BSTRG$, BTEXT$, PRTNUM%, ~
             OFFSET%, HEIGHT%, CHKSUM%, PASSES%, ADDROF(RETCODE%))

IF RETCODE%<>0 THEN PRINT "Bar Code Error # ";RETCODE% \ RETURN

PRINT "Len of String to be printed is "; LEN(BCSTG$)
OPEN #1: "PRINTER", OUTPUT SEQUENTIAL, RAWMODE
OPEN #2, "PRINTER", OUTPUT SEQUENTIAL
PRINT #1: BCSTG$
PRINT #2: "Len of String to be printed is ";LEN(BCSTG$)
CLOSE #1 \ CLOSE #2
RETURN
```

