

Hard Copy Printing with Graphics

[Alyce http://alycesrestaurant.com/](http://alycesrestaurant.com/)

Table of Contents

[Hard Copy Printing with Graphics](#)

[Hard copy printing limitations](#)

[What do the resolutions mean?](#)

[Scaling graphics for the printer.](#)

[GETBMP v PRINT](#)

[Font size.](#)

[Font placement.](#)

[Moving to the next line of text.](#)

[Text location with the stringwidth? command.](#)

[Printing columns.](#)

[Flush Caveat](#)

[Form Print Demo](#)

[Printing with MS Paint](#)

[Demo of MS Paint Printing](#)

For another look at this subject, See [Rod Bird's article](#).

Hard copy printing limitations

Liberty BASIC does not permit text to be LPRINTed and graphics to be PRINTed on the same sheet of paper. If text and images must be printed together, the text must be printed on a graphicbox or in a

graphics window along with the images, then sent to the printer with the graphics **PRINT** command.

```
#win.graphicbox "print"
```

Liberty BASIC 4.01 introduced a new, optional parameter to the graphics **PRINT** command that allows the programmer to specify the screen resolution to translate to the printed page. The choices are

- VGA
- SVGA
- XGA
- Custom Pixel Width (expressed as a number)

What do the resolutions mean?

The printer resolution is not tied to the current screen resolution. It is not necessary to set the screen resolution to the same resolution used in the **PRINT** command. The screen resolution can be VGA and graphics can be sent to the printer with the SVGA or XGA resolution, for instance.

A typical sheet of paper is 8.5" by 11". The unprintable margin differs from printer to printer, but a typical margin is one quarter inch on each side, which makes one half inch less printable area for the width ,and one half inch less printable area for the height. Assuming printable area of 8" by 10.5" the size of the printed page as translated to screen pixels is as follows:

- VGA 640 x 840
- SVGA 800 x 1050
- XGA 1024 x 1344

You can specify the number of pixels in the graphicbox that will fill the printed page width, so you MAY match the printed page width size to the width of your graphicbox, if you'd like. To use a custom pixel width:

```
#win.graphicbox "print 900"
```

Scaling graphics for the printer.

For the VGA setting, graphics that cover approximately 640 pixels wide by 840 pixels high will print on a single page of paper. Remember, the larger the resolution, the smaller each item prints on the page. A printing demo for VGA is included at the bottom of this article.

Use the numbers for location and dimensions of graphic elements that coincide with the chosen printer resolution. A line that is 640 pixels wide will span the width of a VGA page, but it will only go part of the way across an SVGA page.

GETBMP v PRINT

The graphics command GETBMP only works on the visible area. It cannot capture parts of the image that are beyond the visible boundaries of the graphicbox. This limitation does not apply to printing. The PRINT command prints portions of the graphics that are not visible on the screen, but are contained in the graphicbox.

Font size.

Fonts can be sized in points or in pixels. If you want complete control of the size and position as relates to other graphic elements, then set the size of the font in pixels. If only one number is given in the font command for the size the font is sized in points. If two numbers are given the font is sized in pixels. If the first number is 0 the default width for the font is used.

```
'font sized in points
print #1.g, "font arial 10"

'font sized in pixels, with default width
print #1.g, "font courier_new 0 12"

'font sized in pixels, with width specified
print #1.g, "font arial 12 30"
```

Font placement.

Graphic text is located at the current pen position. When the program begins the pen is at 0,0. The text location refers to the LOWER LEFT CORNER of the text. Text printed when the window is first opened is actually printed just above the visible portion of the graphicbox. It is always a good idea to use the PLACE command to position text. Remember that graphics text commands must begin with at least one backslash (or pipes) character.

```
#1.g "place 20 348"
#1.g "\Some text!"
```

Moving to the next line of text.

Each carriage return moves the pen down the page by the height of the current font. The following code prints a line of text then moves down three lines to print the next line of text.

```
#1.g "\First line of text."  
#1.g "\\\Second line is down three lines!"
```

If you want to keep track of the pen location in the Y direction as text is added to the graphicbox, multiply the number of carriage returns by the height, in pixels, of the font. Add this number to the starting location of the text.

current location = starting location + (font height x number of carriage returns)

Text location with the stringwidth? command.

Use the stringwidth? command to discover if a text string will fit, or use it to center the text. The command retrieves the width in pixels of the string according to the current font. The string of text must be contained in a string variable. Here is an example of usage:

```
name$ = "Page 1"  
#1.g "stringwidth? name$ width"
```

The width in pixels will be contained in the variable "width" after this command is made.

Printing columns.

If you need to print columns of numbers and/or text which must line up, be sure to use a monospaced (fixed-width) font. That is a font in which all characters are the same width. A variable-width font has a different width for each character, so an uppercase 'W' will take up more space than a semicolon, for instance. One example of a fixed-width font is "courier new."

```
#1.g "font courier_new 0 22"
```

Flush Caveat

Liberty BASIC users have reported inconsistent printed results when they've issued multiple flush commands. For best results issue all drawing commands, before you issue a single flush command and then print.

Thanks for this advice, Janet.

Form Print Demo

```
nomainwin
'hard copy form printing with various fonts,
'colors, and graphics

'one quarter inch margin each side means:
'max width for vga on 8.5" paper is 640
'max width for vga on 11" paper is 840
'test on your printer to get exact numbers

WindowWidth=670:WindowHeight=DisplayHeight-100
UpperLeftX=10:UpperLeftY=10

'graphicbox dimensions, add 20 to width for scrollbar:
graphicbox #1.g, 0,0,660,840
open "Printing a Form" for window as #1
#1 "trapclose [quit]"

#1.g "vertscrollbar on 0 840"
#1.g "down; fill white"
#1.g "color blue; backcolor white"
#1.g "size 3"

'print a blue box as a border:
#1.g "place 0 0; box 636 834"

'print a headline:
#1.g "color black"
#1.g "font arial 14 28 bold"
name$ = "Headline!"
#1.g "stringwidth? name$ width"
'center headline by subtracting its
'width in pixels from the total width
'and dividing the difference by two
xLoc = (640-width)/2
#1.g "place ";xLoc;" 30"
#1.g "\Headline!"
```

```
'print a reddish line:  
#1.g "color 255 100 20"  
#1.g "size 4"  
#1.g "line 4 40 632 40"  
  
'print some text - each line is 20 pixels high  
#1.g "color darkgreen; place 10 80"  
#1.g "font times_new_roman 0 20"  
#1.g "\Here is a line of text."  
#1.g "\Here is a second line of text."  
#1.g "\More text after a blank line."  
  
'print a series of circles:  
#1.g "color darkcyan; backcolor cyan;size 2"  
for i = 40 to 600 step 50  
    #1.g "place ";i;" 260"  
    #1.g "circlefilled 25"  
next  
  
'print columns  
'use fixed width font so that text lines up properly  
#1.g "color black; backcolor white"  
#1.g "font courier_new 0 14"  
#1.g "place 60 540"  
#1.g  
"\Item           Quantity       Price Each       Total"  
#1.g  
"\Hamburger      1             $ 3.34           $ 3.34"  
#1.g  
"\Fries          2             $ 2.00           $ 4.00"  
#1.g  
"\Soda           23            $ 1.00           $23.00"  
#1.g  
"\_____"  
#1.g  
"\Total          $30.34"  
  
'print footer  
#1.g "color darkgray"  
#1.g "font verdana 0 13 italic"  
  
name$ = "Page 1"  
#1.g "stringwidth? name$ width"  
'center footer by subtracting its  
'width in pixels from the total width
```

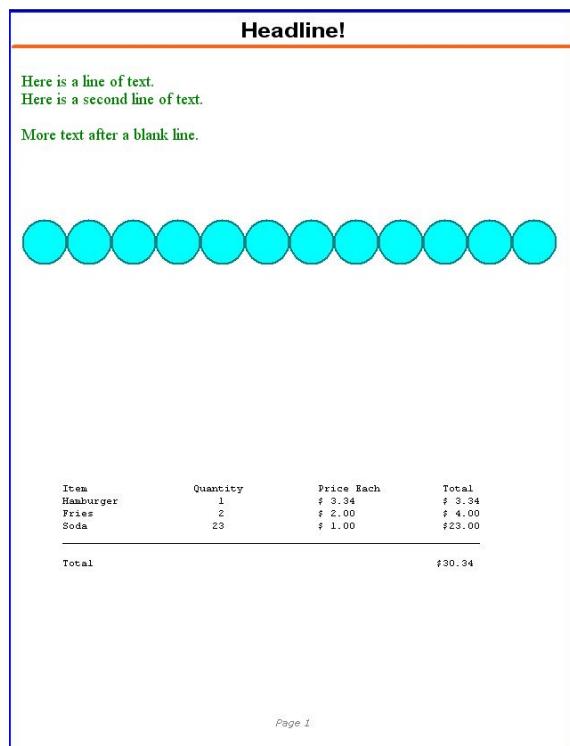
```
'and dividing the difference by two
xLoc = (640-width)/2
#1.g "place ";xLoc;" 804"
#1.g "\Page 1"
#1.g "flush"

wait
'send to printer at 640x880 page resolution:
#1.g "print vga"
dump

wait

[quit] close #1:end
```

The code above produces this printout:



Printing with MS Paint

All Windows computers have MS Paint installed. You can ask Paint to print graphics. This is done with the **RUN** command. Paint reads the command line and acts on the instructions it contains. The command line should include the filename of the saved BMP and the "/p" switch that instructs Paint to send output to the printer. Liberty BASIC allows us to specify a mode for running the file. If we include HIDE, Paint will do the work without displaying a window, so it is invisible to the user.

Filenames should be enclosed in quotes so that Paint can properly parse the command line. The ASCII character value for the quotation mark is chr\$(34).

The syntax is as follows:

```
run "mspaint.exe " + chr$(34) +
"c:\folder\filename.bmp" + chr$(34) + " /p", HIDE
run "mspaint.exe " + chr$(34) + bmpFile$ + chr$(34) + " /p", HIDE
```

Demo of MS Paint Printing

```
'demo of hard-copy graphics printing
'with MS Paint
nomainwin
open "Graphics Print Test" for graphics_nsb as #1
#1 "trapclose [quit]"
#1 "down; fill yellow;color darkblue"
#1 "backcolor pink;size 5;place 150 150"
#1 "circlefilled 100;flush"
#1 "getbmp test 0 0 300 300"

RunFile$=DefaultDir$+"\atest.bmp"
bmpsave "test",RunFile$
run "mspaint.exe " + chr$(34) + RunFile$ + chr$(34) + " /p", HIDE
#1 "down; fill white;color darkblue;backcolor white;place 10 100"
#1 "\Graphics sent to MSPaint for printing."
wait
[quit]
close #1:unloadbmp "test"
end
```