

Printing 2D barcode using fonts from Visual Basic 6

Table of Contents

Printing Tips	1
Sample Code	2

In Visual Basic you can "print" to more than just a physical printer. You can print to a form, a picture control, and also to any physical printer connected to your PC or to any shared network printer to which your PC is connected.

The following three statements print "test" to a form, a picture box, and a printer:

```
form1.print "test"  
picture.print "test"  
printer.print "test"
```

Although three share the identical syntax, they produce very different results. Understanding the difference is important to produce high quality barcodes.

In the first prints, the drawing surface is the screen. A screen has a nominal resolution at 96 dpi, meaning that 1 pixel has a width of approximately 10 *mils*¹. A typical laser printer on the contrary has a resolution of 600 dpi, which translates to 1.67 mils per pixel. The difference means that you can achieve much high precision by printing directly to the printer.

In most occasions, the barcode images drawing on screen is for visual demonstration only. It is unnecessary for the barcode on the screen to be readable by a scanner.

Some customers mistakenly believe that the drawing is always scalable among different devices. They print the barcode to a picture box first, and then print the picture box out to the printer. Although vector graphics is scalable among devices, raster images do not. As a result, the barcodes produced have very low quality.

In summary, never print barcodes to screen first and transfer the resulted images later to the printer. Always print to the printer directly.

Printing Tips

We'll briefly review the printing features provided by the VB here. These samples show some of the most typical printing tasks you'd likely need.

Scale Mode

The ScaleMode determines the measurement unit you are going to use. You can set the scale mode to pixels, twips, inches, centimeters, points, character, and millimeters. Because of the resolution, the pixels scale mode is not portable among devices. 100 pixels appear much longer on a screen than on a printer.

EndDoc

When you print to a form or picture box, the image appears immediately. This is not the case for the printer. You must explicitly tell the printer where a page stops. Call EndDoc method when you finish all the drawings.

¹1 mil = 1/1000th inch.

Printing at a specified location

If you'd like to print the barcode at a specified location, that is say 6 inch down and 1 inch over, you must set the `currentx` and `currenty` properties accordingly, as the code below:

```
printer.scalemode = vbInches
printer.currentx = 1.0
printer.currenty = 6.0
```

Print multiple lines

Visual Basic moves the printer cursor down a line after printing a line. Unfortunately the VB always moves the cursor to the left side of the printer. To have the lines started from the same `currentx`, you need to save this property and set it before calling `Print`:

```
pos_x_save = printer.currentx
printer.print "line 1"
Printer.currentx = pos_x_save
printer.print "line 2"
printer.currentx = pos_x_save
```

Parse the barcode string

The `print` method does not recognize line feed and carriage return characters. On the other side, the `Datamatrix/PDF417` font encoder returns a single string separating lines with line feed and carriage return characters. You can specify the line separator with other strings, but the encoder always returns a single string. To get line by line you can use the `split` function provided by VB:

```
Dim separator As String
Dim lines() As String

separator = vbCrLf
lines() = split(encoder_results, separator)
MsgBox "Total " & UBound(lines) & "lines."
```

Sample Code

The code below prints a Data Matrix barcode at a point 1.5 inches to the left edge of the page, 3.0 inches from the top. The barcode string is stored in an edit control called `BarcodeEdit`. Replace `BarcodeEdit.Text` with your barcode string.

```
Private Sub Command_Print_Barcode_Click()
'To print the barcode, we recommend using the built-in Printer object
'Do not print to a picture box and transfer it to printer. That results
'in a low quality barcode.

'You can change it to other values to adjust the barcode size.
Printer.FontSize = 12
Printer.FontName = "MRV DataMatrix"

'We use inch as measurement unit here.
Printer.ScaleMode = vbInches

'We want to print the barcode at point (1.5 inch, 3 inch)
'Note - printer usually has a margin 0.25" at four directions.
'You may consider this fact when you set the printer cursor
Dim StartX, StartY As Double
StartX = 1.5
StartY = 3#
Printer.CurrentX = StartX
Printer.CurrentY = StartY
```

```
'The Printer always go back to the left edge of the page after executing
'a print statement. Unless we print the barcode to the left ledge of the page,
'we have to parse the barcode string into lines, and print them one by one.
'Fortunately, VB provides a nice helper function for this purpose.
Dim separator As String
Dim lines() As String
separator = vbCrLf
lines() = Split(BarcodeEdit.Text, separator)

'Now we do the actual print line by line
Dim i As Integer
For i = 0 To UBound(lines) - 1
    Printer.Print lines(i)
    'Note: the Printer always go back to the beginning of the next line
    'after executing Print. We need to set CurrentX every time after printing a line
    Printer.CurrentX = StartX
Next

'EndDoc must be called to actual print
Printer.EndDoc

End Sub
```

