

Re: Get RAW Bitmap Data from a file

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2008-05/msg00792.html>

- *From:* RB0135 <robear@xxxxxxxxxxxxxxx>
 - *Date:* Sun, 18 May 2008 05:18:07 -0700 (PDT)
-

Armin,

Thanks again for your reply.

Ok.. here is some more information. There is too much code to paste here on the printing and I didnt think at the time that was all that relevant, when I am sure I was NOT pulling the correct data out:

I looked at my code again and realised after the line `rawImage = New Byte(size - 1)`, I had to braces `{ }` after the code, which was left over from the C# conversion. However, this made my source compile. When I removed them, and used the `Redim`, I got the same result anyway.

The Roll Printer is an Epson TM-T88 and a Posiflex PP7000 (The Posiflex uses the Epson command set). Both are giving the same result, so that is why I am also presuming I have the wrong raw data of the Bitmap

I am using code from this MS Knowledgebase that allows printing RAW data to the printer, rather than a driver (there isnt a driver for these Roll printers and using the Generic/text driver does not allow all characters to be transmitted to it). Here is the link:
<http://support.microsoft.com/?id=322090>

To get the information to the printer, I need to send it the Define Graphic Download command:

```
RawPrinterHelper.SendStringToPrinter(pDocket.PrinterSettings.PrinterName.ToString(),  
Chr(29) & "*" & Chr(bmBitmap.Width / 8) & Chr(bmBitmap.Height / 8))  
(this routine is in the above KB article, and the Chr(29) command is  
the Epson command to turn on Define Graphic Download).
```

Here is a section of the manual for the Epson:

`GS * x y d1 ... d(x * y * 8)` defines a downloaded bit image using `x * 8` dots in the horizontal direction and `y * 8` dots in the vertical direction. Once a downloaded bit image has been defined, it is available until another definition is made; `ESC &` or `ESC @` is executed; the

Re: Get RAW Bitmap Data from a file

printer is reset; or the power is turned off. When this command is executed, the user-defined characters are cleared. The default setting is no downloaded bit image defined.

Back to my program.....

Once I have the rawimage Byte array, I am sending this through that Epson Command, via this command from the KB Article:

```
' Allocate some unmanaged memory for those bytes.  
pUnmanagedBytes = Marshal.AllocCoTaskMem(rawimage.Length)  
' Copy the managed byte array into the unmanaged array.  
Marshal.Copy(byBytes, 0, pUnmanagedBytes, image.Length)
```

```
RawPrinterHelper.SendBytesToPrinter(pDocket.PrinterSettings.PrinterName.ToString(),  
pUnmanagedBytes, rawimage.Length)
```

Of course, I get the right sized image being printed on the printer, but looks NOTHING like it should. As I mentioned, it just seems like a lot of lines (within the size of the image) like an out of sync TV picture...

This is the example they give:

```
x=16: y=5  
PRINT #1, CHR$(&H1D);"*";CHR$(x);CHR$(y);  
FOR i=1 TO x*y*8  
READ a$: d=VAL("&H"+a$)  
PRINT #1, CHR$(d);  
NEXT i  
PRINT #1, CHR$(&H1D);"/";CHR$(0);CHR$(&HA);␣ Normal  
END
```

```
DATA FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,C0,00,00,00,03,C0  
DATA 00,00,00,03,CF,FF,FF,FF,F3,CF,FF,FF,FF,F3,CF,FF  
DATA FF,FF,F3,CF,FF,FF,FF,F3,CF,FF,FF,FF,F3,CF,C0,FC  
DATA 03,F3,CF,C0,FC,03,F3,CF,C0,FC,03,F3,CF,C0,FC,03  
DATA F3,CF,C0,FC,03,F3,CF,C0,FC,03,F3,CF,C0,FC,03,F3  
DATA CF,C0,FC,03,F3,CF,C0,FC,03,F3,CF,C0,00,03,F3,C0  
DATA 00,00,00,03,C0,FF,00,03,F3,C3,FF,C0,03,F3,C7,FF  
DATA E0,03,F3,C7,FF,F0,03,F3,CF,FF,F8,03,F3,CF,FF,FC  
DATA 03,F3,CF,E3,FE,03,F3,CF,C1,FF,03,F3,CF,C0,FF,83  
DATA F3,CF,C0,7F,C7,F3,CF,C0,3F,FF,F3,CF,C0,1F,FF,F3  
DATA CF,C0,0F,FF,E3,CF,C0,07,FF,E3,CF,C0,03,FF,C3,C0  
DATA 00,00,FF,03,C0,00,00,00,03,C0,3F,FF,FC,03,C0,FF  
DATA FF,FF,03,C3,FF,FF,FF,C3,C7,FF,FF,FF,E3,C7,FF,FF  
DATA FF,E3,CF,FF,FF,FF,F3,CF,F0,00,0F,F3,CF,C0,00,03  
DATA F3,CF,C0,00,03,F3,CF,C0,00,03,F3,CF,C0,00,03,F3
```

Re: Get RAW Bitmap Data from a file

DATA CF,C0,00,03,F3,CF,C0,00,03,F3,CF,C0,00,03,F3,CF
DATA C0,00,03,F3,CF,C0,00,03,F3,C0,00,00,00,03,C0,00
DATA 00,00,73,C0,00,00,03,C3,C0,00,00,1E,03,C0,00,00
DATA 70,03,C0,00,03,C0,03,C0,00,1E,00,03,C0,00,78,00
DATA 03,C0,03,C0,00,03,C0,0E,00,00,03,C0,78,00,00,03
DATA C3,C0,00,00,03,CE,00,00,00,03,C0,00,00,00,03,CF
DATA FF,FF,FF,F3,CF,FF,FF,FF,F3,CF,FF,FF,FF,F3,CF,FF
DATA FF,FF,F3,CF,FF,FF,FF,F3,CF,FF,FF,FF,F3,CF,C0,0F
DATA C0,03,CF,C0,0F,C0,03,CF,C0,0F,C0,03,CF,C0,0F,C0
DATA 03,CF,C0,0F,C0,03,CF,E0,1F,C0,03,CF,FF,FF,C0,03
DATA CF,FF,FF,C0,03,C7,FF,FF,80,03,C7,FF,FF,80,03,C1
DATA FF,FE,00,03,C0,3F,F0,00,03,C0,00,00,00,03,C0,0F
DATA FF,F0,03,C0,FF,FF,FF,03,C3,FF,FF,FF,C3,C7,FF,FF
DATA FF,E3,C7,FF,FF,FF,E3,CF,FF,FF,FF,F3,CF,F0,00,0F
DATA F3,CF,C0,00,03,F3,CF,C0,00,03,F3,CF,C0,00,03,F3
DATA CF,C0,00,03,F3,CF,C0,00,03,F3,CF,C0,00,03,F3,CF
DATA C0,00,03,F3,CF,F0,00,0F,F3,CF,FF,FF,FF,F3,C7,FF
DATA FF,FF,E3,C7,FF,FF,FF,E3,C3,FF,FF,FF,C3,C0,FF,FF
DATA FF,03,C0,0F,FF,F0,03,C0,00,00,00,03,C0,FF,00,03
DATA F3,C3,FF,C0,03,F3,C7,FF,E0,03,F3,C7,FF,F0,03,F3
DATA CF,FF,F8,03,F3,CF,FF,FC,03,F3,CF,E3,FE,03,F3,CF
DATA C1,FF,03,F3,CF,C0,FF,83,F3,CF,C0,7F,C7,F3,CF,C0
DATA 3F,FF,F3,CF,C0,1F,FF,F3,CF,C0,0F,FF,E3,CF,C0,07
DATA FF,E3,CF,C0,03,FF,C3,C0,00,00,FF,C3,C0,00,00,00
DATA 03,C0,00,00,00,03,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF

I cant show you the graphic it prints. I can certainly email you the whole manual in PDF..

The bitmap that I am using has come from POSIFLEX and is what they state to test for Printing to the printer.

Now, I understand the 1bpp, but it seems the printer takes one byte per 8 pixels, so I was presuming that the RAW data is in the correct format. Also, when running the LockBits, it only allows me to use the format1bpp parameter. Any other format, even DONTCARE fails with INVALID PARAMETER.

You mentioned you dont know what I do with the rawimage data. As I mention here, I send it to the printer, but I am wondering and asking, do I need to process it further before sending it to the printer

I will check the link you gave me.. much appreciated.

If you can offer anymore assistance, or need extra information, please let me know.

Thanks,
Robert

On May 18, 5:54 pm, "Armin Zingler" <az.nos...@xxxxxxxxxxx> wrote:

Re: Get RAW Bitmap Data from a file

Re: Get RAW Bitmap Data from a file

"RB0135" <rob...@xxxxxxxxxxxxxx> schrieb

Thanks for your reply..

The code I used and presented in the original post was from the MS Robotic Lib SDK and does compile OK..

I understand your point about the redim, but mine compiles fine without it...

Strange. I tried it in VB 2008 Express and, as I expected, the syntax is not valid because, in VB, in opposite to C#, there is no "[]" for arrays. Therefore, "New Byte(size -1)" is interpreted as a constructor, but a Byte does not have such a constructor.

Are you really using VB.Net? ;-)

The part of code using the MemoryStream and a picture box, was another Idea I was testing togethe bytes, but it is not relevant to the overall issue.

The various bits of code are tests I am trying to do togetPIXEL DATAin the right format and the righdatafrom the BMPfile.

The datais transferred to the Roll Printer in Bytes, where it interperets it. Each bit is on or off.

All this is explained in the original post.

Not this way, nevertheless I got it now.

Thats what I need the rawimagedatato be.. A bit representation of the image from the BMPfile.. I dont want the header information, just, I suppose, what the image looks like in Bytes.

Re: Get RAW Bitmap Data from a file

Does this make sense?

Yes, but you don't show how you pass it to the roll printer.

From your reply, the lockbits is the right way to go, but I am not getting the representation needed from the file.

How do you check this? I don't see any code that prints anything.

I have read that BMP's store the data from bottom to top, left to right (or similar).. Is this the case?

Do I need more processing of the raw image data to get the representation of the image from it?

I can only guess that you expect one byte per pixel, but you specified `Imaging.PixelFormat.Format1bppIndexed` as the format which means one bit per pixel (1 byte = 8 pixel). I don't know how you process the data in `rawImage`. In addition, IIRC, there is no indexed bitmap support in a certain area that I unfortunately don't remember currently.

It's always good to have a look here: <http://www.bobpowell.net/>
In the GDI+ FAQs, there is an example of creating a 1bpp image. There is also something about directly accessing pixel data. Maybe this helps in any way, too.

And we have `m.p.d.framework.drawing`.

Though, I'm still interested in how you really process 'rawImage'.

Armin