Subject: Re: DrawWin32.cpp: PrintDraw::InitPrinter()
Posted by Tom1 on Mon, 24 Nov 2008 11:41:22 GMT
View Forum Message <> Reply to Message

I'm testing this on Windows XP. Luckily the Epson Stylus Photo 1290 driver has a preview option in XP and it does not even require the printer to exist to be able to install the driver and use its preview option.

There are two separate problems:

1) Only about 60% of width and about 60% of height of the printable area is covered by the printout.

2) The DrawImage(rect,image) produces a broken image probably due to the resolution mismatch (360x360 versus 600x600). It does not get correctly re-scaled from 600 dpi to 360 dpi in the GDI or the printer driver.

Please see the attached files broken.png and fixed.png.

Current code:

```
::SetMapMode(handle, MM_ANISOTROPIC);
::SetViewportExtEx(handle, inchPixels.cx, inchPixels.cy, NULL);
::SetViewportOrgEx(handle, 0, 0, NULL);
::SetWindowExtEx(handle, 600, 600 , NULL);
::SetWindowOrgEx(handle, 0, 0, NULL);
```

Result with current code:

My slightly enhanced solution looks like this:

```
::SetMapMode(handle, MM_ANISOTROPIC);
int dpi=min(inchPixels.cx,inchPixels.cy);
::SetWindowExtEx(handle, dpi*pagePixels.cx/inchPixels.cx, dpi*pagePixels.cy/inchPixels.cy,
NULL);
::SetWindowOrgEx(handle, 0, 0, NULL);
::SetViewportExtEx(handle, pagePixels.cx, pagePixels.cy, NULL);
::SetViewportOrgEx(handle, 0, 0, NULL);
pagePixels.cx=dpi*pagePixels.cx/inchPixels.cx;
pagePixels.cy=dpi*pagePixels.cy/inchPixels.cy;
inchPixels.cx=dpi;
inchPixels.cy=dpi;
```

The result after the above changes:

With the above change, it works with the standard DrawImage.

Just an opinion, but I think the best printing quality can be obtained using the printer's hardware resolution (and optionally dropping it with integer dividers if it is excessively high). For correct page size and scaling I always use Draw::GetPagePixels() and Draw::GetPixelsPerInch() to get the map scale right on both screen and printer.

(This is irrelevant for the current issue at hand, but since it popped up: The SetDIBitsToDevice() uses hardware resolution and does not respect these Viewport/Window mappings except maybe for the starting point. StretchDIBits(), however, does work with the mappings and is also pretty economic way to put low resolution images to high resolution printers, as it does not necessarily pre-rescale the image to the printer resolution before sending it to the printer.)

// Tom

File Attachments

1) broken.png, downloaded 1217 times
2) fixed.png, downloaded 1198 times