
Subject: Re: Draw::DrawImageOp optimization bug
Posted by [Tom1](#) on Thu, 04 Dec 2008 10:07:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:

Uh oh. DrawImage was the reason we have started all this stuff anyway.

Why do you suggest to throw all of that now?

Note that it is at least supposed to fix those line artifacts problem in the first place.

I do not wish to throw away the BeginNative/EndNative/Native functions. Definitely not. They are really the key to use the native resolution in printing. That's great and they should stay there.

The problem is that DrawImageOp and DrawDataOp use the BeginNative/EndNative pair and the Native translations as if the coordinates input would be dots in all cases. This makes the coordinate space different for vector objects and the image objects.

Based on your code, I assume, you wanted to put the image to the printer on the native resolution. What happened was that if native mode was already on, the Native() calls did a second re-mapping of coordinates causing error in position and scale.

Basically switching the mode and the Native() calls would have been OK, if a check of the native recursion level was done and the Native() calls were conditional (if(native==1)) after switching. Anyway, the correct mapping of pixels to coordinate space was already achieved with the change that skipped ::SetDIBitsToDevice() call and used some other 'bitmap thing'. So I concluded that those calls are not needed in DrawImageOp and DrawDataOp at all. I removed the calls from there, tested it and it worked. Both in native and in dots mode.

Now, if you do these changes I suggest, I guess the situation will be that we should have a Draw:: for printing that by default is mapped with 600 dpi points and works OK with all printers we have tested so far. Additionally, if BeginNative is called, it will completely switch to printers native dpi mode and provides a consistent printing area at the printers native resolution instead of the 600 dpi.

The changes to GetPagePixels and GetPixelsPerInch support this. They will provide the correct values for the drawing area also after switching to native mode. Those changes will not break dots mode in any way.

Regards,

Tom
