Subject: Re: Native DPI
Posted by mirek on Mon, 15 Jun 2015 12:01:58 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Mon, 15 June 2015 00:00Hi,

From a wider point of view, this sounds like the right way to go. Applications with standard controls scale up beautifully in general without any additional effort, and that is important.

However, from my narrow point of view, I additionally need to be sure that I can still have the low level access to "pixel for pixel" graphics. The important questions are:

1. What mechanisms are used for determining the current (per monitor) DPI (or scaling percentage) and how is this information exposed through upp?
2. How do I create a Ctrl that works at the native "pixel-for-pixel" resolution regardless the current DPI?

We we still speaking theoretically, but the answers would be:

1. Not yet 100% sure... :)
2. You call something like "HiDPIAware();" in constructor.

Quote:
As I understand it, this automatic scaling is an attempt to remove the concept of pixel as a measure of distance and replace it with a roughly one quarter of a millimeter unit (more precisely one pixel at 96 dpi).

I would not mix any physical resolution into it (I never considered the "dpi" concept viable for display). For non-aware Ctrl, one "logical" pixel would equal to two "physical" pixels, that is all.

(OK, I know that this might burn us again if "quadruple HiDPI" ever happens... :)

Quote:
I bet I will use Painter extensively to render correctly scaled graphics within my controls with graphical content. The Painter rendered content will obviously need to be rendered at 1:1 pixel ratio on screen.

Yes.

Mirek