
Subject: What is the best way to handle scaling icons?
Posted by [cbpporter](#) on Thu, 09 Aug 2018 13:34:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

I've been having a lot of troubles with scaling under Windows. I went though the entire GUI to touch it up, testing under 125%, 150%, 175% and 200% scaling. Updated CodeEditor too to properly scale and be aesthetically pleasing.

The one problem is icons. What is the best way to handle this?

As far as I can tell, icons get scaled too, but they look jagged with some filtering smoothing.

Potential solutions:

1. Have multiple icon sizes? 16, 24, 32? Does U++ support multi-res icons or should I do this in code?
2. Use SVG icons? One could do on applications startup some size detection and render SVG icons into some sort of Image?

One problem with this is that RenderSVGImage from Painter package renders a background. I cleared the background and so far it looks like it is working properly. I'll have to render that image onto a bunch of GUI elements to see if transparency and sub pixel rendering is looking good.

Maybe RenderSVGImage should take the background clear color as a parameter?

Subject: Re: What is the best way to handle scaling icons?
Posted by [mirek](#) on Fri, 10 Aug 2018 18:57:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

We were trying to resolve this with hi-dpi support.

Current solution is as follows:

When the main font height reaches certain pixel size, U++ is switched into HIDPI mode. Now icons in .iml now have resolution setting:

- normal
- hidpi
- unknown

In non-HIDPI mode, all hidpi icons in all .iml are scaled down 2x. In HIDPI mode all normal icons in all .iml are scaled 2x up, using special algorithm that I have picked from console emulators... (so it tends to smooth lines in artificial icons).

There are also helper functions like Zx/Zy (scale numbers based on main font size) and DPI (scales numbers 2x in hidpi mode).

Mirek
