Subject: Re: Documentation how-to...
Posted by cbpporter on Sun, 21 Sep 2008 09:38:38 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 21 September 2008 11:51
I am working on automated "repair tool".

Great! Anyway, SVN is down, so I have to wait with docs.

Quote:
Well, I can create it... Though I doubt about "wchar agnosticity"

It is wchar agnostic in the sense that I want to write the low level renderer first, which on Win32 at least will operate on 16 bits. Even if wchar is 32 bits, for rendering purposes, the latest when WinAPI is called, it will have to be converted to 16 bits, right? We need a not to complicated text renderer, because WinAPI can't really handle mixed font rendering in CJK context at least. I haven't benchmarked it yet, but performance should be equal to previous performance plus the cost of a traverse over the text that must be displayed, so I hope it will be OK. If it is not, I'll need to think of a way to keep the old text rendering, and activate the new one only if the application must be able to display full CJK.

I also started studying some shape recognition, to be able to trace CJK characters and render them manually, for systems that will never be able to have all the fonts installed, but I'm not sure about legal issues here. The traced characters will have their own style, which is distinct form all available styles in a manner that "Arial" font is distinct from "Dotum" font, so it will look very similar on first sight, but when examining the weight of curves, it will be quite different, but still, my data will come from an (or more probably a more than one) available fonts, strictly by pixel analysis, and I don't want to breach any copyright clauses.

This approach has some advantages:
1.It is a viable and lightweight fall back when native rendering using (very large) fonts is not available or not desired.
2.The huge number of CJK characters is reduced to some hundreds of composing characters. This way the rendering for most characters will definable in a couple of bytes.
3.It allows easy definition of gaiji (I'm not going to define any, but I'll include a GUI in the future in my application).
4.It allows easy animation of stroke order.

Disadvantages:
1. Uncertain legal issue.
2. Generating the initial data set will probably take days.
3. Professional fonts will probably have better font hinting.