Subject: Re: Font alignment issues
Posted by mirek on Mon, 03 Aug 2009 15:13:34 GMT
View Forum Message <> Reply to Message

chickenk wrote on Mon, 03 August 2009 10:31OK I recompiled the new sources, the bad news are it does not change anything in my case. As far as I use 88 dpi instead of 96, the display is wrong...

I am not sure it's worth taking so much of your time about that. It may be a rare case coming from my configuration (I'm still wondering what have I done to have such differences, I can't see...).

If you think it's important to fix it because it applies to many people, I'll be glad to help you. But don't waste your time just for my case, I can live with it.

Lionel

Well, I am afraid it might kick us somewhere else... Font metrics is foundation, there is no place for weird behaviour...

Another attempt:

Draw::FontFc.cpp:

```
FcPattern *CreateFcPattern(Font font)
{
 LTIMING("CreateXftFont");
 double sina, cosa;
 int hg = abs(font.GetHeight());
 if(hg == 0) hg = 10;
 String face = font.GetFaceName();
 FcPattern *p = FcPatternCreate();
 FcPatternAddString(p, FC_FAMILY, (FcChar8*)~face);
 FcPatternAddInteger(p, FC_SLANT, font.IsItalic() ? 110 : 0);
 FcPatternAddInteger(p, FC_PIXEL_SIZE, hg);
 FcPatternAddInteger(p, FC_DPI, 96);
 FcPatternAddInteger(p, FC_WEIGHT, font.IsBold() ? 200 : 100);
 FcPatternAddBool(p, FC_MINSPACE, 1);
 FcResult result;
 FcConfigSubstitute(0, p, FcMatchPattern);
 FcDefaultSubstitute(p);
 FcPattern *m = FcFontMatch(0, p, &result);
 FcPatternDestroy(p);
 return m;
}
```

CtrlCore/DrawTextX11.cpp:

```
XftFont *CreateXftFont(Font font, int angle)
{
 LTIMING("CreateXftFont");
 XftFont *xftfont;
 double sina, cosa;
 Std(font);
 int hg = abs(font.GetHeight());
 if(hg == 0) hg = 10;
 int i = font.GetFace();
 if(i < 0 || i >= Font::GetFaceCount())
  i = 0;
 String face = font.GetFaceName();
 FcPattern *p = FcPatternCreate();
 FcPatternAddString(p, FC_FAMILY, (FcChar8*)~face);
 FcPatternAddInteger(p, FC_SLANT, font.IsItalic() ? 110 : 0);
 FcPatternAddInteger(p, FC_PIXEL_SIZE, hg);
 FcPatternAddInteger(p, FC_DPI, 96);
 FcPatternAddInteger(p, FC_WEIGHT, font.IsBold() ? 200 : 100);
 FcPatternAddBool(p, FC_MINSPACE, 1);
 if(angle) {
  FcMatrix mx;
  Draw::SinCos(angle, sina, cosa);
  mx.xx = cosa;
  mx.xy = -sina;
  mx.yx = sina;
  mx.yy = cosa;
  FcPatternAddMatrix(p, FC_MATRIX, &mx);
 }
 FcResult result;
 FcPattern *m = XftFontMatch(Xdisplay, Xscreenno, p, &result);
 if(font.IsNonAntiAliased() || gtk_antialias >= 0) {
  FcPatternDel(m, FC_ANTIALIAS);
  FcPatternAddBool(m, FC_ANTIALIAS,
                   font.IsNonAntiAliased() ? FcFalse : gtk_antialias ? FcTrue : FcFalse);
 }
 if(gtk_hinting >= 0) {
  FcPatternDel(m, FC_HINTING);
  FcPatternAddBool(m, FC_HINTING, gtk_hinting);
 }
 const char *hs[] = { "hintnone", "hintslight", "hintmedium", "hintfull" };
 for(int i = 0; i < 4; i++)
  if(gtk_hintstyle == hs[i]) {
   FcPatternDel(m, FC_HINT_STYLE);
   FcPatternAddInteger(m, FC_HINT_STYLE, i);
  }
 const char *rgba[] = { "_", "rgb", "bgr", "vrgb", "vbgr" };
```

```
 for(int i = 0; i < __countof(rgba); i++)
  if(gtk_rgba == rgba[i]) {
   FcPatternDel(m, FC_RGBA);
   FcPatternAddInteger(m, FC_RGBA, i);
  }
 xftfont = XftFontOpenPattern(Xdisplay, m);
 FcPatternDestroy(p);
 return xftfont;
}
```

(We try to tell him in both cases to use 96dpi - which is fine, because at the time we have are only using correct pixel size).

Mirek
