Subject: Re: Strange issue with text in Painter
Posted by Tom1 on Thu, 17 Jan 2019 13:30:01 GMT

mirek wrote on Thu, 17 January 2019 15:16Tom1 wrote on Thu, 17 January 2019 14:12mirek wrote on Thu, 17 January 2019 14:58...
It starts to look like Win32 does not like the sharing of HDC... (despite of what the documentation says).

What about this:


```
void RenderCharacterSys(FontGlyphConsumer& sw, double x, double y, int ch, Font fnt)
{
 RLOG(CoWork::GetWorkerIndex() << " RenderCharSys " << fnt << " " << ch << " " << (char)ch);
 static Atomic h;
 h++;
 HFONT hfont = GetWin32Font(fnt, 0);
 VERIFY(hfont);
 if(hfont) {
  HDC hdc = CreateIC("DISPLAY", NULL, NULL, NULL);
  HFONT ohfont = (HFONT) ::SelectObject(hdc, hfont);
  GLYPHMETRICS gm;
  MAT2 m_matrix;
  memset(&m_matrix, 0, sizeof(m_matrix));
  m_matrix.eM11.value = 1;
  m_matrix.eM22.value = 1;
  int gsz = GetGlyphOutlineW(hdc, ch, GGO_NATIVE|GGO_UNHINTED, &gm, 0, NULL,
&m_matrix);
  if(gsz < 0) {
   LOGFONT lf;
   int ret = GetObject(hfont, sizeof(lf), &lf);
   RLOG("Failed " << CoWork::GetWorkerIndex() << " RenderCharSys " << fnt << " " << ch << " "
<< (char)ch
             << ", real: "  << " " << lf.lfFaceName << " " << lf.lfHeight);
   RDUMP((void *)hdc);
   RDUMP((void *)hfont);
   RDUMP(GetLastErrorMessage());
  }
  VERIFY(gsz >= 0);
  if(gsz < 0)
   return;
  StringBuffer gb(gsz);
  gsz = GetGlyphOutlineW(hdc, ch, GGO_NATIVE|GGO_UNHINTED, &gm, gsz, ~gb, &m_matrix);
  if(gsz < 0) {
   LOGFONT lf;
   int ret = GetObject(hfont, sizeof(lf), &lf);
   RLOG("Failed2 " << CoWork::GetWorkerIndex() << " RenderCharSys " << fnt << " " << ch << " "
```

```
<< (char)ch
           << ", real: "  << " " << lf.lfFaceName << " " << lf.lfHeight);
  RDUMP((void *)hdc);
  RDUMP((void *)hfont);
  RDUMP(GetLastErrorMessage());
 }
 VERIFY(gsz >= 0);
 if(gsz < 0)
  return;
 RenderCharPath(~gb, gsz, sw, x, y + fnt.GetAscent());
 ::SelectObject(hdc, ohfont);
 ::DeleteDC(hdc);
 }
 h--;
 VERIFY(h == 0);
}
```

As far as I can see, not a single bad or missing letter in 8 minutes of systematic intensive scanning of names. No VERIFYs/ASSERTs either. The log is attached.

BR, Tom

Well, so be it, lets call it a fix. I will remove shared information context from the trunk... There will be small performance penalty, but it should be negligible (as there are caches on many levels anyway).

Hopefully my interpretation is right and we are not just hiding some other bug...

OK, I hope so too. Using Windows can be expensive sometimes...

As for the 'rounding of tolerance' we discussed about one hundred (or so) postings ago, should it be brought in rather sooner or later? I would like to test this current solution without other changes for a while before taking in an optimization that will decrease likelihood for error. (Actually, I believe this will solve it for good, but it's better keep an eye on it for a while... This does not 'add to cost' as I will anyway look at the map during testing of new features I'm building on top of this.)

Of course I can take this fix in now, and then postpone updating the tolerance rounding, until a week or so have passed...

In any case, Thank you Mirek for all your help and support!

Best regards,

Tom