Subject: Re: Strange issue with text in Painter
Posted by mirek on Thu, 17 Jan 2019 12:58:49 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Thu, 17 January 2019 13:40mirek wrote on Thu, 17 January 2019 14:34What about

```
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 = Win32_IC();
  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);
 }
 h--;
 VERIFY(h == 0);
}
```

Again the first VERIFY(gsz >= 0); ASSERTED on line 424. LOG attached...

BR, Tom

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);
}
```