

No luck... Let us add more logging:

```
struct GlyphKey {
    Font fnt;
    int chr;
    double tolerance;

    bool operator==(const GlyphKey& b) const {
        DLOG("*** operator==");
        DDUMP(fnt);
        DDUMP(b.fnt);
        DDUMP(chr);
        DDUMP(b.chr);
        return fnt == b.fnt && chr == b.chr && tolerance == b.tolerance;
    }
    unsigned GetHashValue() const {
        return CombineHash(fnt, chr, tolerance);
    }
};

struct sMakeGlyph : LRUCache<Value, GlyphKey>::Maker {
    GlyphKey gk;

    GlyphKey Key() const { return gk; }
    int Make(Value& v) const {
        GlyphPainter gp;
        gp.move = gp.pos = Null;
        gp.tolerance = gk.tolerance;
        DLOG("*** Make");
        DDUMP(GetHashValue(gk));
        DDUMP(gk.fnt);
        DDUMP(gk.chr);
        PaintCharacter(gp, Pointf(0, 0), gk.chr, gk.fnt);
        int sz = gp.glyph.GetCount() * 4;
        v = RawPickToValue(pick(gp.glyph));
        return sz;
    }
};

void ApproximateChar(LinearPathConsumer& t, Pointf at, int ch, Font fnt, double tolerance)
{
    PAINTER_TIMING("ApproximateChar");
    Value v;
```

```

INTERLOCKED {
    DLOG("==== ApproximateChar " << ch << " " << fnt);
    PAINTER_TIMING("ApproximateChar::Fetch");
    static LRUCache<Value, GlyphKey> cache;
    cache.Shrink(500000);
    sMakeGlyph h;
    h.gk.fnt = fnt;
    h.gk.chr = ch;
    h.gk.tolerance = tolerance;
    DDUMP(GetHashValue(h.gk));
    v = cache.Get(h);
    DDUMP(ValueTo< Vector<float> >(v));
}
#ifdef 0
    GlyphPainter chp;
    chp.move = chp.pos = Null;
    chp.tolerance = tolerance;
    PaintCharacter(chp, Pointf(0, 0), ch, fnt);
    const Vector<float>& g = chp.glyph;
#else
    const Vector<float>& g = ValueTo< Vector<float> >(v);
#endif
    int i = 0;
    while(i < g.GetCount()) {
        Pointf p;
        p.x = g[i++];
        if(p.x > 1e30) {
            p.x = g[i++];
            p.y = g[i++];
            t.Move(p + at);
        }
        else {
            PAINTER_TIMING("ApproximateChar::Line");
            p.y = g[i++];
            t.Line(p + at);
        }
    }
}
}

```

the same (no need to send painting, just .log and tell me what letter is missing / wrong).

Thanks,

Mirek