

This shows the problem:

```
void RenderCharPath2(const char* gbuf, unsigned total_size, FontGlyphConsumer& sw, double
xx, double yy)
```

```
{
    const char* cur_glyph = gbuf;
    const char* end_glyph = gbuf + total_size;
    Pointf pp(xx, yy);
    while(cur_glyph < end_glyph) {
        const TTPOLYGONHEADER* th = (TTPOLYGONHEADER*)cur_glyph;
        const char* end_poly = cur_glyph + th->cb;
        const char* cur_poly = cur_glyph + sizeof(TTPOLYGONHEADER);
        sw.Move(fx_to_dbl(pp, th->pfxStart));
/* while(cur_poly < end_poly) {
    const TTPOLYCURVE* pc = (const TTPOLYCURVE*)cur_poly;
    if (pc->wType == TT_PRIM_LINE)
        for(int i = 0; i < pc->cpfx; i++)
            sw.Line(fx_to_dbl(pp, pc->apfx[i]));
    if (pc->wType == TT_PRIM_QSPLINE)
        for(int u = 0; u < pc->cpfx - 1; u++) {
            Pointf b = fx_to_dbl(pp, pc->apfx[u]);
            Pointf c = fx_to_dbl(pp, pc->apfx[u + 1]);
            if (u < pc->cpfx - 2)
                c = Mid(b, c);
            sw.Quadratic(b, c);
        }
        cur_poly += sizeof(WORD) * 2 + sizeof(POINTFX) * pc->cpfx;
    }
*/ sw.Close();
    cur_glyph += th->cb;
}
}
```

This renders OK:

```
void RenderCharPath2(const char* gbuf, unsigned total_size, FontGlyphConsumer& sw, double
xx, double yy)
```

```
{
    const char* cur_glyph = gbuf;
    const char* end_glyph = gbuf + total_size;
    Pointf pp(xx, yy);
    while(cur_glyph < end_glyph) {
        const TTPOLYGONHEADER* th = (TTPOLYGONHEADER*)cur_glyph;
        const char* end_poly = cur_glyph + th->cb;
        const char* cur_poly = cur_glyph + sizeof(TTPOLYGONHEADER);
```

```

sw.Move(fx_to_dbl(pp, th->pfxStart));
while(cur_poly < end_poly) {
    const TTPOLYCURVE* pc = (const TTPOLYCURVE*)cur_poly;
/*   if (pc->wType == TT_PRIM_LINE)
        for(int i = 0; i < pc->cpfx; i++)
            sw.Line(fx_to_dbl(pp, pc->apfx[i]));
    if (pc->wType == TT_PRIM_QSPLINE)
        for(int u = 0; u < pc->cpfx - 1; u++) {
            Pointf b = fx_to_dbl(pp, pc->apfx[u]);
            Pointf c = fx_to_dbl(pp, pc->apfx[u + 1]);
            if (u < pc->cpfx - 2)
                c = Mid(b, c);
            sw.Quadratic(b, c);
        }
*/   cur_poly += sizeof(WORD) * 2 + sizeof(POINTFX) * pc->cpfx;
}
sw.Close();
cur_glyph += th->cb;
}
}

```

I will now check the next suggestion with raw pick.

BR, Tom
