
Subject: Crash in Painter

Posted by [zsoft](#) on Tue, 29 Jan 2013 17:10:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

One of my beta tester sent me some crash files:

Typical start of them is:

Access violation reading at 0x00000002

0x005be950: class Upp::Image Upp::DownScale(class Upp::Image const &,int,int) + 0x150 bytes

Recognized stack dwords:

0x005c03cd:

??\$Sort@PAUCell@Rasterizer@Upp@@@U?\$StdLess@UCell@Rasterizer@Upp@@@@3@@Up
p@@YAXPAUCell@Rasterizer@0@0ABU?\$StdLess@UCell@Rasterizer@Upp@@@@0@@Z +
0x29d bytes

0x006f95d1: __ehandler\$?DownScale@Upp@@YA?AVImage@1@ABV21@HH@Z + 0x0
bytes

0x005beb80: void Upp::PainterImageSpan::Set(struct Upp::Xform2D const &,class Upp::Image
const &) + 0xf0 bytes

0x007021e8: __ehandler\$?WorkPage@Heap@Upp@@QAEPAUPage@12@H@Z + 0x0 bytes

0x006f95eb:
__ehandler\$?Set@PainterImageSpan@Upp@@QAEAXform2D@2@ABVImage@2@@@Z
+ 0x0 bytes

0x005bb7ee: class Upp::Vector<struct Upp::RGBA> & Upp::operator<=(class
Upp::Vector<struct Upp::RGBA> &,class Upp::Vector<struct Upp::RGBA> const &) + 0x30 bytes

0x006f63de:
__ehandler\$??\$DeepCopyConstruct@UPos@HelpWindow@Upp@@@@Upp@@YAAAUPos@
HelpWindow@0@PAXABU120@@@Z + 0x0 bytes

0x005bb250: struct Upp::Xform2D Upp::operator*(struct Upp::Xform2D const &,struct
Upp::Xform2D const &) + 0x9 bytes

0x005bf2e1: void Upp::BufferPainter::RenderImage(double,class Upp::Image const &,struct
Upp::Xform2D const &,unsigned long) + 0xf1 bytes

I was able to reproduce it, but it is not easy.

It crashes in this code:

```
while(s < e) {  
    for(int n = nx; n--;)  
        t->Put(*s++);  
    t++;  
}
```

The problem is that in this line:

```
t->Put(*s++);
```

s variable points to a not accessible memory address.

Subject: Re: Crash in Painter

Posted by [mirek](#) on Sat, 09 Feb 2013 12:05:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have examined the code, looks OK. I have put a new assert there:

```
Image DownScale(const Image& img, int nx, int ny)
{
    ASSERT(nx > 0 && ny > 0);
    Size ssz = img.GetSize();
    Size tsz = Size((ssz.cx + nx - 1) / nx, (ssz.cy + ny - 1) / ny);
    int div = nx * ny;
    Buffer<RGBAV> b(tsz.cx);
    ImageBuffer ib(tsz);
    RGBA *it = ~ib;
    int scx0 = ssz.cx / nx * nx;
    for(int yy = 0; yy < ssz.cy; yy += ny) {
        for(int i = 0; i < tsz.cx; i++)
            b[i].Clear();
        for(int yi = 0; yi < ny; yi++) {
            int y = yy + yi;
            if(y < ssz.cy) {
                const RGBA *s = img[y];
                const RGBA *e = s + scx0;
                const RGBA *e2 = s + ssz.cx;
                RGBAV *t = ~b;
                while(s < e) {
                    for(int n = nx; n--;)
                        t->Put(*s++);
                    t++;
                }
                while(s < e2)
                    t->Put(*s++);
                ASSERT(s <= ~img + img.GetLength());
            }
        }
        const RGBAV *s = ~b;
        for(int x = 0; x < tsz.cx; x++)
            *it++ = (s++)->Get(div);
    }
    return ib;
}
```

and played for a while with PainterExamples, still OK.

-> I am afraid that I will need more detailed information. E.g. DDUMPs of ssz, tsz, div, it and scx0.

Is not it possible that Image was somewhat damaged?

Mirek

Subject: Re: Crash in Painter

Posted by [zsolt](#) on Mon, 11 Feb 2013 01:09:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks for your efforts. I will try creating a reproducible testcase soon.
