

Well, I have spent last 4 hours with this problem and I really do not know what to think....

See this:

```
#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;
```

```
class Testcase2 : public TopWindow{
```

```
public:
```

```
typedef Testcase2 CLASSNAME;
```

```
Testcase2(){  
    Sizeable();  
    MinimizeBox();  
    MaximizeBox();  
    BackPaint();  
}
```

```
Image MakeImage2(Size sz) const {  
    ImageBuffer ib(sz.cx / 3, sz.cy / 3);  
    Fill(ib, LtBlue(), ib.GetLength());  
    const Color c[] = { LtBlue() };  
    for(int y = 0; y < ib.GetWidth(); y++) {  
        for(int x = 0; x < ib.GetHeight(); x++)  
            ib[x][y] = c[2 * (x > ib.GetHeight() / 2) + (y > ib.GetWidth() / 2)];  
    }  
    return ib;  
}
```

```
Image MakeImage(Size sz) const {  
    ImageDraw iw(sz / 3);  
    iw.DrawRect(sz / 3, Blue);  
    iw.DrawText(0, 0, "O", Roman(sz.cy / 3).Italic(), Red());  
    return iw;  
}
```

```
Image MakeImage1(Size sz) const {  
    ImageBuffer ib(sz);  
    for(int y=0;y<sz.cy;y++){  
        for(int x=0;x<sz.cx;x++){  
            ib[y][x]=(RGBA)Color((16*y/sz.cy)<<4,(16*y/sz.cy)<<4,(16*x/sz.cx)<<4);  
        }  
    }  
}
```

```

}
}
return ib;
}

```

```

void print(){
}

```

```

virtual void Paint(Draw &w){
    Size sz = GetSize();
    w.DrawRect(GetSize(), White);
    w.DrawImage(0, 0, MakeImage(GetSize()));
    w.DrawText(10,500,"Press 'P' to print image, 'R' to print rectangle test, 'C' to another test...");
    w.DrawText(10,530,"(This really drains your ink/toner cartridge, so try e.g. Microsoft XPS Document Writer.)");
}

```

```

bool Key(dword key, int count){
    PrinterJob job;
    switch(key){
    case K_P:
        if(job.Execute()){
            Draw &draw = job;
            draw.StartPage();
            draw.DrawImage(0, 0, MakeImage(draw.GetPagePixels()));
            draw.EndPage();
        }
        return true;
    case K_R:
        if(job.Execute()){
            Draw &draw = job;
            draw.StartPage();
            bool flag = false;
            for(int y = 0; y < 100; y++)
                for(int x = 0; x < 100; x++) {
                    draw.DrawRect(16 * x, 16 * y, 16, 16, flag ? Blue : Red);
                    if((Random() & 31) == 0)
                        flag = !flag;
                }
            draw.EndPage();
        }
        return true;
    case K_C:
        if(job.Execute()){
            Draw &draw = job;
            draw.StartPage();
            for(int y = 0; y < 100; y++)
                for(int x = 0; x < 100; x++)

```

```
        draw.DrawRect(16 * x, 16 * y, 16, 16, Blue);
        draw.EndPage();
    }
    return true;
}
return false;
}

};

GUI_APP_MAIN
{
    Testcase2().Run();
}
```

Now interesting things happen when you press R or C.

These try to emulate the compression issue. When you do R, you have those line artifacts, but C seems without the issue - and only thing that has changed is the color.

Is not it really weird? (Note that it is not even related to Image).

It can also be seen with P... The areas that are "full line" are OK.

Well, I have also tried to mitigate the issue by "overpainting" rectangle (adding to height/width), but interestingly that seems to have produced other issues.

OK, enough for today, next tomorrow. We really need this working BTW.

Mirek
