
Subject: Re: How to efficiently update a large Image?

Posted by [Tom1](#) on Tue, 01 Sep 2009 06:24:53 GMT

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

Mirek,

Here's the testcase. Only after running your test code I realized the problem was related to having a control inside the main window. So, I guess this has something to do with the control offset within the parent control.

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

```
using namespace Upp;
```

```
class PainterCtrl : public Ctrl {
    ImageBuffer ib;
```

```
public:
```

```
    virtual void Paint(Painter &pnr, Size &sz)=0;
```

```
    virtual void Layout(){
        Size sz=GetSize();
        ib.Clear();
        ib.Create(sz);
        BufferPainter pnr(ib);
        Paint(pnr,sz);
    }
```

```
    virtual void Paint(Draw &draw) {
        Rect paintrect=draw.GetPaintRect();
```

```
        Point p = paintrect.TopLeft();
        if(p.x < ib.GetSize().cx && p.y < ib.GetSize().cy)
            SetSurface(draw, paintrect, ib, ib.GetSize(), p);
    }
```

```
};
```

```
struct ExampleCtrl : PainterCtrl {
```

```
    virtual void Paint(Painter &pnr, Size &sz){
        pnr.Move(0,0).Line(sz.cx,0).Line(sz.cx,sz.cy).Line(0,sz.cy).Close().Fill(White()).Stroke(5,Black())
        ;
        pnr.Move(0,0).Line(sz.cx,sz.cy).Stroke(5,Black());
    }
};
```

```

class ExampleTopWindow: public TopWindow{
public:
    ExampleCtrl ec;

    virtual void Layout(){
        ec.SetRect(20,0,GetSize().cx-20,GetSize().cy);
    }

    ExampleTopWindow(){
        Add(ec);
    }
};

GUI_APP_MAIN
{
    ExampleTopWindow win;
    win.Sizeable();
    win.Open();
    win.Run();
}

```

(This example also sort of demonstrates how to boost application performance when drawing complex items with e.g. Painter and then not having to regenerate them each time the window gets an OS initiated paint request.)

// Tom
