

Hi,

Maybe something like this could do the trick:

```
template <class T>
class BlurringDecorator : public T {

private:
    bool doBlur;

    void blurImage(Image& m) {
        m = Sharpen(m, -200);
        m = Sharpen(m, -200);
        m = Sharpen(m, -200);
        m = Sharpen(m, -200);
        m = Sharpen(m, -200);
        m = Sharpen(m, -200);
    }

public:

    BlurringDecorator() : doBlur(true) {}
    virtual ~BlurringDecorator() {}
    void setBlur(bool blur = true) { doBlur = blur; }

    virtual void Paint(Draw& w) {

        if (doBlur) {
            ImageDraw dw(T::GetSize());
            T::Paint(dw);
            Image im = dw;
            blurImage(im);
            w.DrawImage(0, 0, im);
        }
        else {
            T::Paint(w);
        }
    }

};
```

In layout

```
BlurringDecorator<CtrlToBlur> myCtrl;
```

EDIT: just tried, works fine (but only blurrs the current Ctrl ... not what is under it !!!)

---