
Subject: Layout call procedure buggy for invisible controls?

Posted by [kohait00](#) on Wed, 29 Sep 2010 15:41:45 GMT

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

hi all,

i have a hierarchy of Ctrl's which is pretty huge...500+ controls of different types...arbitrary grouped, tree depth is up to 10.. a huge one...

i noticed that when i resize the container, which has them all, is it ofcourse is slowly relayouting, which is understandable.

BUT: when the container, beeing part of ExpanderCtrl, is collapsed, and made invisible, and i still resize the Splitter it is part of, the whole relayouting is still slow as before, though nothing is visible..

i encountered, that SyncLayout is called fully recursively on the whole tree, even if it is invisible..

is there any way to optimize that? or am i doing sth wrong in terms of using the api of upp ctrls?

just a remarker where it is at:

CtrlPos.cpp:162

```
void Ctrl::SyncLayout(int force)
{
    GuiLock __;
    LLOG("SyncLayout " << Name() << " size: " << GetSize());
    bool refresh = false;
    Rect oview = GetView();
    Rect view = GetRect().Size();
    overpaint = OverPaint();
    for(int i = 0; i < frame.GetCount(); i++) {
        Frame& f = frame[i];
        f.frame->FrameLayout(view);
        if(view != f.view) {
            f.view = view;
            refresh = true;
        }
        int q = f.frame->OverPaint();
        if(q > overpaint) overpaint = q;
    }
    if(oview.Size() != view.Size() || force > 1) {
        for(Ctrl *q = GetFirstChild(); q; q = q->next) {
            q->rect = q->CalcRect(rect, view);
            LLOG("Layout set rect " << q->Name() << " " << q->rect);
            q->SyncLayout(force > 1 ? force : 0);
        }
    }
}
```

```
Refresh();  
}  
if(oview != view || force) {  
    State(LAYOUTPOS);  
    Layout();  
}  
if(refresh)  
    RefreshFrame();  
}
```
