
Subject: Re: List of custom controls, how?
Posted by [dolik.rce](#) on Wed, 07 Nov 2012 19:25:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

crydev wrote on Wed, 07 November 2012 10:02Ok I found a ColumnList for Ctrl's. I tested it with standard controls and it works! However, when I try to add my own controls it draws a scrollbar, but it doesn't draw any of my controls.

- Why does it not draw itself? I tried to inherit from Ctrl and ParentCtrl.
- The thread in which the CtrlColumnList was posted also has posts stating that the idea to implement this functionality into the existing ColumnList or ArrayCtrl could be worked out. I have found functions like Add(Ctrl&/* ctrl) in these controls so can I assume the extra code for the CtrlColumnList would not be necessary anymore?

It is very hard to say why it doesn't work without seeing the code. Also, I'm not very familiar with ColumnList as I don't like some aspects of its behavior

But for what you need ColumnList or ArrayCtrl are probably overkill anyway. It can be achieved relatively simply by using a ScrollBar. Here is some code for your inspiration:#include <CtrlLib/CtrlLib.h>

```
using namespace Upp;
```

```
// arbitrary custom Ctrl
struct MyCtrl : public ParentCtrl{
    Label l;
    MyCtrl(){
        Add(l.HSizePos(10,10).VSizePos(5,5));
        l.SetFrame(InsetFrame());
    }
    virtual void SetData(const Value& data){
        l.SetText(AsString(data));
    }
};
```

```
struct App : public TopWindow {
    typedef App CLASSNAME;
    Array<MyCtrl> ctrls;
    ScrollBar sb;
    int height;
    App() {
        //initialize
        Title("Test").Sizeable();
        SetRect(0, 0, 120, 400);
        height = 40;

        //populate
        for(int i = 0; i < 12; i++){
            ctrls.Add() <<= "Widget #" + IntStr(i);
```

```

    Add(ctrls[i].HSizePos().TopPos(height*i, height));
}

//set up scrollbar
sb.Vert().AutoHide().SetTotal(ctrls.GetCount() * height);
sb.WhenScroll = THISBACK(Scroll);
AddFrame(sb);
}
void Scroll() {
    //update position
    int pos = sb.Get();
    for(int i = 0; i < ctrls.GetCount(); i++){
        ctrls[i].TopPos(height*i - pos, height);
    }
}

virtual void Layout() {
    //window was resized, adjust scrollbar
    sb.SetPage(GetSize().cy);
}

virtual bool Key(dword key, int) {
    // pass key event to scrollbar
    return sb.VertKey(key);
}
};

GUI_APP_MAIN{
    App().Run();
};

```

You will probably want to implement something similar to this but based on ParentCtrl instead of TopWindow, so you can position it inside the rest of your app (I wrote as a selfstanding app just so it is easy to compile and test...). The whole idea is to have a ScrollBar and whenever it is moved, you just iterate over your Ctrls and adjust their position. Any Ctrls that are outside of their parent are not rendered, so in the end, it will result in nice smooth scrolling effect. It is really simple

Honza
