
Subject: Re: GridCtrl + rich texts

Posted by [Sender Ghost](#) on Tue, 07 Aug 2012 00:35:43 GMT

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

I thought about generalized way to access Display through GridDisplay and made following example with templates:

Toggle Spoiler

```
#include <GridCtrl/GridCtrl.h>
#include <RichText/RichText.h>
```

```
using namespace Upp;
```

```
template <const Display& (*DisplayFn)()>
class DisplayToGrid : public GridDisplay {
protected:
    const Display& display;
public:
    DisplayToGrid() : display(DisplayFn()) {}
```

```
virtual void Paint(Draw& w, int x, int y, int cx, int cy, const Value& val, dword style,
    Color& fg, Color& bg, Font& fnt, bool found, int fs, int fe)
{
    display.Paint(w, RectC(x, y, cx, cy), val, fg, bg, style);
}
};
```

```
template <const Display& (*DisplayFn)()>
GridDisplay& SingleDisplayToGrid()
{
    return Single<DisplayToGrid<DisplayFn> >();
}
```

```
class DisplayGrid : public GridDisplay {
protected:
    const Display *display;
public:
    DisplayGrid() : display(NULL) {}
    DisplayGrid(const Display& d) : display(&d) {}
    void SetDisplay(const Display& d) { display = &d; }
    const Display& GetDisplay() { return *display; }
```

```
virtual void Paint(Draw& w, int x, int y, int cx, int cy, const Value& val, dword style,
    Color& fg, Color& bg, Font& fnt, bool found, int fs, int fe)
{
    if (display)
```

```

    display->Paint(w, RectC(x, y, cx, cy), val, fg, bg, style);
}
};

class CustomDisplayWithIcon : public DisplayWithIcon {
public:
    virtual void Paint(Draw& w, const Rect& r, const Value& q, Color ink, Color paper, dword style)
const
    {
        w.DrawRect(r, paper);
        DisplayWithIcon::Paint(w, r, q, ink, paper, style);
    }
};

typedef GridDisplay& (*SingleDisplay)();
Tuple3<String, String, SingleDisplay> display[] = {
    "StdDisplay", "Left aligned text", SingleDisplayToGrid<StdDisplay>,
    "StdCenterDisplay", "Center aligned text", SingleDisplayToGrid<StdCenterDisplay>,
    "StdRightDisplay", "Right aligned text", SingleDisplayToGrid<StdRightDisplay>,
    "QTFDisplay", "[G1= [* bold], [/ italic], [- strikeout], [_ underline], [-*/ all previous]]",
    SingleDisplayToGrid<QTFDisplay>
};

const int displayCount = __countof(display);

class App : public TopWindow {
public:
    typedef App CLASSNAME;
    App();

    GridCtrl list;
};

App::App()
{
    Title("GridCtrl Display Example");
    Sizeable().Zoomable();
    const Size sz(640, 320);
    SetRect(sz); SetMinSize(sz);

    list.Chameleon();
    list.AddColumn("Display").Fixed(120);
    list.AddColumn("Example");

    for (int i = 0; i < displayCount; ++i) {
        list.Add(display[i].a, display[i].b);
        list.GetCell(i, 1).SetDisplay(display[i].c());
    }
}

```

```

int count = displayCount;
list.Add("ColorDisplay", Blend(LtGreen(), Magenta())).GetCell(count++,
1).SetDisplay(SingleDisplayToGrid<ColorDisplay>());
list.Add("ImageDisplay", CtrlImg::checkers()).GetCell(count++,
1).SetDisplay(SingleDisplayToGrid<ImageDisplay>());
list.Add("FittedImageDisplay", CtrlImg::checkers()).GetCell(count++,
1).SetDisplay(SingleDisplayToGrid<FittedImageDisplay>());
// And complex example about how to integrate GridDisplay with Display, while using their custom
functions.
CustomDisplayWithIcon& dwi = Single<CustomDisplayWithIcon>();
dwi.SetIcon(CtrlImg::go_forward());
dwi.SetDisplay(StdCenterDisplay());
DisplayGrid& dg = Single<DisplayGrid>();
dg.SetDisplay(dwi);
list.Add("CustomDisplayWithIcon", "Center aligned text with icon").GetCell(count,
1).SetDisplay(dg);

Add(list.VSizePosZ(4, 4).HSizePosZ(4, 4));
}

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

```

Edit: Replaced DisplayWithIcon with CustomDisplayWithIcon class, which draws background rectangle to fix icon drawing issue (and possibly other Display without background rectangle) on GridCtrl cell. Optimized DisplayToGrid initialization.

File Attachments

1) [DisplayGridExample.png](#), downloaded 958 times
