

Except it doesn't work at all!

The problems:

- EditField doesn't have StdBar, are you thinking of EditText?
- As above for WhenMenu, just overload RightDown and call EditField::DoMenu
- I've also changed it to use ArrayCtrl::StdBar and set the ArrayCtrl cursor correctly to enable this, which is more in line with the way the ArrayCtrl works.

Here's my version:

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;
```

```
struct EditStringSpecial : EditString {
    ArrayCtrl *GetArrayCtrl() {
        for(Ctrl *q = GetParent(); q; q = q->GetParent())
            if(ArrayCtrl *a = dynamic_cast<ArrayCtrl *>(q))
                return a;
        return NULL;
    }

    static void DoMenu(EditStringSpecial *x) {
        class MenuBar bar;
        x->MenuBar(bar);
        ArrayCtrl *a = x->GetArrayCtrl();
        if(a) {
            a->SetCursor(a->GetLineAt(GetMousePos().y - a->GetScreenView().TopLeft().y));
            bar.Separator();
            a->StdBar(bar);
        }
        bar.Execute();
    }

    virtual void RightDown(Point p, dword keyflags) {
        PostCallback(callback1(DoMenu, this));
    }

    typedef EditStringSpecial CLASSNAME;

    EditStringSpecial() {
        ClearFrames();
    }
};

GUI_APP_MAIN
```

```
{
    ArrayCtrl a;
    a.AddColumn("Label");
    a.AddColumn("Text").Ctrls<EditStringSpecial>();
    for(int i = 0; i < 300; i++)
        a.Add("Label " + AsString(i), AsString(i));
    a.SetLineCy(Draw::GetStdFontCy() + 8);

    a.Duplicating(true).Removing(true).AppendLine(true).Appending(true).Inserting(true);

    TopWindow app;
    app.Add(a.SizePos());
    app.Sizeable();
    app.Run();
}
```

Otherwise it's very nice, though it would be good to avoid the `dynamic_cast`.

One remaining problem is that if `ArrayCtrl` doesn't add anything to the menu then you end up with a hanging separator, but I don't what to do about that. It would be nice if `MenuBar` ignored these automatically.
