
Subject: Re: GridCtrl and diff Edit()'s per row, same column

Posted by [mrjt](#) on Thu, 04 Jun 2009 14:11:17 GMT

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

I don't use GridCtrl, but this is the technique for ArrayCtrl (deopending on your needs GridCtrl might be overkill anyway):

```
#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;
```

```
class CtrlLibTest : public TopWindow {
    struct Question : public Moveable<Question> {
        void Set(String question, int _type, Value _default_value = Value()) {
            type = _type;
            text = question;
            default_value = _default_value;
        }
        dword type;
        String text;
        Value default_value;
    };
private:
```

```
    ArrayCtrl list;
    Vector<Question> questions;
    Array<Ctrl> ctrls;
    Button finish;
    Button close;
public:
```

```
    typedef CtrlLibTest CLASSNAME;
    CtrlLibTest();

    void SetQuestions();
    void FillList();
    void GetAnswers();
    String FormatAnswer(int type, const Value &v);
};
```

```
CtrlLibTest::CtrlLibTest()
```

```
{
    Title("Question/Answer program");
    SetRect(Size(300,300));
```

```
    list.ColumnWidths("186 93").HSizePosZ().VSizePosZ(0, 32);
    finish.Ok().SetLabel("Finish").LeftPosZ(4, 64).BottomPosZ(4, 24) <=< THISBACK(GetAnswers);
    close.Cancel().SetLabel("Cancel").RightPosZ(4, 64).BottomPosZ(4, 24) <=<
    Rejector(IDCANCEL);
    *this << list << finish << close;
```

```
list.AddIndex(); // Question id
list.AddColumn("Question");
list.AddColumn("Answer");
```

```
SetQuestions();
FillList();
}
```

```
void CtrlLibTest::SetQuestions()
{
    questions.Add().Set("Feeder: Filled With", STRING_V);
    questions.Add().Set("Feeder: Refilled", BOOL_V);
    questions.Add().Set("Honey Super: Frames Filled", INT_V);
    questions.Add().Set("Honey Super: Harvested", INT_V);
}
```

```
void CtrlLibTest::FillList()
{
    ctrls.Clear();
    for (int i = 0; i < questions.GetCount(); i++) {
        list.Add(i, questions[i].text, questions[i].default_value);
        switch (questions[i].type) {
            case INT_V:
                ctrls.Create<EditInt>();
                break;
            case STRING_V:
                ctrls.Create<EditString>();
                break;
            case BOOL_V:
                ctrls.Create<Option>();
                break;
            case DOUBLE_V:
                ctrls.Create<EditDouble>();
                break;
            default:
                NEVER();
                continue;
        }
        list.SetCtrl(i, 1, ctrls.Top());
    }
}
```

```
void CtrlLibTest::GetAnswers()
{
    if (!Accept()) return;
    // Because ArrayCtrl doesn't Accept properly:
    for (int i = 0; i < ctrls.GetCount(); i++)
        if (!ctrls[i].Accept()) return;
```

```

String qtf("Here are your answers:&");
for (int i = 0; i < list.GetCount(); i++)
    qtf << list.Get(i, 1) << "- "
    << FormatAnswer(questions[(int)list.Get(i, 0)].type, list.Get(i, 2)) << "&";
PromptOK(qtf);
}

```

```

String CtrlLibTest::FormatAnswer(int type, const Value &v)
{
    if (type == BOOL_V)
        return ((bool)v) ? "Yes" : "No";
    return StdConvert().Format(v);
}

```

```

GUI_APP_MAIN
{
    CtrlLibTest().Execute();
}

```
