Subject: ArrayCtrl: Edit doesn't like Option
Posted by cbpporter on Mon, 25 Oct 2010 15:15:05 GMT
View Forum Message <> Reply to Message

Let me start by saying that this is the first time I used ArrayCtrl. I tried it before, but it seemed overkill and I have been using ColumnList up until now and it served my needs quite well.

But now I need it and I'm having a problem with it. I have some columns with Edit(someEditor). Like EditString. Works fine. But when adding Ctrls<Option> to some other columns, Edit no longer works for the previous ones. Probably related to the way the embedded Option gets focused. It had huge troubles with Option and OptionTree and was wondering if there is a solution for this without having to go through all that trouble again.

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by cbpporter on Tue, 26 Oct 2010 08:03:48 GMT
View Forum Message <> Reply to Message

Problem solved!

Thank you very much, SenderGhost!

The CreateOption method is key from this post: link

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by kohait00 on Tue, 07 Dec 2010 08:37:06 GMT
View Forum Message <> Reply to Message

i've got another problem, which is similar to yours, so i just reuse your thread

problem: ArrayCtrl has got Option's as Editors, and some EditInt's.. everytime changes are made, i need to forward them in a custim action..(OnAction)

```
ArrayCtrl ac;
...
Ctrl* p = new Option();
p->WhenAction = THISBACK(OnAction);
SetCtrl(k, 1, p);
++k;
...
p = new EditInt();
p->WhenAction = THISBACK(OnAction);
ac.SetCtrl(k, 1, p);
...
```

```
void MyClass::OnAction()
{
   DoWork(ac.Get(0), ac.Get(1));
}
```

this works fine for EditInt, because the cursor is set properly, when EditInt is 'selected' to be edited, because EditField has SetFocus somewhere in it.. with Option, no cursor is set, when i cklick it, so the OnAction fails with an ASSERT.

when i manually select the row, clicking on the right remaining spot, that belongs to arrayctrl, and then, on the Option, it works..

any help, how to make Option be Focus aware? i dont like to mess with the upp code.. maybe i'm doing sth wrong ..

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by unodgs on Tue, 07 Dec 2010 10:08:28 GMT
View Forum Message <> Reply to Message

You could use GridCtrl it moves cursor always. However if you prefer ArrayCtrl just make your own control that derives from ArrayCtrl and override ChildGotFocus() method. In this method place the cursor in the correct location (row). This method is called before control's action callback so everything should work.

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by kohait00 on Tue, 07 Dec 2010 10:22:16 GMT
View Forum Message <> Reply to Message

as far as i could see, GetChilfFocus is used in ArrayCtrl as well, the problem is, that Option does not aquire focus , there is no SetFocus() in Pusher..

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by unodgs on Tue, 07 Dec 2010 11:37:04 GMT
View Forum Message <> Reply to Message

Sorry. I made a mistake I meant overriding void ChildMouseEvent(Ctrl *child, int event, Point p, int zdelta, dword keyflags). I use this method in GridCtrl to move cursor if embedded control is clicked and it works perfectly.

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by kohait00 on Tue, 07 Dec 2010 11:39:54 GMT
View Forum Message <> Reply to Message

thanks a lot. i'll give it a try..

wouldn't it be better to also implement this for ArrayCtrl?

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by unodgs on Tue, 07 Dec 2010 12:14:25 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Tue, 07 December 2010 06:39
wouldn't it be better to also implement this for ArrayCtrl?
Well, moving cursor is IMO the correct way to deal with embedded controls and most of grids works this way otherwise you break consistency in control's callbacks. On the other hand Mirek has different opinion here. If I remember correctly he wanted to be able to type the text in one row without loosing focus by clicking option/button/droplist in another row or the same row but in another column. Cursor is moved only if you click on edit control. Actually I don't know in which scenario this can be useful .

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by kohait00 on Tue, 07 Dec 2010 13:03:08 GMT
View Forum Message <> Reply to Message

i think the same..
clicking on a Option switch means changing a data set in a row, which then needs to be updated to database, i.e..
so maybe he still can be convinced of it.

meanwhile, i might try the GridCtrl, if it offers the same possibilities..

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by kohait00 on Wed, 08 Dec 2010 08:51:56 GMT
View Forum Message <> Reply to Message

do you have any code already as a fix for ArrayCtrl?
i mean a ChildMouseEvent(Ctrl *child, int event, Point p, int zdelta, dword keyflags) based one, just to have a base for mirek to talk about

EDIT:
ok, i stiched an add to ArrayCtrl, in a similar manner like you did in GridCtrl, now it's up on Mirek to decide if this goes well..thankyou for yor support

ArrayCtrl.h:38

```
 virtual void ChildMouseEvent(Ctrl *child, int event, Point p, int zdelta, dword keyflags);
```

ArrayCtrl.cpp:713

```
void ArrayCtrl::ChildMouseEvent(Ctrl *child, int event, Point p, int zdelta, dword keyflags)
{
 if(child != GetFocusCtrl())
 {
  if(event == LEFTDOWN || event == RIGHTDOWN || event == MOUSEWHEEL)
  {
   Point p = FindCellCtrl(child);
   if(!IsNull(p))
   {
    if(nocursor)
     ScrollInto(p.y);
    else
     SetCursor(p.y);
   }
  }
 }
 Ctrl::ChildMouseEvent(child, event, p, zdelta, keyflags);
}
```

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by kohait00 on Thu, 09 Dec 2010 08:31:27 GMT

View Forum Message <> Reply to Message

there is a

```
Pusher::ClickFocus();
```

function, that enables Option and others to have SetFocus functionality, so ArrayCtrl changes Cursor right.

but the general question as stated above remains..

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by mirek on Sun, 26 Dec 2010 00:12:58 GMT

View Forum Message <> Reply to Message

kohait00 wrote on Thu, 09 December 2010 03:31there is a

Pusher::ClickFocus();

function, that enables Option and others to have SetFocus functionality, so ArrayCtrl changes Cursor right.

but the general question as stated above remains..

IMO, you might not want to move focus in ArrayCtrl for the same reason you might not want to move it in general dialog. (Well, as unodgs stated, it always seemed illogical to me. I really dislike those focus rectangles on buttons, options and switches

Also, note there is even a global setting, call Ctrl::ClickFocus(true) and you are done...

Mirek

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by kohait00 on Sun, 26 Dec 2010 08:46:31 GMT

View Forum Message <> Reply to Message

thanks, thats a help, bringing together both worlds

Subject: Re: ArrayCtrl: Edit doesn't like Option
Posted by Sender Ghost on Thu, 22 Dec 2011 01:43:56 GMT

View Forum Message <> Reply to Message

Hello.

While reading uppdev sources, I found ArrayOption example.

I think, it could be used here, like follows:
Toggle Spoiler

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

using namespace Upp;

class App : public TopWindow {
private:
 int toggle_count;
 LabelBase *toggle_label;
public:
 typedef App CLASSNAME;
 App();
 // Ctrls
 ArrayCtrl list;
 ArrayOption option;
 EditString text;
 // Bars
 void ArrayBar(Bar& bar);
 // Methods
 void FillWithData();
 void ToggleCount(bool value);
 void ToggleOption();
 void ToggleHeaderOption();
 // Events
 void OnHeaderClick();
 void OnOptionClick();
};

App::App() : toggle_count(0)
{
 Title("ArrayOption example");
 CenterScreen().Sizeable().MinimizeBox().MaximizeBox();
 const Size sz(320, 240);
 SetMinSize(sz); SetRect(sz);

 list.AutoHideSb();
 // Initialize ArrayOption column and assign label address
 toggle_label = &option.AddColumn(list).HeaderTab().Fixed(30).SetAlign(ALIGN_CENTER);
 // Initialize ArrayCtrl column
 list.AddColumn("Content", 120).Edit(text);
 // Assign callbacks
 list.HeaderTab(0).WhenAction = THISBACK(OnHeaderClick);
 list.WhenBar = THISBACK(ArrayBar);
 option.WhenAction = THISBACK(OnOptionClick);

 Add(list.VSizePosZ(4, 4).HSizePosZ(4, 4));
}
// Bars
void App::ArrayBar(Bar& bar)
{
 if (list.HasFocus() && list.IsCursor())
  bar.AddKey(K_SPACE, THISBACK(ToggleOption));

 list.StdBar(bar);
}
// Methods
void App::FillWithData()
{
 GuiLock __;
 list.Clear();

 for (int i = 1; i <= 10; ++i)
  list.Add(i % 2 == 0, FormatIntRoman(i, true));

 toggle_count = 5;
 ToggleHeaderOption();
}

void App::ToggleCount(bool value)
{
 if (value)
  ++toggle_count;
 else
  --toggle_count;
}

void App::ToggleOption()
{
 GuiLock __;

 list.Set(0, !list.Get(0));
 ToggleCount(list.Get(0));
 ToggleHeaderOption();
}

void App::ToggleHeaderOption()
{
 static bool changed = true;

 if (toggle_count == list.GetCount()) {
  toggle_label->SetImage(CtrlsImg::Get(CtrlsImg::I_O1), 2);
  changed = true;
 }
 else if (changed) {
  toggle_label->SetImage(CtrlsImg::Get(CtrlsImg::I_O0), 2);
  changed = false;
 }
}
// Events
void App::OnHeaderClick()
{
 GuiLock __;

 bool value = true;
 const int count = list.GetCount();
 if (toggle_count == count) {
  value = false;
  toggle_count = 0;
 }
 else
  toggle_count = count;

 for (int i = 0; i < count; ++i)
  if (list.Get(i, 0) != value)
   list.Set(i, 0, value);

 ToggleHeaderOption();
}

void App::OnOptionClick()
{
 ToggleCount(list.Get(0));
 ToggleHeaderOption();

 PromptOK(String().Cat() << "{{1:2:3@L [* Index:: Option:: Content]::! "
  << list.GetCursor() << ":: " <<  FormatBool(list.Get(0)) << ":: " << list.Get(1) << " }}");
}

GUI_APP_MAIN
{
 Ctrl::GlobalBackPaint();

 App app;
 app.FillWithData();
 app.Run();
}
```

File Attachments

1) ArrayOptionExample.png, downloaded 856 times