Subject: Embed editfields in ArrayCtrl
Posted by mdelfede on Thu, 07 Jan 2010 23:30:51 GMT

View Forum Message <> Reply to Message

Embedding Editfields in ArrayCtrl offers nice editing and filtering capabilities, and the function

```
Ctrl<EditField>()
```

offers a fast way to do it but.... it has also a couple of caveats :

1 - Embedded fields get frames, which is ugly inside the array
2 - Embedded fields grab ArrayCtrl rightclick context menu

This simple template solves both :

```
template<class E> class Embedded : public E
{
 protected:

  void RightDown(Point p, dword keyflags)
  {
   Ctrl *c = this;
   ArrayCtrl *a;
   while( (c = c->GetParent()) != NULL && ((a = dynamic_cast<ArrayCtrl *>(c)) == NULL))
    ;
   if(a)
   {
    Rect cRect = E::GetScreenRect();
    Rect aRect = a->GetScreenRect();
    p.y += cRect.top - aRect.top - a->HeaderObject().GetHeight();
    p.x += cRect.left - aRect.left;
    a->RightDown(p, keyflags);
   }
  }

 public:
  typedef Embedded<E> CLASSNAME;

  Embedded<E>() { E::ClearFrames(); }
};
```

It's usage is simple, just define your ctrl as, for example :

```
typedef Embedded<EditDouble> EditDoubleEmbedded;
```

Then use as a normal editfield in arrayctrl.
The field will get no frames and forward context menu clicks to its ArrayCtrl container.

Ciao

Max

Subject: Re: Embed editfields in ArrayCtrl
Posted by mirek on Fri, 08 Jan 2010 12:45:00 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Thu, 07 January 2010 18:30Embedding Editfields in ArrayCtrl offers nice editing and filtering capabilities, and the function

Ctrl<EditField>()

offers a fast way to do it but.... it has also a couple of caveats :

1 - Embedded fields get frames, which is ugly inside the array
2 - Embedded fields grab ArrayCtrl rightclick context menu

This simple template solves both :

```
template<class E> class Embedded : public E
{
 protected:

  void RightDown(Point p, dword keyflags)
  {
   Ctrl *c = this;
   ArrayCtrl *a;
   while( (c = c->GetParent()) != NULL && ((a = dynamic_cast<ArrayCtrl *>(c)) == NULL))
    ;
   if(a)
   {
    Rect cRect = E::GetScreenRect();
    Rect aRect = a->GetScreenRect();
    p.y += cRect.top - aRect.top - a->HeaderObject().GetHeight();
    p.x += cRect.left - aRect.left;
    a->RightDown(p, keyflags);
   }
  }

 public:
  typedef Embedded<E> CLASSNAME;

  Embedded<E>() { E::ClearFrames(); }
};
```

It's usage is simple, just define your ctrl as, for example :

typedef Embedded<EditDouble> EditDoubleEmbedded;

Then use as a normal editfield in arrayctrl.
The field will get no frames and forward context menu clicks to its ArrayCtrl container.

Ciao

Max

Actually, as long as you override RightDown, there is no need to make the menu indirectly. Remember, the problem we had was because there was a call to SetFocus after the menu handling...

No RightDown, no SetFocus...

Mirek

Subject: Re: Embed editfields in ArrayCtrl
Posted by mirek on Fri, 08 Jan 2010 12:57:02 GMT

View Forum Message <> Reply to Message

This seems to work quite nicely:

```
#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 DoRemove(ArrayCtrl *a) {
  a->DoRemove();
 }

 static void DoMenu(EditStringSpecial *x) {
  MenuBar bar;
  x->StdBar(bar);
  ArrayCtrl *a = x->GetArrayCtrl();
  if(a) {
   bar.Separator();
   bar.Add("Append a line", callback(a, &ArrayCtrl::DoAppend));
   bar.Add(a->IsCursor(), "Delete a line", callback1(DoRemove, a));
  }
  bar.Execute();
 }

 void MyBar(Bar& bar)
 {
  PostCallback(callback1(DoMenu, this));
 }

 typedef EditStringSpecial CLASSNAME;

 EditStringSpecial() {
  WhenBar = THISBACK(MyBar);
 }
};

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

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

Mirek

Subject: Re: Embed editfields in ArrayCtrl
Posted by mrjt on Fri, 08 Jan 2010 13:49:14 GMT
View Forum Message <> Reply to Message

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.

Subject: Re: Embed editfields in ArrayCtrl
Posted by mirek on Fri, 08 Jan 2010 14:28:17 GMT

View Forum Message <> Reply to Message

mrjt wrote on Fri, 08 January 2010 08:49Except it doesn't work at all!

The problems:
- EditField doesn't have StdBar, are you thinking of EditText?

May I suggest you update your U++ from time to time? (was added 2-3 months back).

Quote:
- As above for WhenMenu, just overload RightDown and call EditField::DoMenu

Means EditField::RightDown does not get called...

Quote:
- 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.

That one is a good idea. I had these left because of experimenting with another solution. So:

```
#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) {
  MenuBar bar;
  x->StdBar(bar);
  ArrayCtrl *a = x->GetArrayCtrl();
  if(a) {
   bar.Separator();
   a->WhenBar(bar);
  }
  bar.Execute();
 }

 void MyBar(Bar& bar)
 {
  PostCallback(callback1(DoMenu, this));
 }

 typedef EditStringSpecial CLASSNAME;

 EditStringSpecial() {
  WhenBar = THISBACK(MyBar);
 }
};

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

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

Mirek

Subject: Re: Embed editfields in ArrayCtrl
Posted by mirek on Fri, 08 Jan 2010 14:33:40 GMT

View Forum Message <> Reply to Message

mrjt wrote on Fri, 08 January 2010 08:49
- 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.

Not really. You had to do that exactly because you have not got called EditField RightDown - that one sets focus and this moves the cursor in ArrayCtrl

Note that the fundamental problem, one we have spent 2 hours of icq exchange with mdelfede yesterday, is that if you just resolve problem by changing EditField menu routine, adding ArrayCtrl Bar if ArrayCtrl is ancestor widget, you run into the issue with ArrayCtrl deleting lines - in that case widget gets deleted by menu.Execute in RightDown and the method contines afterwards.

That is why we need PostCallback.

Mirek

Subject: Re: Embed editfields in ArrayCtrl
Posted by mrjt on Fri, 08 Jan 2010 14:56:52 GMT

View Forum Message <> Reply to Message

luzr wrote on Fri, 08 January 2010 14:28mrjt wrote on Fri, 08 January 2010 08:49Except it doesn't work at all!

The problems:
- EditField doesn't have StdBar, are you thinking of EditText?

May I suggest you update your U++ from time to time? (was added 2-3 months back).
Suggestion noted!

I should have realised that if it was either you or me being wrong I didn't stand a very good chance

Subject: Re: Embed editfields in ArrayCtrl
Posted by mdelfede on Fri, 08 Jan 2010 17:39:37 GMT

View Forum Message <> Reply to Message

luzr wrote on Fri, 08 January 2010 13:57This seems to work quite nicely:
....

Mirek

Well, didn't test it, but it seems to me quite more complex than my small template... Which are the advantages ?
Mine just forwards the rightclick to ArrayCtrl; all process is then done there, so no problems deleting the control inside the menu handler.... tested and working.

If you want to append the EditField menu to the ArrayCtrl one it can be done easy within the ArrayCtrl... no time today, but I guess I could make a small ArrayCtrl derived class to do it.

What is the advantage of using postcallback ?

Max

Subject: Re: Embed editfields in ArrayCtrl
Posted by mdelfede on Fri, 08 Jan 2010 17:40:59 GMT

View Forum Message <> Reply to Message

luzr wrote on Fri, 08 January 2010 15:33

....

That is why we need PostCallback.

Mirek

Wrong, we don't

Max

Subject: Re: Embed editfields in ArrayCtrl
Posted by mirek on Fri, 08 Jan 2010 23:44:36 GMT

View Forum Message <> Reply to Message

mdelfede wrote on Fri, 08 January 2010 12:39luzr wrote on Fri, 08 January 2010 13:57This seems to work quite nicely:
....

Mirek

Well, didn't test it, but it seems to me quite more complex than my small template... Which are the advantages ?

It mixes EditField menu with ArrayCtrl menu.

Perhaps not that important, but I thought that should be possible, so I invested some effort to achieve it.

Quote:
What is the advantage of using postcallback ?

Generally, PostCallback is good to "break the scope" - the function gets called at the "baseline" global scope (not in EditField::RightDown scope).

Subject: Re: Embed editfields in ArrayCtrl
Posted by mirek on Fri, 08 Jan 2010 23:47:53 GMT

View Forum Message <> Reply to Message

mdelfede wrote on Fri, 08 January 2010 12:39luzr wrote on Fri, 08 January 2010 13:57This seems to work quite nicely:
....

Mirek

Well, didn't test it, but it seems to me quite more complex than my small template...

Not to say anything bad about your templat - it is clever "not knowing enough about interface definition" design. But it is sort of overkill - if you are going to override RightDown, you can skip the part about translating mouse coordinates and invoking ArrayCtrl's RightDown and just create the menu of only ArrayCtrl items and call Menu::Execute in RightDown...

Subject: Re: Embed editfields in ArrayCtrl
Posted by mdelfede on Sat, 09 Jan 2010 08:38:12 GMT

View Forum Message <> Reply to Message

luzr wrote on Sat, 09 January 2010 00:47

Not to say anything bad about your templat - it is clever "not knowing enough about interface definition" design. But it is sort of overkill - if you are going to override RightDown, you can skip the part about translating mouse coordinates...

ArrayCtrl::RightDown() doesn't need a correct mouse point ?

Quote:
 ....and invoking ArrayCtrl's RightDown and just create the menu of only ArrayCtrl items and call Menu::Execute in RightDown...

If I've right understood, you mean I should override EditField's RightDown(as I deed), just invoke ArrayCtrl's one (as in my template) but without need to translate mouse coords, then override also ArrayCtrl's RightDown() in order to invoke it's Menu::Execute ?
Besides that then I need to override ALSO the ArrayCtrl's RightDown (which forces me to derive from ArrayCtrl instead of using it directly), but how do then ArrayCtrl know on which row my mouse pointer is ? I think it's needed for row deletion, at least.
I added mouse translation just because on first try I didn't and the arrayctrl was always fetching the first row.

BTW, the best way would of course be to integrate the behaviour in ArryaCtrl, making it hook inside embedded ctrls menus, but then we would face again with the problem of Ctrls deletion, which can't be done without override Ctrls RightCtrl.
As you correctly pointed to me, the 'true' problem is not the ctrl removing from inside the callback by itself, but the fact that Ctrls RightDown tries to re-set focus on itself AFTER the callback execution.

Thinking a bit more, the solution should be :
1- Add the behaviour to ArrayCtrl, but...
2- Use your PostCallback way to handle the event

I think I'll try it today

Max

Subject: Re: Embed editfields in ArrayCtrl
Posted by mirek on Sat, 09 Jan 2010 11:47:45 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Sat, 09 January 2010 03:38luzr wrote on Sat, 09 January 2010 00:47

Not to say anything bad about your templat - it is clever "not knowing enough about interface definition" design. But it is sort of overkill - if you are going to override RightDown, you can skip the part about translating mouse coordinates...

ArrayCtrl::RightDown() doesn't need a correct mouse point ?

I mean that you can call menu.Execute directly, not care about ArrayCtrl RightDown. But now thinking about it, you need to move cursor too, right....

Quote:

Thinking a bit more, the solution should be :
1- Add the behaviour to ArrayCtrl, but...
2- Use your PostCallback way to handle the event

PostCallback sultion works quite well IMO. And it is sort of standard solution for these situations.

Mirek