

---

Subject: ArrayCtrl cell consisting of edit and helper button

Posted by [kevinle10@gmail.com](mailto:kevinle10@gmail.com) on Wed, 19 Apr 2006 02:13:53 GMT

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

---

What would be the easiest way to have an ArrayCtrl with cells whose each of them consists of an edit field and a helper button (button caption is like "..."). An example can be seen in VC++ 2005 by going to Tools->Options->VC++ Directories and look at the list on the right hand side.

Basically, the ArrayCtrl supports inline editing, but each cell at the time of being edited also shows a small button at the right edge.

Thanks

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [fudadmin](mailto:fudadmin) on Wed, 19 Apr 2006 04:56:09 GMT

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

---

kevinle10@gmail.com wrote on Wed, 19 April 2006 03:13: What would be the easiest way to have an ArrayCtrl with cells whose each of them consists of an edit field and a helper button (button caption is like "..."). An example can be seen in VC++ 2005 by going to Tools->Options->VC++ Directories and look at the list on the right hand side.

Basically, the ArrayCtrl supports inline editing, but each cell at the time of being edited also shows a small button at the right edge.

Thanks

I don't know what is the easiest but the first idea: I would try to make like a container control consisting of Edit and Button and try to add it to a column...

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [kevinle10@gmail.com](mailto:kevinle10@gmail.com) on Wed, 19 Apr 2006 05:16:45 GMT

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

---

Thanks. That makes sense. I'll try and let everyone knows. In the mean time, if someone else has some other ideas, I'll entertain them.

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [fudadmin](mailto:fudadmin) on Wed, 19 Apr 2006 05:38:35 GMT

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

---

Yes, it seems that it works ok. Took me 5 min to add to the other app.

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [fudadmin](#) on Wed, 19 Apr 2006 05:53:55 GMT

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

---

ok, you can try this and tell if it's what you expected.

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

```
class Ctrl2 : public ParentCtrl {
    EditField edit;
    Button button;
public:
    typedef Ctrl2 CLASSNAME;
    Ctrl2();
    ~Ctrl2() {}
};
```

```
Ctrl2::Ctrl2()
{
    edit.SetRect(0,0,50,15); //one problem are sizes >15...
    button.SetRect(55,0,85,15);
    button.SetLabel("Help!!!");
    Add(edit);
    Add(button);
}
```

```
void Extra2(One<Ctrl>& ctrl)
{
    ctrl.Create<Ctrl2>();
}
```

```
GUI_APP_MAIN
```

```
{
    TopWindow w;
    ArrayCtrl arr;

    arr.AddColumn("col1",2);
    arr.AddColumn("col2",5).Ctrls(Extra2);
    arr.SetLineCy(20);

    arr.Add("aaa");
    arr.Add("bbb");

    arr.SizePos();
    w.Add(arr);

    w.Run();
}
```

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [mirek](#) on Wed, 19 Apr 2006 12:31:54 GMT

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

---

BTW, consider using layout....

Also, you will likely have to implement GetData/SetData to pass editfield value through ArrayCtrl.

Mirek

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [kevinle10@gmail.com](#) on Wed, 19 Apr 2006 15:57:58 GMT

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

---

Hm, I wonder if subclassing from EditField and override the Paint() method along with some Mouse actions methods would be better. Again, the idea is to have the edit field occupying the entire cell until it gets the focus, then the button appears.

IN general, I just wonder if one of the goals of Ultimate++ from now on is to create more controls. That would truly make U++ ultimate.

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [fudadmin](#) on Wed, 19 Apr 2006 16:19:40 GMT

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

---

kevinle10@gmail.com wrote on Wed, 19 April 2006 16:57

IN general, I just wonder if one of the goals of Ultimate++ from now on is to create more controls. That would truly make U++ ultimate.

In this case, "ultimateness" depends on the amount of community members capabilities to contribute. The more of it, the more "ultimate"... So, you and everyone is welcome to join all the goals.

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [mirek](#) on Wed, 19 Apr 2006 16:35:54 GMT

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

---

kevinle10@gmail.com wrote on Wed, 19 April 2006 11:57

IN general, I just wonder if one of the goals of Ultimate++ from now on is to create more controls. That would truly make U++ ultimate.

In general, the primary goal of U++ is to make creating widgets very easy and make them very interoperable.

Mirek

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [forlano](#) on Fri, 05 May 2006 23:31:26 GMT

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

---

luzr wrote on Wed, 19 April 2006 14:31BTW, consider using layout....

Hello,

I've modified a bit the Aris' example above using the layout as Mirek suggested to create an embedded two-control made by an editint followed by a label. What is entered in the edit is immediately replicated in the label. Here it is:

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

#define LAYOUTFILE <edithelp/a.lay>
#include <CtrlCore/lay.h>

//----- begin class -----
class EditLabel : public WithEditLabel<TopWindow> {
public:
    typedef EditLabel CLASSNAME;
    void WriteLabel();
    EditLabel();
};

void EditLabel::WriteLabel()
{ lbl.SetLabel( AsString(~edit) );
  //SetFocus();
}

EditLabel::EditLabel()
{ CtrlLayout(*this, "");
  edit <=< THISBACK(WriteLabel);
}

void Extra2(One<Ctrl>& ctrl)
{ ctrl.Create<EditLabel>();
}

//----- end class -----

GUI_APP_MAIN
{
    TopWindow w;
    ArrayCtrl arr;
```

```
arr.AddColumn("col1",20);
arr.AddColumn("col2",20).Ctrls(Extra2);
arr.SetLineCy(20);
```

```
arr.Add("aaa");
arr.Add("bbb");
```

```
arr.SizePos();
w.Add(arr);
```

```
w.Run();
}
```

```
LAYOUT(EditLabel, 96, 19)
ITEM(EditInt, edit, LeftPosZ(2, 32).TopPosZ(1, 17))
ITEM(Label, lbl, SetFrame(ThinInsetFrame()).LeftPosZ(36, 60).TopPosZ(1, 17))
END_LAYOUT
```

Now the horizontal alignment is improved drastically and there is no problem with size > 15 that can be tuned as one like.

Moreover appear the annoying problem of the cursor of the array that does not follow the click on the embedded control. To overcome it I've added

```
SetFocus();
```

connected to edit.WhenAction but then come two new unwanted side effects:

- 1) WhenAction was already used to pass the value to the label at the right. So while one type in the editint it continue to focus when should not be necessary (in principle).
- 2) Instead happens that after the first SetFocus() the cursor of the array is synchronised with the embedded control but the focus escape from editint and finish somewhere else in the row and one cannot continue the edit of the field!

Any idea to overcome the problem?

Luigi

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [fudadmin](#) on Sat, 06 May 2006 00:04:23 GMT

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

---

forlano wrote on Sat, 06 May 2006 00:31

Any idea to overcome the problem?

Luigi

Morning always brings the solution...

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [forlano](#) on Sat, 06 May 2006 00:13:55 GMT

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

---

fudadmin wrote on Sat, 06 May 2006 02:04forlano wrote on Sat, 06 May 2006 00:31  
Any idea to overcome the problem?

Luigi

Morning always brings the solution...

I hope .

Your twoctrl can be very useful... if I knew how to handle it!

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [fudadmin](#) on Sat, 06 May 2006 00:58:26 GMT

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

---

try this:

```
void EditLabel::WriteLabel()
{
    lbl.SetLabel( AsString(~edit) );
    SetFocus();
    edit.SetFocus();
    edit.SetSelection(edit.GetLength());
}
```

btw, why do you use a label and not a button?

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [fudadmin](#) on Sat, 06 May 2006 01:03:32 GMT

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

---

Also, try to change (with the above)

WithEditLabel<TopWindow> -> WithEditLabel<ParentCtrl>

and

CtrlLayout(\*this,""); -> CtrlLayout(\*this);

also:

ctrl.Create<EditLabel>().SizePos();

and this:

```
LAYOUT(EditLabel, 124, 32)
```

```
ITEM(EditInt, edit, LeftPosZ(12, 32).VCenterPosZ(17, 1))
```

```
ITEM(Label, lbl, SetFrame(ThinInsetFrame()).LeftPosZ(64, 56).VCenterPosZ(16, 0))
```

```
END_LAYOUT
```

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [forlano](#) on Sat, 06 May 2006 08:38:39 GMT

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

---

fudadmin wrote on Sat, 06 May 2006 02:58try this:

```
void EditLabel::WriteLabel()
{
    lbl.SetLabel( AsString(~edit) );
    SetFocus();
    edit.SetFocus();
    edit.SetSelection(edit.GetLength());
}
```

btw, why do you use a label and not a button?

Thanks!

Everything work perfectly. Your improvements automatically include a feature that I wanted: each time I pressed the TAB key the focus pass to the next editint.

I need a label to show the match result [1-0], [0-1] etc..., while in the editint I enter their short code 1, 0, and so on.

Let me ask you (this is a good moment for it) what is the difference between deriving from TopWindow and CtrlParent. It seems the second one is better for embedded widget in that it merge better with the around widget. While the first one seems more indicate for indipendent widget like dialog. Is it correct?

Luigi

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button

Posted by [forlano](#) on Sat, 06 May 2006 08:44:12 GMT

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

---

forlano wrote on Sat, 06 May 2006 10:38fudadmin wrote on Sat, 06 May 2006 02:58try this:

```
void EditLabel::WriteLabel()
{
    lbl.SetLabel( AsString(~edit) );
    SetFocus();
    edit.SetFocus();
    edit.SetSelection(edit.GetLength());
}
```

btw, why do you use a label and not a button?

Thanks!

Everything work perfectly. Your improvements automatically include a feature that I wanted: each time I pressed the TAB key the focus pass to the next editint.  
I need a label to show the match result [1-0], [0-1] etc..., while in the editint I enter their short code 1, 0, and so on.

Let me ask you (this is a good moment for it) what is the difference between deriving from TopWindow and CtrlParent. It seems the second one is better for embedded widget in that it merge better with the around widget. While the first one seems more indicate for indipendent widget like dialog. Is it correct?

BTW, can you add two words for the tricky row:  
`edit.SetSelection(edit.GetLength());`

without it I can't enter more than one char.

Luigi

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button  
Posted by [fudadmin](#) on Sat, 06 May 2006 09:48:16 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

forlano wrote on Sat, 06 May 2006 09:44

Let me ask you (this is a good moment for it) what is the difference between deriving from TopWindow and CtrlParent. It seems the second one is better for embedded widget in that it merge better with the around widget. While the first one seems more indicate for indipendent widget like dialog. Is it correct?

Luigi

Yes, I think so.

---

---

Subject: Re: ArrayCtrl cell consisting of edit and helper button  
Posted by [mirek](#) on Sun, 07 May 2006 17:50:39 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Quote:

Let me ask you (this is a good moment for it) what is the difference between deriving from TopWindow and CtrlParent. It seems the second one is better for embedded widget in that it merge better with the around widget. While the first one seems more indicate for indipendent widget like dialog. Is it correct?

The main difference is that TopWindow is kind of heavy - it contains all data needed to "run" top-level window.



