
Subject: Ctrl * GetCallbackCtrl()
Posted by [unodgs](#) on Tue, 31 Oct 2006 07:56:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi!

Now in upp isn't possible to obtain ctrl address in its callback method eg:

```
Button btn;  
btn <<= THISBACK(Action);
```

```
void Action()  
{  
    //What ctrl is calling this action??  
    Ctrl * ctrl = GetCallbackCtrl();  
}
```

Such method should work at least for default WhenAction (with such a restriction it can be coded at Ctrl class level)

I need it for my grid. I just have to move cursor to the proper line of grid - line where clicked ctrl is. Because now child control grabs all events the underlying grid doesn't receive any messages from system and for example LeftDown is not called (where I set the cursor line)

Is adding such a method possible?

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [mirek](#) on Tue, 31 Oct 2006 10:34:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

I think we already have a better option:

```
Button btn;  
btn <<= THISBACK1(Action, line);
```

```
void Action(int line)  
{  
    ...  
}
```

Of course to what degree this is applicable to your problem is debatable (but I think GetCallbackCtrl does not solve this problem as well).

In ArrayCtrl, the issue is solved using ChildGotFocus...

Mirek

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [unodgs](#) on Tue, 31 Oct 2006 12:08:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Tue, 31 October 2006 05:34I think we already have a better option:

```
Button btn;  
btn <<= THISBACK1(Action, line);
```

```
void Action(int line)  
{  
...  
}
```

Line number is unknown until ctrl is clicked...

Quote:

Of course to what degree this is applicable to your problem is debatable (but I think GetCallbackCtrl does not solve this problem as well).

In ArrayCtrl, the issue is solved using ChildGotFocus...

Mirek

ChildGotFocus is not enough. Option, Button, DropList don't get focus when you click on them using left mouse button...
It works fine only for Edit*

Try:

```
void MakeDropList(One<Ctrl>& ctrl)  
{  
    ctrl.Create<DropList>()  
        .Add(0, "None")  
        .Add(1, "Minimal")  
        .Add(2, "Full");  
}
```

```
void MakeButton(One<Ctrl>& ctrl)  
{  
    ctrl.Create<Button>();  
    ctrl->WantFocus();  
}
```

```
void MakeEdit(One<Ctrl>& ctrl)  
{  
    ctrl.Create<EditString>();  
}
```

```
ctrl->WantFocus();  
}
```

```
arr.AddColumn("1").Ctrls(MakeDropList);  
arr.AddColumn("2").Ctrls(MakeButton);  
arr.AddColumn("3").Ctrls(MakeEdit);
```

and click at columns with button or droplist. Cursor wont change its position.

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [mirek](#) on Tue, 31 Oct 2006 12:20:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

unodgs wrote on Tue, 31 October 2006 07:08luzr wrote on Tue, 31 October 2006 05:34I think we already have a better option:

```
Button btn;  
btn <<= THISBACK1(Action, line);
```

```
void Action(int line)  
{  
...  
}
```

Line number is unknown until ctrl is clicked...

How so? You do not know on which line you have placed the Button?

Anyway, instead of int line, you can use Ctrl *ctrl and THISBACK1(Action, &btn).

Quote:

ChildGotFocus is not enough. Option, Button, DropList don't get focus when you click on them using left mouse button...
It works fine only for Edit*

Of course. That is what I expect from the ArrayCtrl (moving focus moves cursor).

Actually, what I wanted to say is that perhaps not even WhenAction is enough here.... but OTOH all depends on what you really expect.

Quote:

and click at columns with button or droplist. Cursor wont change its position.

Yes.

Mirek

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [unodgs](#) on Tue, 31 Oct 2006 12:46:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:How so? You do not know on which line you have placed the Button?

Anyway, instead of int line, you can use Ctrl *ctrl and THISBACK1(Action, &btn).

Yes, you're right, line is constant for ctrl..
Second solution is even better, I already thought about that.

Quote:Of course. That is what I expect from the ArrayCtrl (moving focus moves cursor).

Actually, what I wanted to say is that perhaps not even WhenAction is enough here.... but OTOH all depends on what you really expect.

The problem is that in my case cursor should always change line if click is in another place. It is connected with callbacks
WhenUpdateRow, WhenUpdateCell mostly used to updating a database. If I click option ctrl in cell the value of cell is changed - WhenUpdateCell should be called and cursor should be moved to this row - to update previous row and to be prepare to call WhenUpdateRow at the new row.
Moving cursor for all controls is more logical to me. Wide used developer express grid behave the same.

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [unodgs](#) on Tue, 31 Oct 2006 13:01:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:Anyway, instead of int line, you can use Ctrl *ctrl and THISBACK1(Action, &btn).

Ok, I used similar form, and it works perfectly. I wonder why I didn't wanted to do it earlier...

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [unodgs](#) on Thu, 02 Nov 2006 08:36:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok, it's not so perfect... Because if I change cursor in ChildGotFocus for edits the cursor change

its place immediately on left click. For rest of controls cursor change place in WhenAction routine and it happens mostly on left mouse up.

Anyway I patched a little bit CtrlCore and I hope you'll accept this. What I've done is:

CtrlCore.h : Ctrl class

```
virtual void ChildMouseAction(Ctrl * ctrl, int event, Point p) {}
```

CtrlMouse.cpp

```
Image Ctrl::DispatchMouseEvent(int e, Point p, int zd) {
    if(!IsEnabled())
        return Image::Arrow();
    if(captureCtrl && captureCtrl != this && captureCtrl->IsMouseActive())
        return captureCtrl->MEvent0(e, p + GetScreenRect().TopLeft() -
            captureCtrl->GetScreenRect().TopLeft(), zd);
    Ctrl *top = this;
    if(e == MOUSEWHEEL && !GetParent()) {
        Ctrl *w = GetFocusCtrl();
        if(w) {
            top = w->GetTopCtrl();
            p = GetMousePos() - top->GetScreenRect().TopLeft();
        }
    }
    Ctrl *q = top->ChildFromPoint(p);
    if(parent)
        parent->ChildMouseAction(q, e, p);
    return q ? q->DispatchMouseEvent(e, p, zd) : top->MEvent0(e, p, zd);
}
```

This line was added:

```
if(parent) parent->ChildMouseAction(q, e, p);
```

With such a simple modification I can override ChildMouseAction in my grid ctrl and has control over mouse events before they go to any child control.

What do you think about it?

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [unodgs](#) on Thu, 02 Nov 2006 08:41:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

There should be

```
if(parent && q)
parent->ChildMouseAction(q, e, p);
```

ChildMouseAction should be renamed to ChildMouseEvent I think..

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [mirek](#) on Thu, 02 Nov 2006 13:39:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

unodgs wrote on Thu, 02 November 2006 03:36Ok, it's not so perfect...

Quote:

Because if I change cursor in ChildGotFocus for edits the cursor change its place immediately on left click. For rest of controls cursor change place in WhenAction routine and it happens mostly on left mouse up.

Anyway I patched a little bit CtrlCore and I hope you'll accept this. What I've done is:

CtrlCore.h : Ctrl class

```
virtual void ChildMouseAction(Ctrl * ctrl, int event, Point p) {}
```

CtrlMouse.cpp

```
Image Ctrl::DispatchMouseEvent(int e, Point p, int zd) {
if(!IsEnabled())
return Image::Arrow();
if(captureCtrl && captureCtrl != this && captureCtrl->IsMouseActive())
return captureCtrl->MEvent0(e, p + GetScreenRect().TopLeft() -
captureCtrl->GetScreenRect().TopLeft(), zd);
Ctrl *top = this;
if(e == MOUSEWHEEL && !GetParent()) {
Ctrl *w = GetFocusCtrl();
if(w) {
top = w->GetTopCtrl();
p = GetMousePos() - top->GetScreenRect().TopLeft();
}
}
Ctrl *q = top->ChildFromPoint(p);
if(parent)
```

```
parent->ChildMouseAction(q, e, p);  
return q ? q->DispatchMouseEvent(e, p, zd) : top->MEvent0(e, p, zd);  
}
```

This line was added:

```
if(parent) parent->ChildMouseAction(q, e, p);
```

With such a simple modification I can override ChildMouseAction in my grid ctrl and has control over mouse events before they go to any child control.

What do you think about it?

Have you tested? This does not seem to be correct to me; if I remember well, parent is always NULL here.... (DispatchMouseEvent is called by host platform for TopWindow).

Anyway, I think the requirement itself is sound.

Mirek

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [unodgs](#) on Thu, 02 Nov 2006 13:51:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:Have you tested? This does not seem to be correct to me; if I remember well, parent is always NULL here.... (DispatchMouseEvent is called by host platform for TopWindow).

Well, it works at least in my gridtest. Grid instance is receiving children mouse messages.

Quote:

Anyway, I think the requirement itself is sound.

Does it mean something similar will be delivered ? If yes that would be great! Hope to see this very soon...

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [mirek](#) on Thu, 02 Nov 2006 13:55:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Should not the default implementation call ChildMouseEvent for parent?

Also, check MouseHook (you could implement the same thing most likely with hooking mouse messages, but with a bit more trouble...).

Mirek

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [unodgs](#) on Thu, 02 Nov 2006 14:11:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Thu, 02 November 2006 08:55 Should not the default implementation call ChildMouseEvent for parent?

I don't understand. In my modification it is just called for parent!

Quote:

Also, check MouseHook (you could implement the same thing most likely with hooking mouse messages, but with a bit more trouble...).

I'm aware of MouseHook, but it's too general. It's ok for tooltips but not for specific cases.

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [mirek](#) on Thu, 02 Nov 2006 14:15:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
virtual void ChildMouseAction(Ctrl * ctrl, int event, Point p)
{
    if(parent) parent->ChildMouseAction(ctrl, event, p);
}
```

Mirek

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [unodgs](#) on Thu, 02 Nov 2006 14:52:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Thu, 02 November 2006 09:15

```
virtual void ChildMouseAction(Ctrl * ctrl, int event, Point p)
{
    if(parent) parent->ChildMouseAction(ctrl, event, p);
}
```

Mirek

Good question. I have to think it over.

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [unodgs](#) on Thu, 02 Nov 2006 21:48:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

unodgs wrote on Thu, 02 November 2006 09:52luzr wrote on Thu, 02 November 2006 09:15

```
virtual void ChildMouseAction(Ctrl * ctrl, int event, Point p)
{
    if(parent) parent->ChildMouseAction(ctrl, event, p);
}
```

Mirek

Good question. I have to think it over.

It seems that is unnecessary. I have created custom ctrl with StaticRect and Option in this rect. Then I add it to grid for each row (Ctrls()). Even if I was clicking most inner ctrl grid was receiving mouse events of this ctrl. It seems that one call in DispatchMouseEvent is enough as it sends events through full ctrls chain.

Subject: Re: Ctrl * GetCallbackCtrl()
Posted by [mirek](#) on Thu, 02 Nov 2006 21:54:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

unodgs wrote on Thu, 02 November 2006 16:48unodgs wrote on Thu, 02 November 2006 09:52luzr wrote on Thu, 02 November 2006 09:15

```
virtual void ChildMouseAction(Ctrl * ctrl, int event, Point p)
{
    if(parent) parent->ChildMouseAction(ctrl, event, p);
}
```

Mirek

Good question. I have to think it over.

It seems that is unnecessary. I have created custom ctrl with StaticRect and Option in this rect. Then I add it to grid for each row (Ctrls()). Even if I was clicking most inner ctrl grid was receiving mouse events of this ctrl. It seems that one call in DispatchMouseEvent is enough as it sends events through full ctrls chain.

If that is true, something is wrong

(My bet is that "wrong" is implementation - DispatchMouseEvent really is not the right place).

Mirek

Subject: Re: Ctrl * GetCallbackCtrl()

Posted by [unodgs](#) on Thu, 02 Nov 2006 22:08:03 GMT

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

Quote:If that is true, something is wrong

(My bet is that "wrong" is implementation - DispatchMouseEvent really is not the right place).

I've only tried to solve my problem and my first thought was that DispatchMouseEvent is the right place. If you know better / more correct solution please implement it! I really need it.
