Subject: OptionTree - something missing? [FIXED]
Posted by James Thomas on Thu, 21 Sep 2006 14:59:31 GMT
View Forum Message <> Reply to Message

Currently there doesn't seem to be a way of correctly configuring the OptionTree control from code, if you set one of the options it's parent does not get correctly updated.

In the example below I set an option using two different methods and neither results in a change to the parent nodes state (it should turn grey).

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

class AWindow : public TopWindow
{
public:
 typedef AWindow CLASSNAME;
 Option _optA, _optAA, _optAAA, _optAAB, _optAAC;
 OptionTree _tree;
 Button _btn;

 AWindow()
 {
  _tree.SetRoot(_optA, "Root");
  _tree.Add(0, _optAA, "Node");
  _tree.Add(1, _optAAA, "Leaf 1");
  _tree.Add(1, _optAAB, "Leaf 2");
  _tree.Add(1, _optAAC, "Leaf 3");


  _tree.LeftPosZ(0, 140).TopPosZ(0, 164);
  _btn.LeftPosZ(0, 140).TopPosZ(164, 20);
  _btn <<= THISBACK(ButtonPush);

  Add(_tree);
  Add(_btn);

  _tree.Set(3, 1); // Set through the TreeCtrl::Set - no call to SetOption
 }

 void ButtonPush()
 {
  _optAAC.Set(1); // Set using the Option ctrl - also no call to SetOption
 }
};

GUI_APP_MAIN
{
 AWindow w;
```

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

The work of setting the parent node is done in the OptionTree::SetOption function, which never gets called. I can understand that setting the option directly probably shouldn't work, but calling the Set function should. Perhaps there needs to be an override of TreeCtrl::Set() so that SetOption can be called, but there may be a more elegant way.

Strangely I think this used to work in release 602 because I'm sure I would have noticed, but I can't find any relevant changes to Option or OptionTree so I could be wrong.

EDIT: I'm using 605 RC, but I've looked at the dev source and the problem looks like it's still there.

Subject: Re: OptionTree - something missing?
Posted by mirek on Thu, 21 Sep 2006 19:04:52 GMT
View Forum Message <> Reply to Message

Fixed (with not very nice fix..).

You however need to use TreeCtrl::Set, because with current widget model, there is no way how to detect non-user update in Option by OptionTree (sort of corner case).

Subject: Re: OptionTree - something missing?
Posted by mirek on Thu, 21 Sep 2006 19:47:34 GMT
View Forum Message <> Reply to Message

P.S.: Thanks for the good example.

Subject: Re: OptionTree - something missing?
Posted by James Thomas on Fri, 22 Sep 2006 09:56:36 GMT
View Forum Message <> Reply to Message

Thank you for the quick fix, I seem to be the only person using this control

---

Subject: Re: OptionTree - something missing?
Posted by nixnixnix on Sat, 07 Apr 2007 04:18:38 GMT
View Forum Message <> Reply to Message

I'm using it too and I have a question:

is it possible to select items as well as checking and unchecking the options?

I have a GIS style interface in which I use the options to set visibility but I also want to be able to have a selected item. at the moment clicking on the item activates the option. would it be possible to configure it so that the user has to click on the actual option box and leave the item text to detect selection?

great control - very easy to use. I'm just one of those people who is never happy

Nick

p.s. any examples of changing the item hierarchy by dragging and dropping items?

Subject: Re: OptionTree - something missing?
Posted by mirek on Sat, 07 Apr 2007 07:04:31 GMT
View Forum Message <> Reply to Message

You have to use TreeCtrl and little magic:

```
struct MyOption : Option {
 virtual void LeftDown(Point p, dword keyflags) {
  SetFocus();
  Option::LeftDown(p, keyflags);
 }
};

.......
TreeCtrl tree;
Array<MyOption> option; // clear before loading tree...
.......

TreeCtrl::Node node(option.Add());
option.Top().SetLabel(text);
int id = tree.Add(parentid, node);
```

Mirek

Subject: Re: OptionTree - something missing?
Posted by nixnixnix on Wed, 11 Apr 2007 04:21:48 GMT
View Forum Message <> Reply to Message

Thanks Mirek, that worked a treat

ended up looking like this

```
struct TreeOption : Option {
 virtual void LeftDown(Point p, dword keyflags) {
  if(p.x<this->GetSize().cy) // in the checkbox
   Option::LeftDown(p, keyflags); // def func
  if(this->GetData()==true) // explicit for clarity
   SetFocus(); //shame there is no killfocus()
 }
};
```

are there any examples of changing the tree item hierarchy by dragging and dropping or do I need to go do it with a recttracker and if so can I show the contents whilst dragging? guess i would need to subclass recttracker?

thanks again,

nick

Subject: Re: OptionTree - something missing?
Posted by mirek on Wed, 11 Apr 2007 07:59:17 GMT
View Forum Message <> Reply to Message

Recttracker is not the best bet here (contrained to single widget).

You should use LocalLoop and CursorImage to do the job.

Other than that, generalised D&D support is major drawback of current U++ and subject of intense research...

Mirek

Subject: Re: OptionTree - something missing?
Posted by nixnixnix on Wed, 11 Apr 2007 21:35:15 GMT
View Forum Message <> Reply to Message

thanks again Mirek. Have looked for LocalLoop in the documentation and on the forums etc but not finding it. Any examples you know of?

LocalLoop only appears to have three loopy functions: GetMaster; SetMaster and Run so am not sure what to do with it.

Nick

p.s. a search for getmaster or setmaster return nothing so am a bit stumped but if I can get it working nice all in a derived class of TreeCtrl, might be a good addition to UPP? Will definitely post the class on here as an example if i can get it working

---

Subject: Re: OptionTree - something missing?
Posted by mirek on Thu, 12 Apr 2007 13:05:43 GMT
View Forum Message <> Reply to Message

nixnixnix wrote on Wed, 11 April 2007 17:35thanks again Mirek. Have looked for LocalLoop in the documentation and on the forums etc but not finding it. Any examples you know of?

LocalLoop only appears to have three loopy functions: GetMaster; SetMaster and Run so am not sure what to do with it.

Ops, sorry. I thought it is already documented...

Anyway, Run "runs a local modal loop over Master" and recieves all events intended to Master.

Means, until you break the modal loop, all events will be passed to this special Ctrl (in fact, sort of pseudo-ctrl).

Look at RectTracker code, it is implemented using LocalLoop.

Mirek

Subject: Re: OptionTree - something missing?
Posted by nixnixnix on Fri, 13 Apr 2007 01:58:31 GMT
View Forum Message <> Reply to Message

ok think i got that but how does run stop? I want it to stop when the left mouse button gets released and then I want to know where the mouse is so I can check for drop targets.

So far, the mouse cursor doesnt change when am dragging and the run doesnt appear to have any stop condition. Do I need to override LocalLoop?

Nick

```
class LayerTree : public TreeCtrl
{
public:
 typedef LayerTree CLASSNAME;
```

```
 virtual void RightDown(Point p, dword flags);
 virtual void LeftDown(Point p, dword flags);

 int GetNodeIDAt(Point p);

};
....

void LayerTree::LeftDown(Point p, dword flags)
{
 // here we implement drag and drop for nodes in the tree view


 // first see if there is a node at this point p
 int i,id=GetNodeIDAt(p);

 if(id<0)
  return;  // didnt click on a node

 Rect rc = GetNode(id).ctrl->GetRect();

 // draw the node into an image
 // sample the tree ctrls view in this rectangle and make an image
 Size sz(rc.Width(),rc.Height());

 ImageDraw w(sz);

 GetNode(id).ctrl->DrawCtrl(w);

 Image img = w;

 // start local loop and see where it ends
 RectTracker rt(*this);

 // use the image as a cursor
 rt.SetCursorImage(img);

 rt.SetMaster(*this);

 rt.Run();
// rt.Track(rc,ALIGN_CENTER,ALIGN_CENTER);


}
```

```
int LayerTree::GetNodeIDAt(Point p)
{
 // step through all nodes in the tree
 // and see if one of them contains p
 int id,n=GetLineCount(); //all (visible) items
 Ctrl* ptr;
 Rect rc;

 for(int i=0;i<n;i++)
 {
  id = GetItemAtLine(i);
  TreeCtrl::Node node = GetNode(id);
  ptr = node.ctrl;
  if(ptr)
  {
   rc = ptr->GetRect();

   if(rc.Contains(p))
    return id;
  }
 }

 return -1;
}
```

hmmm, should have start my own thread really...

I tried the following

```
void LayerTree::LeftUp(Point p, dword flags)
{
 m_pRT->EndLoop();  // tried this with recttracker as member pointer

 EndLoop();  // and this
}
```

and nothing appears to have any effect. Am still quite new to UPP and it seems I still have a lot to learn. There does appear to be a loop starting. But I dont understand why the cursor does not change and I don't understand why the master (LayerCtrl) cannot intercept LeftUp and stop the loop. I'm missing something fundamental and obvious it seems.

I think this is the most complex part of my GUI but I need to know it can be done before I go further.

Cheers,

Nick

Subject: Re: OptionTree - something missing?
Posted by mirek on Fri, 13 Apr 2007 07:46:50 GMT
View Forum Message <> Reply to Message

nixnixnix wrote on Thu, 12 April 2007 21:58ok think i got that but how does run stop?

It is called LocalLoop because it locally takes control about handling input events for the Master. All events go into LocalLoop virtual methods.

You have to call EndLoop for the LocalLoop, not the TreeCtrl.

BTW, there is also PointLooper class, maybe it can be helpful here too.

Mirek

Subject: Re: OptionTree - something missing?
Posted by nixnixnix on Sun, 15 Apr 2007 04:58:57 GMT
View Forum Message <> Reply to Message

thanks for the continuing help and advice.

I tried PointLooper and it almost works but not quite in that it will paint the option ctrl but the rest is always black and too small in any case. (Is there some limit on the size of the cursor?)

However, I think I know what direction to take. I am attempting to derive a new class from RectTracker so I can customise the way it draws. So far its messy but promising

Cheers,

Nick
