Subject: treectrl with ctrl
Posted by sapiency on Wed, 26 Aug 2009 13:39:05 GMT
View Forum Message <> Reply to Message

hi,

with controls in Nodes it is not possible to use 'Value' and 'Key'

```
void  TreeCtrl::Set(int id, Value v)
{
 Item& m = item[id];
 if(m.ctrl)
  m.ctrl->SetData(v);
 else {
  m.value = m.key = v;
  RefreshItem(id);
 }
 SetOption(id);
}

void  TreeCtrl::Set(int id, Value k, Value v)
{
 Item& m = item[id];
 if(m.ctrl)
  m.ctrl->SetData(v);
 else {
  m.key = k;
  m.value = v;
  RefreshItem(id);
 }
 SetOption(id);
}
```

This could be solved if there would be a virtual method in Ctrl

```
virtual void SetData(const Value& data, const Value& value) {}
```

and modify the method in TreeCtrl to:

```
void  TreeCtrl::Set(int id, Value k, Value v)
{
 Item& m = item[id];
 if(m.ctrl)
  m.ctrl->SetData(k, v);
 else {
  m.key = k;
  m.value = v;
  RefreshItem(id);
 }
 SetOption(id);
}
```

Maybe it is possible to extend the code

regards

reinhard

Subject: Re: treectrl with ctrl
Posted by mirek on Fri, 28 Aug 2009 09:47:42 GMT

View Forum Message <> Reply to Message

sapiency wrote on Wed, 26 August 2009 09:39hi,

with controls in Nodes it is not possible to use 'Value' and 'Key'

```
void  TreeCtrl::Set(int id, Value v)
{
 Item& m = item[id];
 if(m.ctrl)
  m.ctrl->SetData(v);
 else {
  m.value = m.key = v;
  RefreshItem(id);
 }
 SetOption(id);
}

void  TreeCtrl::Set(int id, Value k, Value v)
{
 Item& m = item[id];
 if(m.ctrl)
  m.ctrl->SetData(v);
 else {
  m.key = k;
  m.value = v;
  RefreshItem(id);
 }
 SetOption(id);
}
```

This could be solved if there would be a virtual method in Ctrl

virtual void SetData(const Value& data, const Value& value) {}

and modify the method in TreeCtrl to:

```
void  TreeCtrl::Set(int id, Value k, Value v)
{
 Item& m = item[id];
 if(m.ctrl)
  m.ctrl->SetData(k, v);
 else {
  m.key = k;
  m.value = v;
  RefreshItem(id);
 }
 SetOption(id);
}
```

Maybe it is possible to extend the code

regards

reinhard

Well, I think the design is quite right for value - widget now represents the Value of node. This behaviour is also consistent with ArrayCtrl.

There is a sort of question of what to do with key though. I think we should try something better...

Mirek

Subject: Re: treectrl with ctrl
Posted by sapiency on Fri, 28 Aug 2009 23:26:33 GMT
View Forum Message <> Reply to Message

hi Mirek,

ok, I didn't looked at ArrayCtrl ...
and I forgot to look at the details to get the key back too ...

I'll try get the data I need in another way.

thanks

reinhard

Subject: Re: treectrl with ctrl
Posted by kohait00 on Mon, 19 Oct 2009 07:07:27 GMT
View Forum Message <> Reply to Message

the goal behind all this is to be able to put something down in the TreeCtrl (be it a key/value pair or a Ctrl content), tag with a kind of hash information and later be able to retrieve exactly this element again, using the key. well, as it seems it works quite well with the key/value approach, but using ctrl as content, there is no way to assign a hash, that is kept *inside* the tree ctrl and not propagated to the ctrl itself. Find'ing the key then results in a for loop, which again decides, where to take the value, from the internal key database or from the control.
but imagine controls that dont support GetData/SetData, like simple ParentCtrl containing others..this would not work.

thats somehow a drawback in double sense (we use the keys only for verification, not for hasching, --> performance)

maybe instead of setting up another SetData() in the Ctrl:: one could just think of an additional method in the TreeCtrl (and maybe the other controls with the same respective behaviour), something like a

void SetK(int id, Value key); which does *not* propagate it to a ctrl's SetData but still sets the internal keys, and a FindK(Value key) method which only searches the internal keys, no matter what content, control or key/value.

so one could still use the Find(Value key) methods, and use the others if one knows what to do..
changig the API is hard, when we have a lot of progs already used to this behaviour.

cheers

Subject: Re: treectrl with ctrl
Posted by mirek on Wed, 21 Oct 2009 06:45:12 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Mon, 19 October 2009 03:07the goal behind all this is to be able to put something down in the TreeCtrl (be it a key/value pair or a Ctrl content), tag with a kind of hash information and later be able to retrieve exactly this element again, using the key. well, as it seems it works quite well with the key/value approach, but using ctrl as content, there is no way to assign a hash, that is kept *inside* the tree ctrl and not propagated to the ctrl itself. Find'ing the key then results in a for loop, which again decides, where to take the value, from the internal key database or from the control.
but imagine controls that dont support GetData/SetData, like simple ParentCtrl containing

others..this would not work.

thats somehow a drawback in double sense (we use the keys only for verification, not for hasching, --> performance)

maybe instead of setting up another SetData() in the Ctrl:: one could just think of an additional method in the TreeCtrl (and maybe the other controls with the same respective behaviour), something like a

void SetK(int id, Value key); which does *not* propagate it to a ctrl's SetData but still sets the internal keys, and a FindK(Value key) method which only searches the internal keys, no matter what content, control or key/value.

so one could still use the Find(Value key) methods, and use the others if one knows what to do.. changig the API is hard, when we have a lot of progs already used to this behaviour.

cheers

I guess things got a little bit confused. There are now 3 values associated with each node:

id
key
value

Embedded ctrls are using value. You can set key alone using

Set(id, key, GetValue(id))

the only a little bit tricky part is that if you set just single Value for node, it gets assigned to both key and value...

Now above is a little bit confused, but so am I reading your post

Mirek