
Subject: PROPOSAL: TreeCtrl optionally owns Ctrl's in Node

Posted by [kohait00](#) on Wed, 15 Dec 2010 14:52:31 GMT

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

hi all

ArrayCtrl has got the possibility to own the Ctrl's if they are passed as pointer, so one can create some on the fly and have the TreeCtrl take care of their destruction. this is great for editing the represented values.

it does not break the 'everything-belongs-somewhere-rule'. TreeCtrl is then the container..

since these two, ArrayCtrl and TreeCtrl are the major complex grouping Ctrl's they should support it and be of similar skills.

what do you think?

EDIT: at first i thought it is easy task, since TreeCtrl is already setup as to have Ctrl's, but not owned. but i realize that TreeCtrl::Node and TreeCtrl::Item handling is not fitting in design as to handle ownership.

maybe someone could help here. attached is the so far changed TreeCtrl sources

File Attachments

1) [TreeCtrl.h](#), downloaded 528 times

2) [TreeCtrl.cpp](#), downloaded 501 times

Subject: Re: PROPOSAL: TreeCtrl optionally owns Ctrl's in Node

Posted by [kohait00](#) on Tue, 11 Jan 2011 14:03:32 GMT

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

any advances on that one?

Subject: Re: PROPOSAL: TreeCtrl optionally owns Ctrl's in Node

Posted by [mirek](#) on Sun, 16 Jan 2011 10:26:25 GMT

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

Well, I guess there is no real demand to introduce this now...

Frankly, in this department, much more pressing issue is optimization of how ArrayCtrl is handling widgets (it gets quite slow if there is enough lines).

Subject: Re: PROPOSAL: TreeCtrl optionally owns Ctrl's in Node

Posted by [kohait00](#) on Tue, 18 Jan 2011 09:38:42 GMT

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

not quite sure about the demand. see the downloads of the file...

i think it's be great to have them both..not to have to take care of the controls you set up in the TreeCtrl is just a nice thing, keeping user code cleaner (no need to take care of add's and removes)

ps: thanks for the improvements on arrayctrl

Subject: Re: PROPOSAL: TreeCtrl optionally owns Ctrl's in Node

Posted by [mirek](#) on Thu, 20 Jan 2011 13:32:47 GMT

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

Could send me some file so that I can test it?

(Sorry for asking, too lazy to clober together one and I believe you must have had something like for debugging it

Mirek

Subject: Re: PROPOSAL: TreeCtrl optionally owns Ctrl's in Node

Posted by [kohait00](#) on Thu, 20 Jan 2011 14:40:54 GMT

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

thanks for taking on it..

here again a complete packet, based on current revision

i think the major problem is the DND stuff, which uses copies of Node instances. while ArrayCtrl moves them i think.

File Attachments

1) [TreeCtrl.rar](#), downloaded 470 times

Subject: Re: PROPOSAL: TreeCtrl optionally owns Ctrl's in Node

Posted by [kohait00](#) on Mon, 23 May 2011 20:57:08 GMT

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

i will be warming up this thing, but need an advice in which direction to go:

the key problem in TreeCtrl is that Node is meant to be passed around by-value, i.e. performing DnD movings or using the Copy() function..

when a Node references a Ctrl which it owns, it cant be passed around by-value, it either needs to be picked (dtor would take care of destrucion of Ctrl) or a different Method for Moving/DnD of the Nodes needs to be found to avoid copy which would duplicate the Node and double ownership. so Node might need to stay unmouvable (exposed only as const or have a subclass which actually holds the ctrl..

any idea? i dont know what'd be better.

Subject: Re: PROPOSAL: TreeCtrl optionally owns Ctrl's in Node

Posted by [mirek](#) on Wed, 01 Jun 2011 18:49:46 GMT

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

kohait00 wrote on Mon, 23 May 2011 16:57i will be warming up this thing, but need an advice in which direction to go:

the key problem in TreeCtrl is that Node is meant to be passed around by-value, i.e. performing DnD movings or using the Copy() function..

when a Node references a Ctrl which it owns, it cant be passed around by-value, it either needs to be picked (dtor would take care of destrucion of Ctrl) or a different Method for Moving/DnD of the Nodes needs to be found to avoid copy which would duplicate the Node and double ownership. so Node might need to stay unmouvable (exposed only as const or have a subclass which actually holds the ctrl..

any idea? i dont know what'd be better.

Well, general advice in this case is to "give up"

I mean, do not support "predefined" DnD ops for Ctrl owning TreeCtrl. If programmer wants this, he needs to provide his specialized methods.

Mirek
