

---

Subject: Re: hierarchical tree data structure & binding to TreeCtrl

Posted by [mirek](#) on Sun, 11 Nov 2007 22:38:22 GMT

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

---

[quote title=solareclectic wrote on Thu, 01 November 2007 03:35

Now, to the real question... I'm looking for an obvious way to keep that Model/View/Controller separation that we've all been compelled to accept as gospel.

[/quote]

Well, I am afraid, if you are looking for rigid MVC model, you are in the wrong place here...

Quote:

Can I just use TreeCtrl as my datastructure, as it provides all the functionality that I require, including the GUI stuff? Or if MVC separation must be maintained, is there a harm in using two TreeCtrls - one for the View, and one for the Model that just never gets shown?

Actually, why should your model be another TreeCtrl?

Make a real model and refill (or modify) TreeCtrl as required...

Quote:

And, a bonus question... (though for different widgets)

Since I'm taking shots a MVC separation anyway, I'm finding a similar situation with U++ nifty EditField derived widgets, which have all of the range/notNull checking, etc. that I want on to protect my Model object fields from getting set incorrectly, from the GUI or otherwise. If I want to keep, say, an Int datamember in one of my objects AND I want the user to manipulate it, why wouldn't I declare that datamember as EditSpin, instead of Int? Then when I want to present it to the user for editing, I simply have it draw itself?

Are there major performance/memory issues with this behavior?

I am not quite sure I understand the question. Anyway, maybe the answer is that U++ is desgined in a way that you usually do not need have "variable-widget" pairs (using variable to store the data, only use widget for GUI interaction). You do not need the variable, you can store the data directly in widget. You can even think about widgets as "value with possible GUI editing"...

I guess this also replies your "second TreeCtrl" question.

In short, using U++ widgets for other purposes is OK

Mirek

---