Subject: Re: How to create a GridCtrl with fixed cell size
Posted by cbpporter on Thu, 22 May 2008 09:50:17 GMT
View Forum Message <> Reply to Message

[quote title=unodgs wrote on Wed, 21 May 2008 18:45]Quote:
Ok, now I understand. One cell holds one Value object. If you want key/value pair conected with one cell you have to use some structure as Value. As for displaying. You have to extend GridDisplay class which contain some useful methods like SetLeft/Right/CenterImage.
Yes, I know that. This was not a question of how, it was a question of principle. I was asking why isn't there and easier method of doing it. The scenario I described is a fairly common one IMO. It's great that you can use a Display and do really fancy stuff if you need it, but in most cases I just want to display a simple text extracted from the Value in exactly the same position and with same style as if I had inserted a string directly. In such cases, using Display is a little overkill, and witting them can be repetitive and tedious.

Let's take for example a Paint method from Dispaly:
```
virtual void Paint(Draw& w, const Rect& r, const Value& q,
                   Color ink, Color paper, dword style) const
```
It has 6 parameters, and all have useful predefined values which you'll rarely ignore or change and go ahead and use them. A large portion of these displays consists of a simple w.DrawText and maybe some position calculation. Since U++ is in general good at detecting common idioms and offering some API, mechanism or clever trick for it, I was wondering why it doesn't offer something like that in this case. For example, something like:
```
struct MyDisplay: TextDisplay
{
   virtual String Text(const Value &q) const;
}
```