Subject: Re: A question of C++
Posted by mrjt on Fri, 27 Mar 2009 12:29:45 GMT
View Forum Message <> Reply to Message

kbyte wrote on Fri, 27 March 2009 11:29I am trying this:

```
#include "ColoringSystem.h"
...

class CChangeUnitsDlg : public WithChangeUnitsDlgLayout<TopWindow> , public
CColoringSystem
{

}
```

and the CColoringSystem will store all needed colors and the paint of the CChangeUnitsDlg will paint every control 1 by one using the colors in the CColoringSystem class.

Alex

It doesn't work like that, Ctrls are respoionsible for drawing themselves.

You have several options here, but it depends on exactly what you are trying to do:
1- If you want to set colours globally (for all EditCtrls say) you can use EditField::StyleDefault().Write(). You can also do things like SColorFace_Write(Red()) to replace a particular colour.
2- If you need individual colouring for different ctrls of the same type (like username EditString is red, password is Blue) then you are going to have to do something more complicated. Unless you hard-code it this is going to be difficult to do, but possible.
One way would be to use your own version of the Layout macro that creates a string table from a .lay file and then use the table to read styling values from an Xml file. Or something.
3- Tell the user that they can set the colours themselves with ControlPanel->Display and that colours are deliberatley set to match the OS for a good reason