
Subject: ColorScheme for individual cells

Posted by [piotr5](#) on Sat, 05 May 2012 11:57:02 GMT

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

I had to modify ArrayCtrl.h and ArrayCtrl.cpp, to achieve what I want:

in ArrayCtrl.cpp I changed the lines marked with "+":

```
const Display& ArrayCtrl::GetCellInfo(int i, int j, bool f0,
                                     Value& v, Color& fg, Color& bg, dword& st)
{
...
}
if(hasfocus)
    st |= Display::FOCUS;
+ bg = BGColorInfo(i,j);
if(nobg)
    bg = Null;
+ fg = FGColorInfo(i,j);
if((st & Display::SELECT) ||
    !multiselect && (st & Display::CURSOR) && !nocursor ||
    drop) {

...

ArrayCtrl::ArrayCtrl() {
+ ColorScheme=NULL;
```

in ArrayCtrl.h I added to the class ArrayCtrl:

```
public:
struct BGFGFunc {
    virtual Color& BG (const ArrayCtrl* a, int i, int j) const=0;
    virtual Color& FG (const ArrayCtrl* a, int i, int j) const=0;
};

BGFGFunc *ColorScheme;

protected:
const Color& BGColorInfo(int i, int j) const
    {return ColorScheme ? ColorScheme->BG(this,i,j) :
      (i & 1 ? evenpaper : oddpaper);}
const Color& FGColorInfo(int i, int j) const
    {return ColorScheme ? ColorScheme->FG(this,i,j) :
      (i & 1 ? evenink : oddink);}
```

has anybody suggestions how to do that any better?

as for application, I'd like to add duplicate entries into the array, and when they get selected all duplicates get selected too. but of course the user would like to see up front if a duplicate exists, so I colour them differently...

I did think about achieving the same with some richer text-display, but replacing the display for each cell seems a bit of overkill for this simple goal. any other ideas?
