

---

Subject: Re: ArrayCtrl cell color

Posted by [hojtsy](#) on Thu, 02 Feb 2006 20:46:43 GMT

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

---

luzr wrote on Thu, 02 February 2006 14:43

Means, to solve the trouble, use Margin(0) (but you will perhaps have to provide some margins in your Display). Providing the margin inside the Display results in other problems: If the column contains cells with different Display-s (such as lots of StdDisplay cells) all of them should be subclassed to provide the margin. And there seems to be no easy way to make the margin size settable on a per-column basis if the margin is provided by the Display (any many other classes). It is inconvenient for the client code to create multiple subclasses (adding margin for every flavor of Display) just to modify the background color of a cell. I have some other alternative ideas for customizing the margin color without removing the margin, or defining several new classes.

Alternative 1: (new method) void Display::PaintMargin(Draw& w, const Rect& r, const Value& q, Color ink, Color paper, dword s) const

```
{
    w.DrawRect(r, paper);
}
Alternative 2: (new field in the Column and CellInfo):
class Column : FormatConvert {
    ArrayCtrl      *arrayctrl;
    Mitor<int>      pos;
    const Convert   *convert;
    Ptr<Ctrl>       edit;
    const Display   *display;
    const Display   *marginDisplay;
    ....}
```

Obviously both of these alternatives need a small modification in the ArrayCtrl::Paint, to use the method for painting the margin area. And both alternatives have a really small performance hit. But the common benefit of them is that the client code could keep the settable margin in columns and mixed Display cells in the same column without subclassing every Display flavor. Would you consider adding one of these solutions to the library?