

---

Subject: Re: ArrayCtrl cell color

Posted by [mirek](#) on Sun, 19 Feb 2006 22:48:32 GMT

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

---

hojtsy wrote on Fri, 03 February 2006 09:28: It could be PaintBackground. It is a good idea to call this method from the Display::Paint too, because then I only need to redefine the PaintBackground method and not the Paint for recoloring the background.

I was also thinking about how to completely remove the need for client-defined new classes just for the simple purpose of customizing the cell background/text color or font. The default Display could look something like this:

```
void CellDisplay::Paint(Draw& w, const Rect& r, const Value& q,
    Color ink, Color paper, dword s) const
```

```
{
    WString txt;
    Font font = StdFont();

    if(IsType<ArrayCell>(q))
    {
        ArrayCell ac(q);
        txt = ac.txt;
        if(!IsNull(ac.foreground)) ink = ac.foreground;
        if(!IsNull(ac.background)) paper = ac.background;
        if(!IsNull(ac.font)) font = ac.font;
    }
    else {
        txt = IsString(q) ? q : StdConvert().Format(q);
    }

    PaintBackground(w, r, q, ink, paper, s);
    int tcy = GetTLTextHeight(w, txt, StdFont());
    DrawTLText(w, r.left, r.top + max((r.Height() - tcy) / 2, 0), r.Width(), txt, font, ink);
}
```

It's not that complicated or slow in the library and would really simplify the client code when colors/fonts are needed. The client would just use the ArrayCell value in the needed cells and immediately be able to assign colors for each cell withoth any subclassing. I understand that client code could also define this CellDisplay and ArrayCell class, but it seems a common need in many applications, so it could just be in the library.

Implemented as AttrText at StdDisplay level.

Mirek

---