Subject: Re: Custom/Weird Array Setup
Posted by bempson on Sat, 02 Apr 2016 23:33:13 GMT
View Forum Message <> Reply to Message

I thought I would post the code for the version I made just in case anyone was wondering. I also have a screenshot of the result. It's not optimal (because of the base class modifications you need), this is the result of solving the problem with brute force instead of looking for a more correct solution:

```
#ifndef eventtablehpp_h
#define eventtablehpp_h

#include <CtrlLib/CtrlLib.h>

using namespace Upp;

struct CellDisplay : public Display
{
 void Paint(Draw& w, const Rect& r, const Value& q, Color ink, Color paper, dword style) const
 {
  w.DrawRect(r,paper);
  DrawSmartText(w,r.TopLeft().x,r.TopLeft().y-5,5000,String(q),StdFont(),Color(0,0,0));
 }
};

struct CellExtDisplay : public Display
{
 void Paint(Draw& w, const Rect& r, const Value& q, Color ink, Color paper, dword style) const
 {
  w.DrawRect(r,paper);
  DrawSmartText(w,r.TopLeft().x,r.TopLeft().y-10,r.BottomRight().x -
r.TopLeft().x,String(q),StdFont(),Color(0,0,0));
 }
};

class EventTableCtrl : public ArrayCtrl {
 public:
 EventTableCtrl();
 void SetExtData(int,const Value&);
 void SetExtDisplay(const Display&);
 void SetDefaultColumnDisplay(const Display&);
 void Paint(Draw&);
 int InsertRow(const Value&,const Value&,const Value&,const Value&,const Value&);
 void SetRowColor(int,const Color&);
 private:
 const Display *ext_display;
 const Display *column_display;
 Vector<Value> ext_value;
```

```
 Vector<Color> row_color;
};

#endif

#include "eventtable.h"

using namespace Upp;

EventTableCtrl::EventTableCtrl()
 : ext_display(&Single<CellExtDisplay>()), column_display(&Single<CellDisplay>())
{
 NoPopUpEx();
 AddColumn("Label").SetDisplay(*column_display);
 AddColumn("Action").SetDisplay(*column_display);
 AddColumn("Actor").SetDisplay(*column_display);
 AddColumn("Description").SetDisplay(*column_display);
}

void EventTableCtrl::SetExtData(int n,const Value& v)
{
 ext_value.Insert(n,v);
}

void EventTableCtrl::SetExtDisplay(const Display& d)
{
 ext_display = &d;
}

void EventTableCtrl::SetDefaultColumnDisplay(const Display& d)
{
 column_display = &d;
}

void EventTableCtrl::SetRowColor(int row,const Color& color)
{
 row_color[row] = color;
}

int EventTableCtrl::InsertRow(const Value& label,const Value& action,const Value& npc,const
Value& desc,const Value& ext)
{
 int rc = GetCount();
 Vector<Value> vals;
 vals.push_back(String().Cat()<<"\1 [ [*+60 "<<label.ToString()<<"]]");
 vals.push_back(String().Cat()<<"\1 [ [*+60 "<<action.ToString()<<"]]");
 vals.push_back(String().Cat()<<"\1 [ [*+60 "<<npc.ToString()<<"]]");
 vals.push_back(String().Cat()<<"\1 [ [*+60 "<<desc.ToString()<<"]]");
```

```
 if(!ext.IsNull())
  SetExtData(rc,String().Cat()<<"\1 [ [ [@5+50 "<<ext.ToString()<<"]]");
 else
  SetExtData(rc,Value());
 Insert(rc,vals);
 row_color.push_back(Color(255,255,255));
 return rc;
}

void EventTableCtrl::Paint(Draw& w)
{
 SyncColumnsPos();
 bool hasfocus0 = HasFocusDeep();
 Size size = GetSize();
 Rect r;
 r.bottom = 0;
 int i = GetLineAt(GetScroll());
 int xs = -HeaderObject().GetScroll();
 int js;
 for(js = 0; js < GetColumnCount(); js++) {
  int cw = HeaderObject().GetTabWidth(js);
  if ((xs + cw - 0 + (js == GetColumnCount() - 1)) >= 0)
   break;
  xs += cw;
 }
 int sy = 0;
 int dr = HeaderObject().GetTabWidth(0);
 int extlsz = size.cx-dr;
 if(!IsNull(i))
  while(i < GetCount()) {
    int lineYSize=20;
    if(!ext_value[i].IsNull())
    {
     lineYSize += GetSmartTextHeight(ext_value[i].ToString(),extlsz);
     lineYSize += 5; //Row lower margin
    }
    SetLineCy(i, lineYSize);
    r.top = GetLineY(i) - GetScroll();
    if(r.top > size.cy)
     break;
    r.bottom = r.top + GetLineCy(i);
    int x = xs;
    dword st = 0;
    int tw = 0;
    Rect h = r;
    for(int j = js; j < GetColumnCount(); j++)
    {
     tw += HeaderObject().Tab(j).GetMargin();
```

```
    }
    for(int j = js; j < GetColumnCount(); j++) {
     int cw = HeaderObject().GetTabWidth(j);
     int jj = HeaderObject().GetTabIndex(j);
     int cm = HeaderObject().Tab(j).GetMargin();
     if(x > size.cx) break;
     r.left = x;
     r.right = x + cw - 0 + (j == GetColumnCount() - 1);
     dword st;
     Color fg, bg;
     Value q;
     const Display& d = GetCellInfo(i, jj, hasfocus0, q, fg, bg, st);
     (st&Display::CURSOR) ? bg=Color(0,200,230) : bg=row_color[i];
     if(w.IsPainting(r)) {
      if(cw < 2 * cm || editmode && i == cursor/* && ColumnAt(jj).edit*/)
       d.PaintBackground(w, r, q, fg, bg, st);
      else {
       d.PaintBackground(w, RectC(r.left, r.top, cm, r.Height()), q, fg, bg, st);
       r.left += cm;
       r.right -= cm;
       d.PaintBackground(w, RectC(r.right, r.top, cm, r.Height()), q, fg, bg, st);
       w.Clip(r);
       GetDisplay(i, jj).Paint(w, r, q, fg, bg, st);
       w.End();
      }
     }
     x += cw;
    }
    if(ext_display && !ext_value[i].IsNull())
    {
     dword st;
     Color fg, bg;
     ext_display->Paint(w,RectC(dr, r.top+20, extlsz, h.Height()), ext_value[i], fg, bg, st);
    }
    //if(horzgrid)
     w.DrawRect(0, r.bottom, size.cx, 1, gridcolor);

    r.left = 0;
    r.right = x;
    if(i == cursor && !nocursor && multiselect && (GetSelectCount() != 1 || !IsSel(i)) && hasfocus0
&& !isdrag)
     DrawFocus(w, r, st & Display::SELECT ? SColorPaper() : SColorText());
    r.bottom += horzgrid;
    r.left = x;
    r.right = size.cx;
    if(!nobg)
     w.DrawRect(r, SColorPaper);
    if(i == dropline && dropcol == DND_INSERTLINE)
```

```
     DrawHorzDrop(w, 0, r.top - (i > 0), size.cx);
    sy = r.bottom;

    i++;
  }
 int ldy = r.bottom;
 r.left = 0;
 r.right = size.cx;
 r.top = r.bottom;
 if(IsAppendLine() && !IsCursor()) {
  r.bottom = r.top + linecy;
  w.DrawRect(r, HasFocus() ? SColorHighlight : SColorFace);
  r.top = r.bottom;
 }
 r.bottom = size.cy;
 if(!nobg)
  w.DrawRect(r, SColorPaper);
 if(GetCount() == dropline && dropcol == DND_INSERTLINE)
  DrawHorzDrop(w, 0, ldy - 1, size.cx);
 scroller.Set(Point(header.GetScroll(), sb));
 return;
}
```

It overrides the Paint() method of ArrayCtrl to do the wizardry that should be done with displays as mentioned by Lance above. To get this to compile you need to change several "private" sections in ArrayCtrl.h to "protected" to expose some of the functions and members the Paint() method needs.

File Attachments

1) finalprog.png, downloaded 315 times