Subject: Custom/Weird Array Setup
Posted by bempson on Sat, 26 Mar 2016 11:01:57 GMT
View Forum Message <> Reply to Message

I've been mucking around with this for a bit. Was wondering if an ArrayCtrl could be set up like the application image attached.

I would assume that would be a custom widget, with the blue text being some sort of special row attached to the standard row, but I don't know.

Is it possible to get the blue text that spans multiple columns working easily or is this strictly a custom widget setup?

Thank you,
Brian

File Attachments

1) arrayextension.png, downloaded 436 times

Subject: Re: Custom/Weird Array Setup
Posted by Lance on Sat, 26 Mar 2016 19:28:32 GMT
View Forum Message <> Reply to Message

Yes you can. Use SetDisplay on all cells on the rows you want to use blue backgroud. And you need to supply a customized Display for that purpose. You may be able to find a clue by examing ArrayCtrl examples in the examples assembly.

Or optionally you can use GridCtrl which I believe could be easier to achieve similar results.

Subject: Re: Custom/Weird Array Setup
Posted by Novo on Sat, 26 Mar 2016 22:22:04 GMT
View Forum Message <> Reply to Message

A good example could be an "Error" grid in TheIDE. It is declared as an Ide::error member of type ArrayCtrl. All logic is implemented by the class ElepDisplay (file uppsrc\ide\Errors.cpp).

Subject: Re: Custom/Weird Array Setup
Posted by bempson on Sun, 27 Mar 2016 03:07:39 GMT
View Forum Message <> Reply to Message

I've looked through some examples, it looks like this needs to be custom from what I see. There is no way to override how the array draws the row itself (because the blue text in the application mentioned is on the same "row" as the column text, one logical row)

So I'm working on a widget based on ArrayCtrl and overriding it's paint method for that silly blue text that breaks all da rules.

Displays won't override the other columns drawing :(

I really like this toolkit though. I'll post progress just for the sake of it.

Subject: Re: Custom/Weird Array Setup
Posted by bempson on Sun, 27 Mar 2016 08:50:11 GMT
View Forum Message <> Reply to Message

Here is my progress so far.

The red portion is a custom display I set up to render text in that area, seems to be working fine.

I had to subclass ArrayCtrl, modify ArrayCtrl to expose certain private variables (the paint() method depends upon them), and modify the paint() routine to paint my special display after all of the columns in the row.

It's ugly (oh man I changed a base class to do it! yuck!), it's kludgy, but hey, it works so far.

I have a question, if I am drawing text in the red area, how can I notify the Ctrl class itself about the size needed to accomodate the text? Right now the values are statically assigned, but I will need to fix this so that somehow, the paint() routing would have some sort of hint as to the size (height) of the rectangle needed for a certain amount of text.

Any suggestions?

File Attachments

1) progress1.png, downloaded 346 times

Subject: Re: Custom/Weird Array Setup
Posted by bempson on Sun, 27 Mar 2016 10:58:25 GMT
View Forum Message <> Reply to Message

I fiddled with this tonight. Got it to work in principle, word wrap was a pain, I did it manually, which is why there are still glitches and whatnot.

I used size.cx from the paint routine to draw the space for the text, which isn't correct apparently, because the text spans beyond the array widget.

Also I must have an off by one error one the space removal or something. But the widget basically works, so I'm happy, dynamic row size and all! Yay.

I'll get the thing working properly soonish.

File Attachments

1) progress2.png, downloaded 355 times

Subject: Re: Custom/Weird Array Setup
Posted by Lance on Sun, 27 Mar 2016 22:52:32 GMT
View Forum Message <> Reply to Message

Impressive.

Along probably a different path, I created something like this with GridCtrl, QTF and merged cells:

File Attachments

1) GridCtrlWithQtfAndMergedCells.png, downloaded 1003 times

Subject: Re: Custom/Weird Array Setup
Posted by Lance on Sun, 27 Mar 2016 23:31:02 GMT
View Forum Message <> Reply to Message

Here is the code:

```
#include <GridCtrl/GridCtrl.h>
#include <RichText/RichText.h>

using namespace Upp;


// An adhoc protocol
//  A cell will merge all consective cells right to it that are with Null value.
//
// note, there is still (at least) one missing piece, the text is not clipped to cell rect

class QtfGridDisplay : public GridDisplay {
public:
 virtual void Paint(Draw& w, int x, int y, int cx, int cy, const Value& val, dword style,
  Color& fg, Color& bg, Font& fnt, bool found, int fs, int fe)
 {
  GridCtrl *p=this->parent;
  Color rbg=parent->Get(row,0)==1 ? Color(0,250,128) : bg;

  int realcx=cx;
  if(col!=1 && IsNull(p->Get(row,col)))
   return; // the cell is merged, so no painting for it
  for(int c=col+1; c<p->GetColumnCount();++c)
   if(IsNull(p->Get(row,c)))
    realcx+=p->GetWidth(c+2);
   else
    break;
  w.DrawRect(x,y,realcx,cy, rbg);
  DrawSmartText(w,x,y,realcx,String(val),fnt,fg);
 }

 static QtfGridDisplay& get(){ static QtfGridDisplay d; return d; }
};

class GridHilite : public TopWindow {
public:
 typedef GridHilite CLASSNAME;
 GridHilite();

private:
 GridCtrl grid1;
};


GridHilite::GridHilite()
{
 grid1.AddHiddenColumn("bg"); // hidden column with value 1 will be painted with a greenish
background
 grid1.AddColumn("Label");
 grid1.AddColumn("Action");
 grid1.AddColumn("NFC");
 grid1.AddColumn("Description");


 String s="\1[ [ [*+75 Set Flags]&][ [@5+75 MiguelFollowing`=true]&][ [@5+75
MiguelLeaving`=false]&][ [@5+75 MiguleWandering`=false]]]";
 grid1.AddRow(1,"\1[*+75
Wander]",s).SetDisplay(QtfGridDisplay::get()).Height(GetSmartTextHeight(s,500));
 grid1.AddRow(0,"","\1[*+75 Trigger]","","\1[*+75Trigger
nsfwander]").SetDisplay(QtfGridDisplay::get());


 // Here are 4 plain text, I will put it in the 2nd column, and make it with hilite background;
 String s1="Line 1";
 String s2="grid1.VertGrid(false).HSizePosZ(2,2).VSizePosZ(2,2)";
 String s3="grid1.AddColumn(\"Description\")";
 String s4="OK, that's about it";

 String combined=String().Cat()<<'\1'<<"[ [ [*+75 "<<DeQtf(s1)<<"]&]"
```

```
  "[ [@5+75 "<<DeQtf(s2)<<"]&]"   // use DeQtf to escape certain special characters
  "[ [@5+75 "<<DeQtf(s3)<<"]&]"   // use MainMenu->Assist->QtfDesigner to get markup strings.
  "[ [@5+75 "<<DeQtf(s4)<<"]]";

 grid1.AddRow(1,"",combined).SetDisplay(QtfGridDisplay::get())
  .Height(GetSmartTextHeight(s,500));

 grid1.VertGrid(false).HSizePosZ(2,2).VSizePosZ(2,2);
 Add(grid1);
}

GUI_APP_MAIN
{
 GridHilite().Run();
}
```

Subject: Re: Custom/Weird Array Setup
Posted by bempson on Mon, 28 Mar 2016 07:00:51 GMT
View Forum Message <> Reply to Message

Awesome. I'm probably going to use your approach, mine butchered a base class and is generally garbage from a coding standpoint.

I'll have to take a close look at this!

Subject: Re: Custom/Weird Array Setup
Posted by bempson on Fri, 01 Apr 2016 04:17:34 GMT
View Forum Message <> Reply to Message

Just got a chance to try your code. Works pretty good and doesn't require any hacks like mine did.

Just one question (had this problem with mine), is there a way to get the selection (blue color) to overlay the cells with the green background when selected?

Thank you for the awesome code snippet. Would it be worth it to post mine?

Subject: Re: Custom/Weird Array Setup
Posted by Lance on Fri, 01 Apr 2016 19:20:43 GMT
View Forum Message <> Reply to Message

Do you mean you want a visual clue when the greenish-backgrounded row's are selected, e.g., give it a darker background color?

If that's what you are asking, the answer is yes, and it's quite simple to achieve it. Make use of the style parameter to determine if the current cell is selected, and supply a different background color to your like, and that's it.

Subject: Re: Custom/Weird Array Setup

Posted by Lance on Fri, 01 Apr 2016 19:38:24 GMT

View Forum Message <> Reply to Message

To save you some search into GridCtrl code. Also, text clipping added.

```
#include <GridCtrl/GridCtrl.h>
#include <RichText/RichText.h>

using namespace Upp;


// An adhoc protocol
//  A cell will merge all consective cells right to it and with Null value.
//
// note, there is still (at least) one missing piece, the text is not clipped to cell rect
class QtfGridDisplay : public GridDisplay {
public:
 virtual void Paint(Draw& w, int x, int y, int cx, int cy, const Value& val, dword style,
  Color& fg, Color& bg, Font& fnt, bool found, int fs, int fe)
 {
  GridCtrl *p=this->parent;
  Color rbg=parent->Get(row,0)!=1? bg: (style&GD::CURSOR)? Color(0,200,85):Color(0,250,128);

  int realcx=cx;
  if(col!=1 && IsNull(p->Get(row,col)))
   return; // the cell is merged, so no painting for it

  for(int c=col+1; c<p->GetColumnCount();++c)
   if(IsNull(p->Get(row,c)))
    realcx+=p->GetWidth(c+2);
   else
    break;
  w.DrawRect(x,y,realcx,cy, rbg);
  w.Clip(x,y,realcx,cy);
  DrawSmartText(w,x,y,realcx,String(val),fnt,fg);
  w.End();
 }

 static QtfGridDisplay& get(){ static QtfGridDisplay d; return d; }
};
```

```
class GridHilite : public TopWindow {
public:
 typedef GridHilite CLASSNAME;
 GridHilite();

private:
 GridCtrl grid1;
};


GridHilite::GridHilite()
{
 grid1.AddHiddenColumn("bg");
 grid1.AddColumn("Label");
 grid1.AddColumn("Action");
 grid1.AddColumn("NFC");
 grid1.AddColumn("Description");

 grid1.SetDisplay(QtfGridDisplay::get());


 String s="\1[ [ [*+75 Set Flags]&][ [@5+75 MiguelFollowing`=true]&][ [@5+75
MiguelLeaving`=false]&][ [@5+75 MiguleWandering`=false]]]";
 grid1.AddRow(1,"\1[*+75 Wander]",s).Height(GetSmartTextHeight(s,500));
 //grid1.JoinCells(2,0,4,0);

 grid1.AddRow(0,"","\1[*+75 Trigger]","","\1[*+75Trigger nsfwander]");


 // Here are 4 plain text, I will put it in the 2nd column, and make it with hilite background;
 String s1="Line 1";
 String s2="grid1.VertGrid(false).HSizePosZ(2,2).VSizePosZ(2,2)";
 String s3="grid1.AddColumn(\"Description\")";
 String s4="OK, that's about it";

 String combined=String().Cat()<<'\1'<<"[ [ [*+75 "<<DeQtf(s1)<<"]&]"
  "[ [@5+75 "<<DeQtf(s2)<<"]&]"   // use DeQtf to escape certain special characters
  "[ [@5+75 "<<DeQtf(s3)<<"]&]"   // use MainMenu->Assist->QtfDesigner to get markup strings.
  "[ [@5+75 "<<DeQtf(s4)<<"]]";

 grid1.AddRow(1,"",combined,Null,"\1[*+75 Here]")
  .Height(GetSmartTextHeight(s,500));

 grid1.VertGrid(false).HSizePosZ(2,2).VSizePosZ(2,2);
 Add(grid1);
}

GUI_APP_MAIN
```

```
{
 GridHilite().Run();
}
```

Subject: Re: Custom/Weird Array Setup
Posted by bempson on Sat, 02 Apr 2016 09:54:20 GMT
View Forum Message <> Reply to Message

Thank you! This solution works wonderfully, I can't thank you enough.

This IDE/Toolkit combo is brilliant, I'm still new to it, but I'm amazed at how easy it can be to get the results you want.

QTF is a nice way to manipulate rich text.

Subject: Re: Custom/Weird Array Setup
Posted by Lance on Sat, 02 Apr 2016 21:57:03 GMT
View Forum Message <> Reply to Message

You're welcome, bempson.

I feel you have very good c++ background. U++ will suit you well. You just need some time getting use to it.

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 324 times

Subject: Re: Custom/Weird Array Setup
Posted by Lance on Sun, 03 Apr 2016 12:10:02 GMT
View Forum Message <> Reply to Message

Thanks for sharing, bempson. Nice job! I know your pain. Some U++ classes is hard to derive from. Or probably they are not meant to be derived for unknown reason.