Subject: [Solved albeit ugly] Scrolling messes up GridCtrl in certain condition
Posted by Lance on Thu, 28 Feb 2013 12:06:19 GMT
View Forum Message <> Reply to Message

See attached sample code.

Run the example, click "refresh" button to populate the GridCtrl. Drag the vertical scollbar and move down, the GridCtrl will be messed up.

Also, the sum row reports incorrect total. The last row in the GridCtrl is not counted. Click any column header to sort the GridCtrl, the correct sum will appear. Call UpdateSummary() manually after changing data will fix it. Not sure if this is required by design.

```
SaleInvoiceListTab::SaleInvoiceListTab()
{
 CtrlLayout(*this);
 refresh<<=THISBACK(RefreshReport);
 //RefreshReport();

        // Note: uncomment above line will eliminate the Paint issue

}
```

Thanks,

Lance

File Attachments

1) TestGridSummary.zip, downloaded 468 times

Subject: Re: [GridCtrl] Scrolling messes up GridCtrl in certian condition
Posted by Lance on Fri, 01 Mar 2013 23:42:55 GMT
View Forum Message <> Reply to Message

A temporary fix for the above problem(In my case, open with populated grid is not practical):

remove the GridCtrl from its parent and add back after repopulation.

And also pay special attention to the aggregate functions of GridCtrl. Do a UpdateSummary() after repopulation since the very last row might be skipped.

Disclaimer: it might be because I don't know the proper (designed) way of using the Ctrl. E.g., months back I learned that programmatically changing column widths should be sandwiched in

Ready(false) and Ready(true). Note in this case, I tried to do the same (sandwich data changes in Ready(...) calls) but to no avail.

PS: Remove & Add back will fix the incorrect summary issue.

I also tried Hide() - change - Show(), this way the problem remains. It seems some action taken when a GridCtrl is initially activated has not been synchronized properly by subsequent changes to the GridCtrl structure/data.

Subject: Re: [GridCtrl] Scrolling messes up GridCtrl in certian condition
Posted by Lance on Sat, 02 Mar 2013 00:20:41 GMT
View Forum Message <> Reply to Message

The following code in GridCtrl.cpp tells us why reason==OPEN is so special as Init() is called within and everything is guaranteed to work well before further changes. Unfortunately Init() and the most or all the method it calls are private, so RemoveCtrl() and Add() is indeed the best thing we can do for now.

```
void GridCtrl::Init()
{
 bar.Set(WhenToolBar);
 UpdateCols(true);
 /* recalc_rows bo przed otworzeniem grida moglo zostac wywolane setrowheight */
 UpdateRows(resize_row_mode > 0 || recalc_rows);

 UpdateSizes();
 UpdateSb();
 UpdateHolder(true);
 SyncSummary();
 SyncCtrls();
}

void GridCtrl::State(int reason)
{
 if(reason == OPEN)
 {
  Init();
  ready = true;
  //ready po init - updatesb wola layout() a ten syncctrl
  //(ktory w sumie wola sie 3 razy zanim grid sie wyswietli - niepotrzebnie)
 }
```

Subject: Re: [GridCtrl] Scrolling messes up GridCtrl in certian condition
Posted by unodgs on Sat, 02 Mar 2013 12:16:17 GMT
View Forum Message <> Reply to Message

Thank you for the test case and the diagnosis. I'm working on the fix.

Subject: Re: [GridCtrl] Scrolling messes up GridCtrl in certian condition
Posted by Lance on Sat, 02 Mar 2013 17:21:04 GMT
View Forum Message <> Reply to Message

You are very welcome!

And thank YOU for this powerful and popular control!

If only it supports external data and finer control (eg through callbacks/gates for customized sorting, etc)

The control is so complicated that I cannot figure out a way to adapt it in those directions.

Subject: Re: [GridCtrl] Scrolling messes up GridCtrl in certian condition
Posted by unodgs on Sat, 02 Mar 2013 21:30:09 GMT
View Forum Message <> Reply to Message

Lance wrote on Sat, 02 March 2013 12:21
And thank YOU for this powerful and popular control!

My pleasure!. It's good to know it's useful for other people.
Quote:
If only it supports external data and finer control (eg through callbacks/gates for customized sorting, etc)

Please describe in more details. Maybe some of those are easy to add.
Quote:
The control is so complicated that I cannot figure out a way to adapt it in those directions.

I know. It's even too complicated to me Especially embedded controls part. Too many options mixed in one place. Not easy to extend. But that's my fault..
Few years ago I planned to create better (with code refactored) GridCtrl 2.0 that would also support grouping and "freely positioned columns" in a header as well as in a row. I know it sounds cryptic but I mean something similar to the grid from DevExpress. They can for example have one column under the other.
I started some work on it but it turned out to be quite complicated and I suspended the work. Maybe I'll get back to it maybe not it simply depends on how many of my commercial apps will use u++ in a future. I really like native apps and I used to choose them over the web apps. But over the time browsers changed as well as html and my point of view changed too.
But I'm still using my control a lot in my existing applications and I'm open to add new features if

they make sense. I will also try to finish grid's documentation because it's hard for the users to use all available functions if they are not aware of them

Sorry for a long answer. In fact I wanted to let you know that bug is fixed and the code is in svn. Please check!

Subject: Re: [GridCtrl] Scrolling messes up GridCtrl in certian condition
Posted by Lance on Sun, 03 Mar 2013 01:03:37 GMT
View Forum Message <> Reply to Message

Hi Daniel:

Thanks for the quick action.

Quote:

Quote:
If only it supports external data and finer control (eg through callbacks/gates for customized sorting, etc) Smile

Please describe in more details. Maybe some of those are easy to add.

http://www.ultimatepp.org/forum/index.php?t=msg&goto=358 78&#msg_35878

GridCtrl and even ArrayCtrl are complicated mostly because they have to be, I guess. I spent many hours on each of them but fail to understand either very well.

Again, thank you very much for time and effort contributed to U++ by you, Mirek, Dolik, Koldo, and many others.