
Subject: Re: How to create a GridCtrl with fixed cell size
Posted by [cbpporter](#) on Tue, 20 May 2008 00:23:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

mrjt wrote on Mon, 19 May 2008 19:09: It seems to me that what you need is very similar to ColumnList, with only 2 (main) differences:

- items drawn left to right (not top to bottom)
 - number of columns determined dynamically based on item width
- so my suggestion would be to modify that.

I've attached my quick and dirty attempt (it works quite well though). Use like a ColumnList, but you'll have to set ItemWidth to get sensible results when using a Display (default width is 50). Thank you very much for using your free time to try this. You are right: I do need something like a ColumnList: actually I was using that before, but I need another draw order. It seems that the way such a list with multiple columns is drawn is confusing for a lot of people, so I had to disable the multiple column option, leaving it to a default of one. But because of the waste of space, I find myself in this situation...

I'll definitely try out your code and probably merge it into ColumnList if possible, seeing as U++ lacks a traditional list control with an optional "icons" layout (like CListCtrl from MFC).

But before that, since I already started with GridCtrl, I have to clear up some issues first.

I used this pretty ugly code to get the control to look and behave as I want it. By using Add variants I could only get a static number of items on a row, so I had to use Set:

```
void KanjiFlash::PopulateGrid()
{
    if (!curList)
        return;

    grdKanji.Clear();
    Vector<Kanji *> &v = *curList;
    int cx = 0, cy = 0;

    int w = grdKanji.GetSize().cx / CellHeight;
    int h = v.GetCount() / w;
    if (v.GetCount() * 1.0 / w != h)
        h++;

    //section 1
    //for (int i = 0 ; i < h; i++)
    // grdKanji.AddRow("").AlignBottom().Height(CellHeight);
    for (int i = 0; i < v.GetCount(); i++)
    {
        const Kanji& kanji = *v[i];
        grdKanji.Set(cy, cx, kanji.Literal());
        cx++;
    }
}
```

```

if (cx >= w)
{
    cx = 0;
    cy++;
}
}

//section 2
//for (int i = 0 ; i < h; i++)
// grdKanji.GetRow(i).AlignBottom().Height(CellHeight);
}

void KanjiFlash::Layout()
{
    int w = grdKanji.GetSize().cx / CellHeight;

    grdKanji.Ready(false);
    grdKanji.Clear(true);
    for (int i = 0; i < w; i++)
        grdKanji.AddColumn("", CellHeight);
    grdKanji.Ready(true);
    PopulateGrid();
}

```

This pretty much works, except some problems.

There are two sections of commented out code. I need to uncomment the first section If I want to adjust the look of each individual row. Here I get two problems:

1. The control doesn't seem to show the scrollbar correctly, and even if it shows it, sometimes the last row is scrolled only about to the middle. It seems that the Height statement is the culprit (CellHeight is 40 under my tests).
2. AlignBottom or any other align doesn't seem to work.

If I uncomment the second section (it doesn't mater if the first is commented or not), rows are drawn at incorrect positions, but react to mouse at correct coordinates.

Also on a side note, I couldn't find a GetRowCount, event though a GetColumnCount is present.