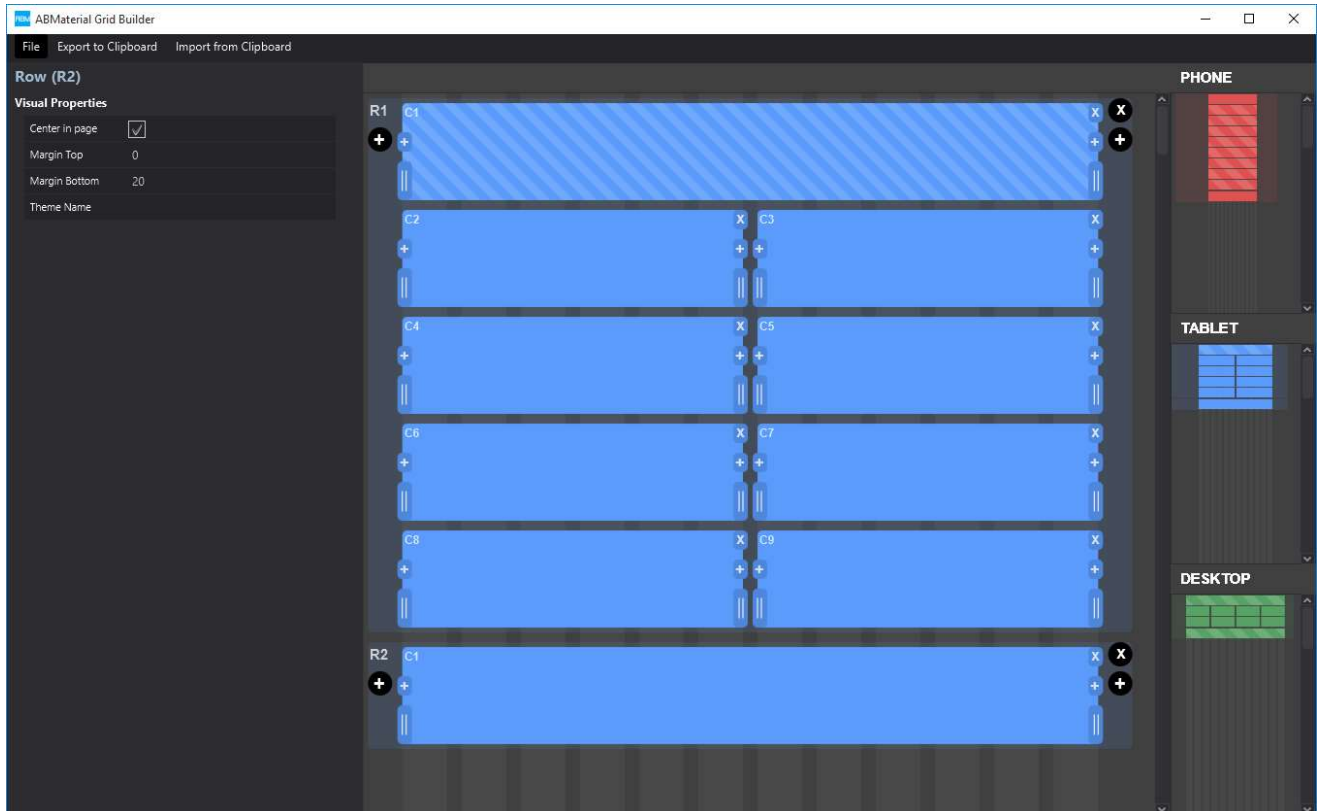


Introduction

Take a look at this section to quickly understand how grids work! Make sure you have read the Themes topic first. We use themes when we design a grid.

In ABMaterial 2.20+, we have a GRID BUILDER to make it easy to build Grids. You can find this chapter at the end of this topic, but it is very important you grab the concept of the Grid and Cells before proceeding!



12 Cells

Our standard grid has 12 cells. No matter the size of the browser, each of these cells will always have an equal width.



To get a feel of how the grid is used in B4J, take a look at the code below which will produce a grid definition used in this page.

```
// create the page grid
.AddRows(1,True, "").AddCells12(1, "")
.AddRowsM(1,True,0,0, "").AddCellsOS(12,0,0,0,1,1,1, "")
.AddRows(3,True, "").AddCells12(1, "")
.AddRowsM(1,True,0,0, "").AddCellsOS(2,0,0,0,6,6,6, "")
.AddRows(2,True, "").AddCells12(1, "")
.AddRowsM(1,True,0,0, "").AddCellsOS(1,6,6,6,6,6,6, "")
.AddRows(6,True, "").AddCells12(1, "")
.BuildGrid //IMPORTANT once you loaded the complete grid AND before you start adding
components
```

Cells live inside rows

You can create columns in a row by dividing the 12 cells into parts. For example here we use the 12 cells to create 2 columns, each 6 cells wide.

This row is 12-cells wide on all screen sizes

6-cells (one-half)

6-cells (one-half)

Offsets

To offset use the offset parameters. Offsets are **relative** to the previous cell in the row (or 0 for the first one). This way you can create 'gabs' between cells. In the next example we create a 6-cell column with an offset by 6.

This row is 12-cells wide on all screen sizes

6-cells (offset-by-6)

Screen sizes and cell offsets/sizes

Some AddCellxxx() methods accept 6 extra parameters to set the offsets and sizes for small, medium and large screens so you can make your app adept to different device sizes (phone, tablet, desktop).

An overview of the different AddRowxx() methods

AddRows(numberOfRows, centerInRow, themeName)

Param	Description
numberOfRows	You can use this schortcut to add the same row multiple times. Instead of writing 3 times 'page.AddRows(1,True,"").AddCells12(1,"")', you can write 'page.AddRows(3,True,"").AddCells12(1,"")'. Uses a default marginTopPx and marginBottomPx of 20 pixels.
centerInRow	If true, the row will use approx between 75% and 85% of the available screen width, creating some sort of border. (% depends on screen size)
themeName	Optional name of the theme you have created. Can be "".

AddRowsM(numberOfRows, centerInRow, marginTopPx, marginBottomPx, themeName)

Param	Description
numberOfRows	You can use this schortcut to add the same row multiple times. Instead of writing 3 times 'page.AddRows(1,True,"").AddCells12(1,"")', you can write 'page.AddRows(3,True,"").AddCells12(1,"")'
centerInRow	Tune the top margin of the row in pixels, can be negative.
marginTopPx	Tune the bottom margin of the row in pixels, can be negative.
marginBottomPx	If true, the row will use approx 75% and 85% of the available screen width, creating some sort of border. (% depends on screen size)

Param	Description
themeName	Optional name of the theme you have created. Can be "".

An overview of the different AddCellxx() methods

AddCells12(numberOfRows, themeName)

Param	Description
numberOfRows	You can use this schortcut to add the same cell multiple times. Instead of writing 3 times 'page.AddRows(1,True, "")'.AddCells12(1,"')', you can write 'page.AddRows(1,True, "").AddCells12(3,"')'. In this method this has a certain side effect: As described above, one row can only have 12 cells. So creating 3 x 12 Cells will 'wrap' them to the next line.
themeName	Optional name of the theme you have created. Can be "".

AddCellsOS(numberOfCells, offsetSmall, offsetMedium, offsetLarge, sizeSmall, sizeMedium, sizeLarge, themeName)

Param	Description
numberOfRows	You can use this schortcut to add the same cell multiple times.
offsetSmall	Offset of the cell on small screens.
offsetMedium	Offset of the cell on medium screens.
offsetLarge	Offset of the cell on large screens.
sizeSmall	Size of the cell on small screens.
sizeMedium	Size of the cell on medium screens.
sizeLarge	Size of the cell on large screens.
themeName	Optional name of the theme you have created. Can be "".

AddCells12MP(numberOfCells, marginTopPx, marginBottomPx, paddingLeftPx, paddingRightPx, themeName)

Param	Description
numberOfRows	You can use this schortcut to add the same cell multiple times.
marginTopPx	Tune the top margin of the cell in pixels, can be negative.
marginBottomPx	Tune the bottom margin of the cell in pixels, can be negative.
paddingLeftPx	Tune the left padding of the cell in pixels.
paddingRightPx	Tune the right padding of the cell in pixels.
themeName	Optional name of the theme you have created. Can be "".

AddCellsOSMP(numberOfCells, offsetSmall, offsetMedium, offsetLarge, sizeSmall, sizeMedium, sizeLarge, marginTopPx, marginBottomPx, paddingLeftPx, paddingRightPx, themeName)

Param	Description
numberOfRows	You can use this schortcut to add the same cell multiple times.
offsetSmall	Offset of the cell on small screens.

Param	Description
offsetMedium	Offset of the cell on medium screens.
offsetLarge	Offset of the cell on large screens.
sizeSmall	Size of the cell on small screens.
sizeMedium	Size of the cell on medium screens.
sizeLarge	Size of the cell on large screens.
marginTopPx	Tune the top margin of the cell in pixels, can be negative.
marginBottomPx	Tune the bottom margin of the cell in pixels, can be negative.
paddingLeftPx	Tune the left padding of the cell in pixels.
paddingRightPx	Tune the right padding of the cell in pixels.
themeName	Optional name of the theme you have created. Can be "".

Some guidelines

I'm aware this is not an easy part to understand. Start with simple layouts, and use the more advanced methods once you start understanding what is going on. In the beginning I found it usefull to draw my Grid on paper to visualize it. Knowing exactly how your page needs to look is an important part of the design. Further into the tutorial, we're going to meet controls (like ABMContainer) that allows to create grids within grids so again it's **VERY IMPORTANT** you grab this concept. If you have to start adding rows (or even cells) afterwards, you're in for some work going over your code making adjustments!

This is a representation of the grid of this page on paper. Compare this image with the code block at the beginning of the page.

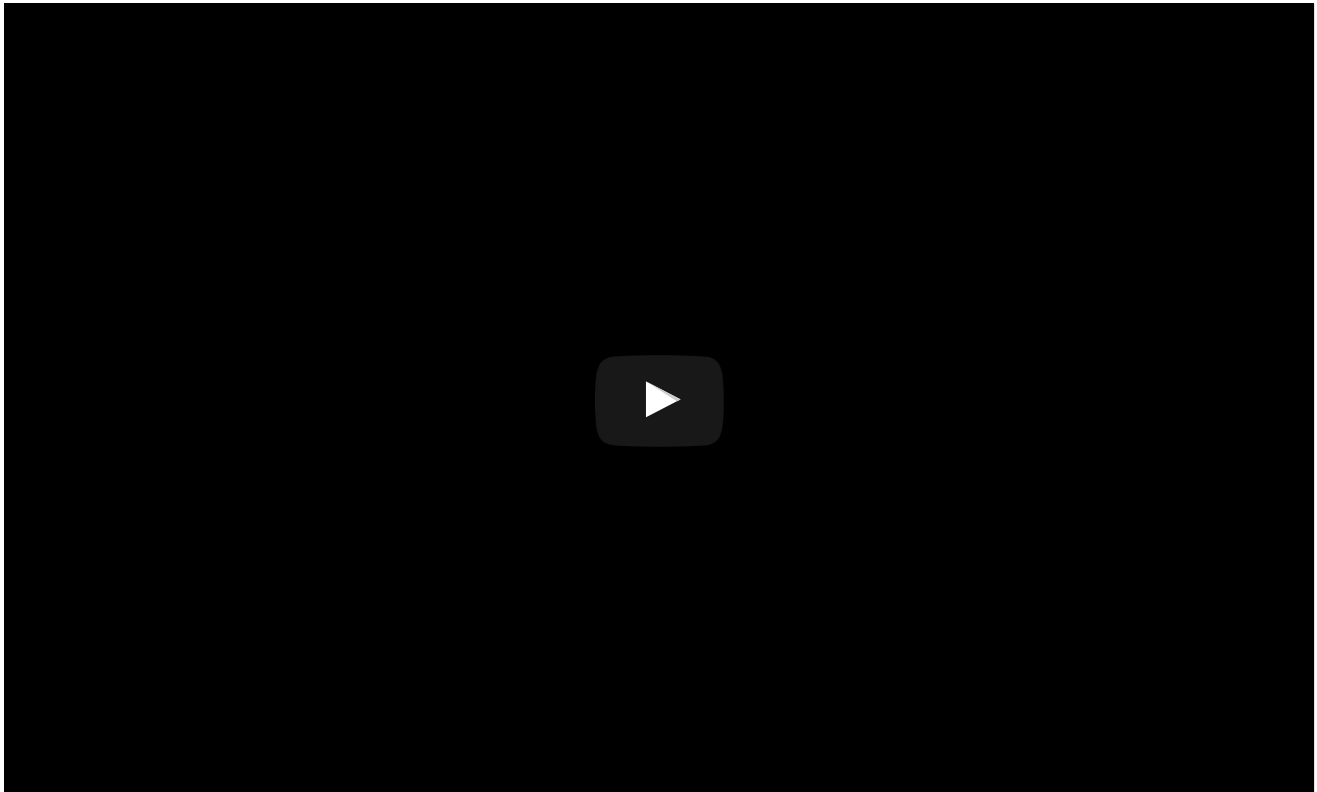
The ABMGridBuilder

The ABMGridBuilder is a tool to make grids easy. It is based on the excellent system introduced by Shoelace.io. I adapted the system so it is usefull for ABMaterial and added some specific functionalities like adding rows and cells before others.

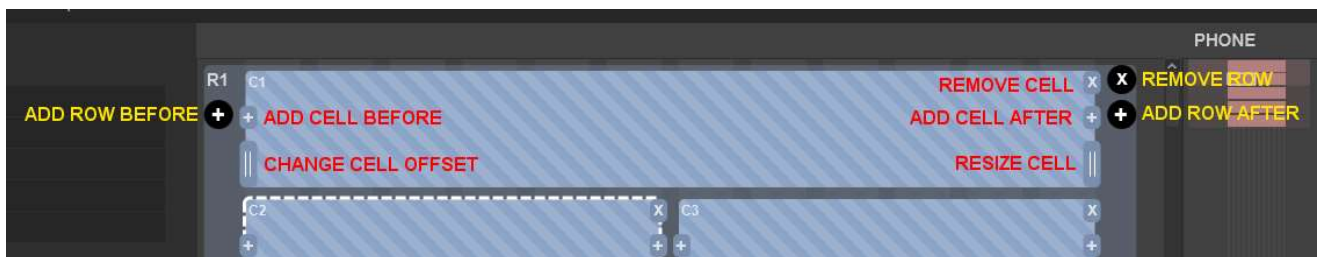
With the tool, you can generate the needed AddRow and AddCell B4J methods described before. Using the Clipboard, you can then paste it at the relevant place in your code.

But even more! You can copy your AddRow and AddCell methods to the clipboard, and ABMGridBuilder can parse this code an 'rebuild' the visual representation. Cool, no?

Have a look at this video which demonstrates the whole story.



Using the function buttons on the cells and the grid you can add/remove/move/resize the rows and cells.



Using the clipboard, you can paste the generated code into your application. You can select the AddRows lines in you code and 'rebuild' it to a visual representation of your grid. This is particulary useful for grids you manually created in ABMaterial version before version 2.20.

This is just a quick introduction, so make sure you also check the B4J forum for some more tutorials and tips & tricks on the ABMGridBuilder.

Next chapter to read should be the ABMPage object in the Helpers menu.